GSCL 2015

International Conference of the German Society for Computational Linguistics and Language Technology

Proceedings of the Conference

Sep 30 – Oct 2, 2015
University of Duisburg-Essen, Germany

Production and Manufacturing by
Gesellschaft für Sprachtechnologie and Computerlinguistik e.V.

Preface

This year's GSCL conference was held at campus Essen of the University of Duisburg-Essen. The University of Duisburg-Essen has connections to two major currents in computational linguistics: On the one hand, especially the German department has favoured the 'traditional' linguistic approaches associated with 'deep models' which try to do justice to the complexity of both linguistic reality and research traditions and which also have deep link to cognitice science, psychology, semiotics, and symbolic artificial intelligence. On the other hand, while historically also connected to symbolic artificial intelligence in the person of Wolfang Höppner, the computer science department has lately been favouring 'shallow' statistical approaches of natural language processing that tend to be geared towards concrete problems, while not eschewing fundamental questions.

The organizers decided early on to adapt the different currents of computational linguistics as the theme for the conference. Despite the different orientations of the organizers (German linguists and computer scientists), the organization of the conference went very harmoniously and consensus could be reached on bibliography standards, time-slots, and conference dinner menus. Judging from the interactions we saw during the conference, this harmony also transferred to the scientific dialogue between the two 'camps' we constructed in the first paragraph of this preface. It is our conviction that the camps are not meant to be separate, and that it is one of GSCL's main assets that it brings together a heterogeneous community from the German-speaking countries and ensures its existence, thus fostering a dialogue that helps either side to develop and leads to a certain convergence.

The two invited talks took up the theme of the conference and also presented some incentives for convergence. The talk by Chris Bieman (nominated by the 'shallow' camp) discussed *Adaptive Natural Language Processing*, and thus took up an issue that is dear to 'deep' linguists (if from a different angle): Language changes continually, and hence a linguistic model must be adaptive, and accommodate linguistic change. Biemann's self-improving linguistic models are a step in this direction. Aarne Ranta (invited by the 'deep' camp) talked about *Interlinguas: Deep and Shallow*, taking the example of different applications designed with his Grammatical Framework for a discussion of the adequacy of certain degrees of 'depth' for concrete work.

The programme of the conference, too, mirrored the different currents of the field of computational linguistics: both 'deep' and 'shallow' models, linguistic ressorces that can be used by either 'side', but it also accommodated papers from the related fields of digital humanities / edition philology and 'pure' (computer-aided) linguistics.

A great number of the submissions lent themselves to poster or demo presentations, which is why the conference had several slots of lightning talks that served as teasers for these presentations, and also a lot of time has ben alotted for viewing the posters and demos.

We thank all participants for making the conference as interesting as it was,

Bernhard Fisseni
Bernhard Schröder
Torsten Zesch

Chairs:

Bernhard Fisseni (University of Duisburg-Essen)
Bernhard Schröder (University of Duisburg-Essen)
Torsten Zesch (University of Duisburg-Essen)

Local Organizers:

Tim Kocher (University of Duisburg-Essen)
Charlotte Wollermann (University of Duisburg-Essen)

Program Committee:

Michael Beißwenger, Universität Dortmund
Delphine Bernhard, Université de Strasbourg
Chris Biemann, Technische Universität Darmstadt
Philipp Cimiano, Universität Bielefeld
Bertold Crysmann, CNRS, Paris
Michael Cysouw, University of Marburg, Germany
Stefanie Dipper, Ruhr-Universität Bochum
Markus Egg, Humboldt-Universität Berlin
Anette Frank, Universität Heidelberg
Jost Gippert, Goethe-Universität Frankfurt
Ulrich Heid, Universität Hildesheim
Gerhard Heyer, University of Leipzig
Tobias Horsmann, University of Duisburg-Essen
Roman Klinger, University of Bielefeld
Markus Kracht, University of Bielefeld
Lothar Lemnitzer, BBAW
Nils Lenke, Nuance Communications, Aaachen
Henning Lobin, Universität Gießen
Anke Lüdeling, Humboldt-Universität zu Berlin
Cerstin Mahlow, Universität Stuttgart
Alexander Mehler, Universität Frankfurt
Stefan Müller, Freie Universität Berlin
Günter Neumann, DFKI Saarbrücken
Uwe Quasthoff, University of Leipzig
Karola Pitsch, Uni Duisburg-Essen
Georg Rehm, DFKI
Frank Richter, Goethe-Universität Frankfurt
Stefan Riezler, Universität Heidelberg
Manfred Sailer, Goethe-Universität Frankfurt
David Schlangen, Universität Bielefeld
Roman Schneider, IDS Mannheim
Thomas Schmidt, Institut für Deutsche Sprache (IDS) Mannheim
Ulrich Schmitz, University of Duisburg-Essen
Manfred Stede, University of Potsdam

Benno Stein, Universität Weimar
Angelika Storrer, Universität Dortmund
Elke Teich, Universität des Saarlandes
Tonio Wandmacher, Systran, Paris, France
Michael Wiegand, Universität des Saarlandes
Andreas Witt, Institut für Deutsche Sprache (IDS) Mannheim
René Witte, Concordia University, Montréal
Heike Zinsmeister, Universität Hamburg

Invited Speakers:

Chris Biemann, Technische Universität Darmstadt
Aarne Ranta, University of Gothenburg

Table of Contents

Conference Program

Invited Talks

Adaptive Natural Language Processing
Chris Biemann

Interlinguas: Deep and Shallow
Aarne Ranta

Full Papers

A Reinforcement Learning Approach for Adaptive Single- and Multi-Document Summarization
Stefan Henss, Margot Mieskes and Iryna Gurevych

Did I Really Say That? – Combining Machine Learning and Dependency Relations to Extract Statements from German News Articles
Thomas Bögel and Michael Gertz

Fast or Accurate? – A Comparative Evaluation of PoS Tagging Models
Tobias Horsmann, Nicolai Erbs and Torsten Zesch

GermaNER: Free Open German Named Entity Recognition Tool
Darina Benikova, Seid Muhie Yimam, Prabhakaran Santhanam and Chris Biemann

God Wat Þæt Ic Eom God – An Exploratory Investigation Into Word Sense Disambiguation in Old English
Martin Wunderlich, Alexander Fraser and Paul Sander Langeslag

Growing Trees from Morphs: Towards Data-Driven Morphological Parsing
Petra Steiner and Josef Ruppenhofer

Rule-based Dependency Parse Collapsing and Propagation for German and English
Eugen Ruppert, Jonas Klesy, Martin Riedl and Chris Biemann

Systematic Acquisition of Reading and Writing: An Exploration of Structure in Didactic Elementary Texts for German
Kay Berkling, Rémi Lavalley and Uwe Reichel

Wie oft schreibt man das zusammen? The Puzzle of Why some Separable Verbs in German are More Separable than Others
Nana Khvtisavrishvili, Stefan Bott and Sabine Schulte im Walde

WISE: A Web-Interface for Spelling Error Recognition for German: A Description and Evaluation of the Underlying Algorithm
Kay Berkling and Rémi Lavalley

Posters

A Case-study of Automatic Participant Labeling
Alexander Kampmann, Stefan Thater and Manfred Pinkal

A Hybrid Approach to Extract Temporal Signals from Narratives
Thomas Bögel, Jannik Strötgen and Michael Gertz

A Resource for Natural Language Processing of Swiss German Dialects
Nora Hollenstein and Noëmi Aepli

Annotating Modality Interdependencies
Eva Reimer, Bianka Trevisan, Denise Eraßme, Thomas Schmidt and Eva-Maria Jakobs

Annotation and analysis of the LAST MINUTE corpus
Dietmar Rösner, Rico Andrich, Thomas Bauer, Rafael Friesen and Stephan Günther

Automatic Induction of German Aspectual Verb Classes in a Distributional Framework
Jürgen Hermes, Michael Richter, and Claes Neuefeind

Correlation between Lexical and Determination Types
Oliver Hellwig and Wiebke Petersen

Digitale Kuratierungstechnologien: Verfahren für die effiziente Verarbeitung, Erstellung und Verteilung qualitativ hochwertiger Medieninhalte
Georg Rehm and Felix Sasaki

Entering Appointments: Flexibility and the Need for Structure?
Karola Pitsch, Ramin Yaghoubzadeh and Stefan Kopp

Konventionalisierung und Interaktion – das Pledari Grond Online
Claes Neuefeind

Towards Parsing Language Learner Utterances in Context
Christine Köhn and Wolfgang Menzel

Demos

Adaptive Natural Language Processing

Chris Biemann
Technische Universität Darmstadt
biem@cs.tu-darmstadt.de

Abstract

In the past decades of NLP, there has been a steady shift away from rule-based, linguistically motivated modeling towards statistical learning and the induction of unsupervised feature representations. However, natural language components used in today's NLP pipelines are still static in the sense that their statistical model or rule-base is created once, then subsequently applied without further change.

In this talk, I will motivate an adaptive approach to natural language processing, where NLP components get smarter through usage over time, following a 'cognitive computing' approach to natural language processing. With the help of recent research prototypes, three stages of data-driven adaptation will be illustrated: feature/resource induction, induction of processing components and continuous data-driven learning. Finally, I will discuss challenges in the evaluation of adaptive NLP components.

Bio

After obtaining his diploma and PhD in computer science from the University of Leipzig, Chris Biemann spent three years in the search industry at Powerset and Bing in California. Since 2011, Chris is assistant professor for language technology at TU Darmstadt, Germany. Chris and his group regularly organize shared tasks and release NLP software and data under permissive licenses. Chris' research interests include unsupervised lexical acquisition, statistical semantics, cognitive computing and web-scale natural language processing.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
page 1, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

Interlinguas: Deep and Shallow

Aarne Ranta
University of Gothenburg
aarne@chalmers.se

Abstract

In 1629, Descartes proposed a "language of true philosophy" to serve as an interlingua for translating between languages. Over 300 years later, "semantic interlingua" appears on the top of the Vauquois triangle, as the deepest possible analysis guaranteeing the best possible translation. But the main stream of machine translation has considered the interlingua unrealistic and worked on lower levels of the Vauquois triangle, such as syntactic and lexical transfer.

However, the interlingua idea has advantages that do not depend on it being a deep semantic representation. An interlingua makes it possible to build highly multilingual systems without a quadratic blow-up of size. It also enables transfer of information, for instance, from high to low resourced languages; a related idea has recently been exploited in the Universal Dependencies project, which uses a shared set of labels and tags as a cross-lingual representation.

Grammatical Framework (GF) is a formalism that was designed for building interlingua-based multilingual grammars. Its original purpose was to enable special-purpose interlinguas precisely capturing the semantics of different domains, such as mathematics or touristic phrases. However, GF also enables interlinguas that are not so deep. They can be based on surface syntactic structures or just chunks of words. Recent developments of this idea have led to a translation system that currently works for all 182 pairs of 14 languages, ranging from English and German to Finnish and Chinese. This system has a stack of interlinguas, where a semantic layer produces high-quality translations whenever the input can be analysed by it, whereas the syntactic and chunk-based layers guarantee the robustness of the system. The interlingual grammar makes the system very compact in size, so that it can be run off-line on mobile devices.

Bio

Aarne Ranta is Professor of Computer Science at the University of Gothenburg and co-founder and CEO of the start-up company Digital Grammars AB. He defended his PhD in 1990 on the application of constructive type theory to natural language semantics, supervised by Per Martin-Löf. The theory developed in the thesis led to the idea of multilingual grammars, implemented as the system GF (Grammatical Framework) when Ranta worked at Xerox Research Centre Europe in 1997–1999. After Xerox, Ranta has led GF as an open-source project, which to date has had over 150 contributors working on over 30 languages. He has supervised ten PhDs and written three books, of which "Grammatical Framework: Programming with Multilingual Grammars" (CSLI 2011) has also appeared in Chinese. Ranta's vision is to get linguistic knowledge formalized in a precise and efficient way and make it usable in practical applications.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
page 2, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

A Reinforcement Learning Approach for Adaptive Single- and Multi-Document Summarization

Stefan Henß
TU Darmstadt,
Germany
stefan.henss@gmail.com

Margot Mieskes
h_da Darmstadt & AIPHES*
Germany
margot.mieskes@h-da.de

Iryna Gurevych
TU Darmstadt & AIPHES*
Germany
gurevych@ukp.informatik.tu-darmstadt.de

Abstract

Reinforcement Learning (RL) is a generic framework for modeling decision making processes and as such very suited to the task of automatic summarization. In this paper we present a RL method, which takes into account intermediate steps during the creation of a summary. Furthermore, we introduce a new feature set, which describes sentences with respect to already selected sentences. We carry out a range of experiments on various data sets – including several DUC data sets, but also scientific publications and encyclopedic articles. Our results show that our approach a) successfully adapts to data sets from various domains, b) outperforms previous RL-based methods for summarization and state-of-the-art summarization systems in general, and c) can be equally applied to single- and multi-document summarization on various domains and document lengths.

1 Introduction

In the history of research on automatic summarization, only few systems have proven themselves capable of handling different summarization scenarios, domains and summarization needs (e.g. single-document summarization vs. multi-document summarization, summarization of news, e-mails, tweets or meetings). Additionally, they rarely take into account that the human summarization procedure involves *decisions* about keeping and/or deleting information (Friend, 2001).

Therefore, we propose Reinforcement Learning (RL) for the task of summarization to model the decision making process involved in producing an extractive summary, i.e. selecting sentences that make up a summary. In our model, the algorithm decides at each step during this selection process which sentence to choose in order to compile an "optimal" summary. As the definition of optimality depends on various factors such as summarization task, needs, domain etc., RL-based methods are in principle highly adaptive to these factors.

Our major contributions are in introducting a new feature set which makes use of the RL methodology in describing sentences with respect to already selected sentences. Second, we use Q-learning in combination with supervised machine learning instead of TD-learning, to model the effects of adding information with respect to any given quality score or error function. Finally, we evaluate our method on several data sets from various domains such as news, scientific publications and encyclopedic articles. Additionally, we tested our method on single- and multi-document summarization scenarios. We compare our results both to available systems and results published in the literature and show that our proposed method outperforms previous RL methods as well as common summarization methods.

The paper is structured as follows: Section 2 presents background and related work. Section 3 contains details of our RL approach and how it differs from previous RL-based summarization methods. Section 4 describes the evaluation of our methods, which data sets we use and the comparison systems. Section 5 presents the results and a discussion of our findings. Section 6 contains the summary and future work.

2 Foundations and Related Work

The work presented here is based on two research areas: automatic summarization and Reinforcement Learning. As reviewing both in detail is beyond the scope of this article, we would like to point the interested reader to works by Nenkova and McKeown (2011), Mani and Maybury (1999)

*Part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) funded by DFG under grant No. GRK 1994/1.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology, pages 3–12, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

and Mani (2001) inter alia for an overview of the major developments in automatic summarization. For a general introduction to RL, we refer to Sutton and Barto (1998). RL itself has been adopted by the Natural Language Processing (NLP) community for various tasks, among others dialog modeling in Question-Answer-Policies (Misu et al., 2012), for learning dialog management models (Ha et al., 2013), parsing (Zhang and Kwok, 2009) and natural language generation (Dethlefs et al., 2011), which we will not go into details about.

2.1 Reinforcement Learning

RL models contain at least a set of *states* (s_t), possible *actions* (a_t) for each state, and *rewards* (r_t) (or penalties) received for performing actions or reaching certain states. The objective of a RL algorithm is to learn from past observations a *policy* π that seeks desirable states and chooses optimal actions with respect to cumulative future rewards.

Reward Function Rewards or penalties are an important concept in RL, which can be used directly ("online") for example through customer feedback or indirectly ("offline") during training. In many scenarios, collecting the maximum possible immediate rewards at each state (greedy approach) does not yield the best long-term rewards. Optimizing long-term rewards is often solved in RL using *temporal-difference* (TD) learning, where states are valued in terms of their long-term quality, i.e., the maximum sum of rewards one can collect from them. The value of a state s_t can be expressed as follows:

$$V(s_t) = r_t + \mathbb{E}\left[\sum_{i=t+1}^{n} r_i|\pi^*\right] = r_t + \max_{s_{t+1}} V(s_{t+1}) \quad (1)$$

That is, the value of a state (s_t) equals the immediate reward r_t plus the expected maximum sum of future rewards following an optimal policy π^* from s_t on. This equals the immediate reward r_t plus the maximum value of any possible next state s_{t+1}. Including expected future rewards also allows providing rewards for finals states s_n only (e.g., rating the final summary). These rewards are thus passed through to a function $V(s_t)$.

With large state spaces, V has to be approximated using features of s_t: $\hat{V}(s_t) \approx V(s_t)$, as due to the recursion $V(s_{t+1})$ when calculating $V(s_t)$, computing an exact $V(s_t)$ for each s_t is unfeasible, as one would have to consider all possible paths $s_{t+1}, ..., s_n$ through states following s_t. Finding an approximation $\hat{V}$ can be achieved through various training algorithms, such as $TD(\lambda)$ (Sutton and Barto, 1998). Given any $\hat{V}$, defining a policy π is straight-forward: At each state s_t, perform the action that yields the maximum (estimated) next-state value $\hat{V}(s_{t+1})$.

Q Learning models the value $Q(s_t, a_t)$ of performing an action a_t in the current state s_t, instead of estimating the value of each possible next state. Facing the large state space of all pairs (s_t, a_t), Q values are also typically not computed exactly for each possible pair individually, but approximated using features of s_t and a_t. As one knows which state s_{t+1} an action a_t leads to in a deterministic environment, the value of *leading* to s_{t+1} is equivalent to the value of *being* at s_{t+1}. Otherwise, Q learning is based on optimizing cumulative future rewards, and the definition of an optimal $Q(s_t, a_t)$ reflects the value of a state-action pair.

2.2 RL in Automatic Summarization

To our knowledge, Ryang and Abekawa (2012) have so far been the first ones who employed RL for the task of summarization. The authors consider the extractive summarization task as a *search* problem, finding the textual units to extract for the summary, where the "final result of evaluation [...] is not available until it finishes" (Ryang and Abekawa, 2012, p. 257). In their framework, a *state* is a subset of sentences and *actions* are transitions from one state to the next. *Rewards* are given "if and only if the executed action is `Finish` and the summary length is appropriate" (Ryang and Abekawa, 2012, p. 259). Otherwise a penalty (i.e. a negative reward) is awarded. Therefore, they only consider the final score of the whole summary. They define the optimal policy as a conditional distribution of an action with regards to the state and the rewards. For learning, they use $TD(\lambda)$. The method was evaluated using the DUC2004 data set (see Section 4 below), and for each cluster, an individual policy was derived.

Recently, Rioux et al. (2014) extended this approach, also using TD. As features, they used bi-grams instead of $tf * idf$ values and employed `ROUGE` (Lin, 2004b) as part of their reward function. Their evaluation was carried out on the DUC2004 and 2006 general and topic-based multi document summarization and showed that they significantly outperformed previous approaches.

3 Our Method for RL-based Summarization

Similar to R&A(2012), we model each summarization state s_t as a subset of sentences (i.e. a poten-

tially incomplete summary) from the source document(s) to be summarized. For any state s_t, there exists a set of possible actions $\mathscr{A}_s$ to proceed. For us, those are *select* actions for all remaining candidate sentences $c \in D \setminus S$, whose selection would not violate a length threshold LC:

$$\mathscr{A}_s = \{c \,|\, c \in D \setminus S, length(\{c\} \cup S) \leq LC\} \quad (2)$$

There are three fundamental differences between our approach and the approach proposed by R&A(2012): First, we define the *reward* function differently. We use rewards during training, based on the reference summaries available. R&A(2012) did not use reference summaries for their rewards, but only define an intrinsic reward function as their focus is on finding an optimal summary with respect to a fixed quality model. We focus on learning selection policies for optimal summaries from external feedback during a training phase. The formal details of this are given below.

The second difference lies in using *Q learning*. This helps us in determining the value of the partial summary s_{t+1} and the value of adding sentence a_t to *state* s_t. The formal details of this will be presented below.

Finally, our method learns one global policy for a specific summarization task, instead of one policy for each document cluster as in R&A(2012).

Reward Functions During training, we give rewards to a specific *action* by comparing the resulting *state* to an expected outcome (e.g. given through reference summaries). In the case of summarization, the *state* is a summary which can still be incomplete and the *action* is the addition of a sentence to this summary.

From our experiments, we found that the increase of the partial summary's evaluation score is a good training feedback for a sentence addition, which is reflected in the equation below:

$$r_t = score(s_{t+1}; H_D) - score(s_t; H_D) \quad (3)$$

In principle, any scoring function for rating the quality of the summary is applicable, thus allowing a flexible adaptation to different summarization objectives and quality criteria. In our evaluation, we use ROUGE (Lin, 2004b) to rate each summary with respect to the corresponding human reference summaries H_D (see Section 4 for details).

Q Learning Previous approaches to RL-based summarization used TD learning. But despite many recent variations of TD learning (see Section 2.1) with linear approximation, for example by Sutton et al. (2009), issues remain in their application for complex tasks such as summarization. First, especially when not using feature transformations like kernel methods, linear models may lack the power of approximating state values precisely. Second, we only know the latest model coefficients, but lack records of past observations – i.e., specific (s_t, a_t) and their rewards – that may be leveraged by more advanced learning methods to discover complex patterns.

Therefore, we use reward functions that depend on human summaries H_D, during a dedicated training phase, i.e., learning an approximation of $Q(s_t, a_t)$. During training, we create summaries, compare them with given H_D and compute rewards as shown above. Finally, we use those rewards in a Q learning algorithm. This is different to R&A(2012) who do not use reference summaries in learning their reward function and thus do not make use of the available, separate training data for learning the state values $\hat{V}(s_t)$. By using H_D, our approach has more capabilities of adopting features of a specific data set by receiving rewards aligned with the training data and evaluation metrics.

As stated earlier, *Q learning* allows us to model the value of the next state s_t after performing action a_t. Q values are typically learned through updates, where the old model is changed according to the difference between the expected $Q(s_t, a_t)$ and its recalculation based on the reward r_{t+1} just received:

$$\begin{aligned} Q(s_t, a_t) \leftarrow\ & Q(s_t, a_t) \\ & + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \end{aligned} \quad (4)$$

The difference in Q is added to the old value with a scaling factor α (the *learning rate*). The discount factor γ emphasizes short-term rewards (see also Table 1). Using approximations of $Q(s_t, a_t)$, this typically means updating the global coefficients used for the linear combination of features of any pair (s_t, a_t), such as in the gradient descent algorithm (Sutton and Barto, 1998).

We learn our policy on a fixed number of training summaries (so-called *episodes*). In case of less training summaries than episodes desired, summaries can be used multiple times. As the observations made from a training summary depend on the strategy learned so far, re-visiting summaries can yield new information each time they are used. During those episodes, a limited number of pairs of (s_t, a_t) are observed, and statistical models based on features of those pairs may suffer from insufficient observations. For example, there may have

been few examples of selecting short sentences during training and any correlation between sentence length and summary quality thus may be insignificant. We therefore consider a trade-off between following the most promising actions and exploring seemingly bad decisions that have rarely been made so far. The former strategy repeatedly performs similar actions to learn to better distinguish between the most promising actions, while the latter accounts for wrong estimates by performing "bad" actions and updating the model accordingly if they prove to be rewarding instead. Therefore, during training, we use an ε-greedy strategy, which sometimes selects a random action rather than the most promising one. This is shown in the equation below,

$$\pi_\varepsilon(s_t) = \begin{cases} \arg\max_{a_t} \hat{Q}(s_t, a_t), & x \sim [0,1] \geq \varepsilon^{ep} \\ a_{t+1} \sim \mathscr{A}_t, & else \end{cases} \quad (5)$$

where, ep denotes the number of training episodes, i.e. for $\varepsilon < 1$, selecting the most promising action over a random selection becomes more likely with more training episodes. Using 1,000 training episodes, we chose $\varepsilon = 0.999$, i.e., for the first episode the selection is purely random, but during the second half of the training, we only follow the best strategy for optimizing the model coefficients along those decisions. Once training is completed, our policy is to always choose the action a_t with the highest corresponding $\hat{Q}(s_t, a_t)$, resulting in one policy for the whole task/data set.

To summarize, during training we collect the features of pairs (s_t, a_t) and their corresponding $\hat{Q}$ values at the time after observing r_{t+1}. Knowing the following state s_{t+1}, we not only use features of (s_t, a_t) but also include features of (s_t, a_t, s_{t+1}). We can then use any supervised machine learning algorithm to learn correlations between those triples and corresponding $\hat{Q}$ values. In our observations, this allows for more precise estimates of Q. The supervised machine learning algorithm in our system is a *gradient boosting model* (Friedman, 2002), where Q is updated every 500 actions during our training phase, using the samples of (s_t, a_t, s_{t+1}) and corresponding $\hat{Q}$ as described. With several thousand actions during training, this update rate is sufficient and allows for more complex models that would take too much time with more frequent updates. Gradient boosting iteratively reduces the error of simple regression trees by training a new tree predicting the previous' error. Thereby, our method is able to capture non-linear feature interactions and it is not prone to overfitting, due to the discretization in the basic regression trees and optimization parameters, such as maximum tree depth.

Algorithm 1 Learning Q

```
samples ← ∅
for i = 1 to episodes do
    ep ← i mod |training summaries|
    t ← 0, s_t ← ∅
    while length(s_t) ≤ LC, A_{s_t,ep} ≠ ∅ do
        if x ~ U(0,1) < 1 − ε^i then
            a_t ← arg max_{a∈A_{s_t,ep}} Q̂(s_t, a)
        else
            a_t ~ A_{ep,s_t}
        end if
        s_{t+1} ← s_t ∪ {a_t}
        r_t ← reward(s_t, a_t, s_{t+1}; H_ep)
        R_t ← r_t + γ max_{a∈A_{ep,s_{t+1}}} Q̂(s_{t+1}, a)
        samples ← samples ∪ {((s_t, a_t, s_{t+1}), R_t)}
        if |samples| mod 500 = 0 then
            Q̂ ← learn-gradient-boosting-model(samples)
        end if
        t ← t + 1
    end while
end for
```

Our algorithm for learning the RL policy is shown in Algorithm 1. The regression, which predicts features for states and actions, we use gradient boosting as described in Friedman (1999).

Finally, once the training phase is completed, we use the latest gradient boosting model of $\hat{Q}$ to define our policy, i.e., always selecting the most promising actions in its application.

4 Experimental Setup

In this section we describe the data sets, system configuration and evaluation method we used to assess the quality of our algorithm.

Data sets In order to evaluate our method and to compare it to the results published by R&A(2012), we use the DUC2004[1] data set. Additionally, we use the DUC2001 and DUC2002 data sets, as they have been frequently used in the past as evaluation data sets. These also offer the advantage, that they do not only contain multi-document summarization (MDS) tasks, but also single-document summarization (SDS), which allows us to prove the applicability of our proposed method also to SDS. Using the standard training-/test-set splits provided by NIST, we are able to compare our results to those published in the literature.

But as these three data sets entirely consist of news texts, we decided to add other genres as

[1] for all DUC related information see `http://duc.nist.gov/`

well. Two less explored data sets are the ACL-Anthology Reference Corpus (ACL-ARC)[2] (Bird et al., 2008), which contains scientific documents from the NLP domain and Wikipedia[3] (Kubina et al., 2013), which contains encyclopedic documents from a wide range of domains. Both are used in a single document summarization task. Additionally, both the documents and the data sets themselves are considerably larger than the DUC data sets. These data sets allow us to test our method on a range of genres and domains and on considerably larger documents and data sets.

For the DUC data sets several manual summaries are available for the evaluation. For the ACL-ARC we use the abstracts as reference summaries, as it has been done in the past by e.g. Ceylan et al. (2010). Whereas for the Wikipedia, the first paragraph can be regarded as a reference summary, as it has been done by e.g. Kubina et al. (2013). The target lengths for the DUC summarization scenarios are taken from the respective guidelines[4]. The target lengths for ACL and Wikipedia have been determined through the average length of the reference summaries.

System Setup Our method uses several parameters which have to be set prior to training. Table 1 lists these and the settings we used. The main difference between the setup for the DUC and the ACL/Wikipedia-Data is the number of *boosting iterations* (400 vs. 800) and the maximal tree depth (16 vs. 10), which is due to the length differences in the three document sets.

Parameter	DUC	ACL/Wiki
Training episodes	1200	1200
Discount factor	0.01	0.01
ε-greedy	$0.999^{episode}$	$0.999^{episode}$
Boosting iterations	400	800
Shrinkage	0.04	0.04
Max. tree depth	16	10
Min. leaf observations	50	50

Table 1: Experimentally determined parameters used during training and evaluation.

We determined the settings for the listed parameters experimentally. Our aim was to avoid overfitting, while still training predictive models in reasonable time. The parameter settings in Table 1 were found to give the best performance.

[2] http://acl-arc.comp.nus.edu.sg/

[3] http://goo.gl/ySgOS based on (Kubina et al., 2013)

[4] http://www-nlpir.nist.gov/projects/duc/guidelines.html

The individual parameters influence various aspects of the training. The more *training episodes* used, the better the results were. But the number of episodes had to be balanced against overfitting caused by the other parameters. The *Discount factor* weights the contribution of a specific reward once an action has been performed. A too high factor can lead to overfitting. The *ε-greedy* parameter guides how likely it is, that a random action is performed, as this can potentially also lead to an optimal result and is therefore worth exploring. During training, the likelihood of choosing a random action is decreased and the likelihood of choosing an optimal action is increased. The *boosting iterations* guide the training for the gradient boosting. Here, it is crucial to find the balance between good results and computing time, as each training iteration is very time-consuming. *Shrinkage* is similar to the learning rate in other learning methods. We had to balance this parameter between good results and time. The smaller this value is set, the longer each iteration takes and accordingly the training. *Max. tree depth* refers to the size of the regression trees trained by the gradient boosting method. Small trees can hardly generalize, whereas big trees tend to overfit on the training data. *Min. leaf observations* also refers to the regression trees. If the leafs are based on too few training observations, the resulting rules might be based on random observations or overfit on too few observations.

Features The features we use can be grouped into three categories: basic features, linguistic and information retrieval (IR) based features and RL-specific features, which we describe in detail below. The three lists presented here make up the whole set of features used in this work.

Basic and IR-based features The group of basic and IR-based features contains features that are generally used in a wide variety of NLP-tasks, such as text classification (see for example (Manning and Raghavan, 2009, Chp. 13)). They capture surface characteristics of documents, sentences and words, such as the number of tokens, the position of a sentence in a document and the relation between the number of characters and the number of tokens. In addition to the already mentioned surface features, we make use of the ratio for example of the numbers of characters per token. We take into account the stop words in a sentence and the number of stop words in relation to tokens. These features focus on describing the elements of a sin-

Basic/Surface Features	Linguistics and IR-based Features
# of tokens in sentence	mean/max/sum of the sentence's stop word-filtered tokens
# of characters in sentence	total/relative term frequencies (tf) in the source document(s) (docs)
# of characters per #tokens	mean tf compared to the entire corpus, using stemming and $tf*idf$
# of upper case characters per #tokens	the sentence's mean/min/max cosine similarity (cs) compared to all other sentences in the docs, stemmed, stop words filtered, bi-grams
absolute position of sentence	cs between the $tf*idf$ of the sentence and the combined source docs' $tf*idf$
relative position of sentence	mean/max/min cs compared to the sentence's tf vector with those of each source doc
distance of sentence from end	readability score of the sentence
# of chars in sentence before/after	mean/total information content of the tokens (Resnik, 1995)
total # of stop words in sentence	
# of stop words per # of tokens	

Table 2: Basic and commonly used features to describe candidate documents, sentences and words in isolation.

gle sentence or token viewed in isolation.

The surface features only describe sentences or words in the context of the local sentence. We use a set of similar features to describe words and sentences in relation to the whole document. Additionally, we make use of standard linguistic and IR-based features. These features characterize a sentence in terms of the accumulated $tf*idf$ values compared to the document or the document cluster. Other, more linguistically oriented features are based on the cosine similarity between a sentence and all other sentences in the document. Finally, we make use of higher level analysis, such as the readability score (Flesch, 1948; Kincaid et al., 1975). Table 2 shows the full list of basic features (right side) and IR-based features (left side).

RL-based features The third group of features makes use of the specific characteristics of RL and are to our knowledge new to the area of machine learning based summarization. The previous two feature groups describe words and sentences in their local context or in relation to the document they occur in. The RL-based features describe a sentence in the context of the previously selected sentences and how adding this sentence changes the current, hypothetical summary. We also use surface features, such as the number of characters or tokens after the candidate sentence has been added to the already selected sentences. We consider the cosine similarity between the candidate sentence and the sentences selected so far as well. Additionally, we determine the ROUGE scores of the hypothetical summary and use the difference between the summary with and without the candidate sentence as a feature. This is based on the definition of "optimality" we use in this work (see also Section 1 above). Using ROUGE as part of the features is not problematic in this case, as we use explicit training data to train our reward function, which is then applied to the testing data. The splits are based on the NIST training- and test-sets for the DUC data. The ACL-ARC and Wikipedia data are sufficiently large to be split into two different sets: 5506 for training, 614 for testing for ACL-ARC and 1936 for training, 900 for testing for Wikipedia.

Baselines and Reference Systems We use various baselines and references: First, we use standard baselines such as HEAD and RANDOM to produce summaries of the data. Second, we use figures reported in the literature. Finally, we make use of available summarization algorithm implementations such as MEAD, SVM and SUMY[5] to produce summaries of the data. SUMY contains implementations of several well-known summarization methods, among them the algorithm described by Luhn (1958) (Luhn (sumy)), the LSA-based summarization method described by Gong and Liu (2001) (LSA (sumy)), the LexRank algorithm (Erkan and Radev, 2004) (LexRank (sumy)) and the TextRank algorithm (Mihalcea and Tarau, 2005) (TextRank (sumy)). This is especially useful for those data sets that have not yet been extensively used, such as the ACL-ARC and the Wikipedia.

In order to test the contribution of our features and the RL methodology, we used the RL methodology with the individual feature groups. *RL-basic* uses the surface features, *RL-advanced* uses the IR-based features, *RL-non-RL* uses both groups and *RL-RL* uses the RL methodology with the RL features only. Additionally, we implement a Learning-to-Rank (L2R) algorithm to examine the performance of our features, regardless of the RL methodology and use a standard regression-based learning as implemented in WEKA[6].

Evaluation We use the ROUGE framework, which is a standard automatic evaluation metric and which allows for comparison be-

[5] https://github.com/miso-belica/sumy

[6] http://www.cs.waikato.ac.nz/ml/weka/

- new total length in characters and tokens when adding the sentence associated with an RL action
- partial summaries before and after adding a sentence are compared to each source document using ROUGE precision and recall, and cosine similarity; we add features for the mean/min/max/summed differences between both summaries
- mean/min/max cosine similarities between the new sentence and each sentence already included in the summary

Table 3: Reinforcement learning specific features to reflect changes during the creation of the summary.

tween previously reported results and ours. We use ROUGE with the following parameters: `-n 4 -m -c 95 -r 1000 -f -A -p 0.5 -t 0 -w 1.2 -2`. Changes for the length constraint were made for DUC 2004 as required (`-b 665` vs. `-l 100`) in the guidelines[7]. For the ACL data, we used the target length of 100 words (`-l 100`), whereas for the Wikipedia data, we used a target length of 290 words (`-l 290`), to reflect the average summary length.

5 Results and Discussion

Our results are indicated with *RL-full*, which is the RL method using the full feature set. Additionally, we use *L2R*, which is the learning-to-rank method, using the non-RL features and *Regression*, which is a standard regression method using the non-RL features. We also determined the benefit of individual feature groups, such as using the RL-method only in combination with the surface features (RL-Surface), the IR- and linguistic based features (RL-Basic) or only the RL-specific features (RL-RL).

Previous RL-based summarization methods were evaluated on the DUC 2004 data set. Table 4 shows the previously reported results compared to our methods. As can be seen, our method clearly outperforms previously published results on R-1. Rioux et al. (2014) achieved a higher R-2 score. This is based on our choice of R-1 as the optimality score, which was based on the correlation between human scores and R-1 (Lin, 2004a).

Rouge	R&A(2012)	R(2014)	RL-full
R-1	0.3901	0.4034	**0.4042**
R-2	0.0948	**0.1140**	0.1012

Table 4: Results for the Multi-Document Scenario based on the DUC 2004 data set, compared to previously reported results.

Table 5 shows the results on the other two MDS tasks (DUC 2001 and 2002), compared to the best result in the literature and the best baseline system. On the DUC2002 data set, the Luhn(sumy) baseline performs better on R-1 than our method. On DUC2001 and R-2 in general, our method gives the best performance.

[7] http://duc.nist.gov/duc2004/tasks.html

Year	System	R-1	R-2
2001	Manna et al. (2012)	0.3306	
	Luhn(sumy)	0.3218	0.0454
	RL-full	**0.3387**	**0.0740**
2002	Manna et al. (2012)	0.3371	
	Luhn(sumy)	**0.3706**	0.0741
	RL-full	0.3660	**0.0810**

Table 5: Results on DUC 2001 and 2002 Multi-Document Summarization Task.

In order to show that our method is also applicable to single document summarization and can also handle larger document collections and longer documents, we also applied our method to SDS tasks of DUC2001 and 2002, ACL and Wikipedia. Table 6 shows our results in comparison to baseline methods. All results show that the full RL setup is superior to other methods, including the TextRank implementation. On DUC 2001, we found a reported R-2 value of 0.204 by Ouyang et al. (2010). The feature analysis shows that for ACL-ARC and Wikipedia the results of the different feature setups and regression learning methods are significantly worse than the full RL setup.

Error Analysis We observed a range of error sources: *First*, manual inspection of the summaries revealed that the automatic summaries could serve as a valid summary, but the overlap between the automatic and the reference summaries are very small. For example in the document on "Superman" from the Wikipedia data (document ID d34b0d339f3f88fe15a8baa17c9c5048), the RL-based summary contained more information about the character and in-world events, whereas the reference summary contained more information about real-world development.

The *second* problem is the too narrow focus and too few details of our summaries. Considering the cluster on the Hurricane Mitch (D30002, DUC2004), we observed that our summary focuses exclusively on the events regarding Honduras and does neither mention the events on the other islands nor the international call for aid.

	DUC 2001		DUC 2002		ACL		Wiki	
System	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
TextRank(sumy)	0.4450	0.1866	0.4799	0.2240	0.3739	0.0844	0.4625	0.1256
L2R	0.4490	0.1934	0.4770	0.2181	0.3966	0.1052	0.4706	0.1276
Regression	0.4572	0.1942	0.4847	0.2187	0.3899	0.0883	0.4768	0.1261
RL-surface	0.4384	0.1849	0.4684	0.2130	0.3765	0.0875	0.4542	0.1086
RL-Basic	0.4264	0.1657	0.4539	0.1926	0.3693	0.0782	0.4645	0.1196
RL-RL	0.4005	0.1377	0.4350	0.1700	0.3325	0.0542	0.4721	0.1211
RL-full	**0.4584**	0.1993	**0.4862**	**0.2252**	**0.4117**	**0.1102**	**0.4850**	**0.1321**

Table 6: Results on the Single-Document-Summarization Scenario based on DUC, ACL and Wikipedia data sets, compared to standard methods used in automatic summarization.

Third, we observe that temporal information, dates and numerical facts in general were rare in our summaries (for example in the cluster on the North Korean famine (D30017, DUC2004)). Where numbers are included, we find that they are mentioned in different formats, as opposed to the reference, which makes it hard for ROUGE to spot them. One example is from D30017, DUC2004, where the references state that "Two thirds of children under age 7 ...", whereas our summary contains "Two thirds of children under age seven ...".

Fourth, we notice that on the ACL-ARC data very often rows and columns of numbers are extracted, which represent results. While to some extent this is valid in a summary, adding whole tables is not beneficial. Work on translating figures and tables into text has been carried out in the past, but is still an ongoing research topic (see for example (Govindaraju et al., 2013)).

Fifth, we observe that the RL summarizer picked direct speech for the summaries, which did not provide additional information, whereas, direct speech rarely occurs in the references. Detecting direct speech is also its own research topic (see for example (Pareti et al., 2013)).

Finally, we notice that our method extracts considerably longer sentences from the sources, than are those contained in the reference summaries. This problem could be reduced by adding sentence compression to the whole setup.

6 Conclusion and Future Work

In this work, we presented our method for extractive summarization based on RL. We made use of exemplary summaries in the training phase, improved on the learning algorithm through immediate RL rewards and modeling features of states *and* actions, proposed a new, memory-based Q learning algorithm, and used non-linear approximation models. Our method produced global policies for each summarization scenario, rather than a local policy for individual clusters. Finally, we introduced a novel feature set, which exploits the capabilities of reinforcement learning to take into account intermediate results in order to determine the next optimal step. We showed that our system outperforms state-of-the-art methods both on single- and multi-document summarization tasks. Through several, systematic experiments, we showed that the combination of the RL method and the features we employed considerably outperform comparison systems and comparable system setups. Additionally, our method can be adapted to various summarization tasks, such as single- and multi-document summarization, but also to other data sets, such as scientific and encyclopedic articles.

As our error analysis in Section 5 shows, there is room for further improvement on various aspects. Some of these refer to other research topics – such as textually describing tables and figures and detecting direct speech. But some aspects will be tackled in the future: First, reducing the sentence length by applying sentence compression methods. This would allow us to add more information to the summary without violating the length constraint, since we can include more shorter sentences describing various aspects of the summarized topic. The problem of different formats of numbers and abbreviations could be addressed through a normalization step before evaluating. In general, names of persons, places and organizations could be given more importance through Named Entity Recognition features.

Finally, we would like to test our method in other summarization scenarios, such as query-based summarization or data sets such as Twitter.

Acknowledgements

This work has been supported by the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1.

References

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC),* Marrakech, Morocco, 26 May – 1 June 2008.

Hakan Ceylan, Rada Mihalcea, Umut Özertem, Elena Lloret, and Manuel Palo. 2010. Quantifying the limits and success of extractive summarization systems across domains. In *Human Lanugage Technlogies: The 2010 Annual Conference of the North American Chapter of the ACL,* Los Angeles, California, June 2010, pages 903–911.

Nina Dethlefs, Heriberto Cuyahuitl, and Jette Viethen. 2011. Optimising natural language generation decision making for situated dialogue. In *Proceedings of the 12th SIGdial Workshop on Discourse and Dialogue,* Portland, Oregon, 17-18 June 2011.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Rudolf Flesch. 1948. A new readability yardstick. *The Journal of applied psychology*, 32(3):221–233.

Jerome H. Friedman. 1999. Stochastic gradient boosting. `http://astro.temple.edu/~msobel/courses_files/StochasticBoosting%28gradient%29.pdf`, March.

Jerome H Friedman. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

Rosalie Friend. 2001. Effects of Strategy Instruction on Summary Writing of College Students. *Contemporary Educational Psychology*, 26(1):3–24.

Yihong Gong and Xin Liu. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and development in information retrieval (SIGIR-01)*, pages 19–25.

Vidhya Govindaraju, Ce Zhang, and Christopher Ré. 2013. Understanding tables in context using standard NLP toolkits. In *Proceedings of the 51st Conference of the Association for Computational Linguistics* Sofia, Bulgaria 4–9 August 2013, pages 658–664.

Eun Young Ha, Christopher M. Mitchell, Kristy Elizabeth Boyer, and James C. Lester. 2013. Learning dialogue management models for task-oriented dialogue with parallel dialogue and task streams. In *Proceedings of the 14th SIGdial Workshop on Discourse and Dialogue,* Metz, France, 22-24 August 2013.

Peter Kincaid, Robert Fishburne Jr, Richard Rogers, and Brad Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document.

Jeff Kubina, John Conroy, and Judith Schlesinger. 2013. ACL 2013 MultiLing Pilot Overview. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 29–38, Sofia, Bulgaria. Association for Computational Linguistics.

Chin-Yew Lin. 2004a. Looking for a few good metrics: Automatic summarization evaluation – how many samples are enough? In *Proceedings of NTCIR Workshop 4, Tokyo, Japan, June 2-4, 2004.*

Chin-Yew Lin. 2004b. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out at ACL 2004,* Barcelona, Spain, 25–26 July, 2006, pages 74–81.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Inderjeet Mani and Mark T. Maybury, editors. 1999. *Advances in Automatic Text Summarization*. Cambridge/MA, London/England: MIT Press.

Inderjeet Mani. 2001. *Automatic Summarization*. Number 3 in Natural Language Processing (NLP). John Benjamins Publishing Company, P.O Box 36224, 1020 Amsterdam, The Netherlands.

Sukanya Manna, Byron J. Gao, and Reed Coke. 2012. A subjective logic framework for multi-document summarization. In *Proceedings of the 24th International Conference on Computational Linguistics,* Mumbay, India, December, 2012, pages 797–808.

Christopher D. Manning and Prabhakar Raghavan. 2009. *An Introduction to Information Retrieval*. Cambridge University Press.

Rada Mihalcea and Paul Tarau. 2005. A language independent algorithm for single and multiple document summarization. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing,* Jeju Island, South Korea, 11–13 October 2005, pages 19–24.

Teruhisa Misu, Kallirroi Georgila, Anton Leuski, and David Traum. 2012. Reinforcement learning of question-answering dialogue policies for virtual museum guides. In *Proceedings of the 13th SIGdial Workshop on Discourse and Dialogue,* Seoul, South Korea, 05-06 July 2012.

Ani Nenkova and Kathleen McKeown. 2011. *Automatic Summarization*. Foundations and Trends in Information Retrieval. Now Publishers Inc.

You Ouyang, Wenjie Li, Qin Lu, and Renxian Zhang. 2010. A study on position information in document summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING),* Beijing, China, 23–27 August 2010, pages 919–927.

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* Seattle, Washington, USA, October 2013, pages 989–999.

Philip Resnik. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada*, pages 448–453.

Cody Rioux, Sadid A. Hasan, and Yllias Chali. 2014. Fear the reaper: A system for automatic multi-document summarization with reinforcement learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), October 25-29, 2014, Doha, Qatar.*, pages 681–690.

Seonggi Ryang and Takeshi Abekawa. 2012. Framework of automatic text summarization using reinforcement learning. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* Seattle, Washington, USA, October 2013, pages 256–265.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement Learning: An Introduction*, volume 1. Cambridge Univ Press.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. 2009. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Canada, June 14-18, 2009*, pages 993–1000. ACM.

Lidan Zhang and Chan Kwok. 2009. Dependency parsing with energy-based reinforcement learning. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT), Paris, October 2009.*

Did I Really Say That? – Combining Machine Learning & Dependency Relations to Extract Statements from German News Articles

Thomas Bögel **Michael Gertz**
Institute of Computer Science
Heidelberg University
69120 Heidelberg, Germany
{thomas.boegel,gertz}@informatik.uni-heidelberg.de

Abstract

We present a system to extract statements of public figures from unstructured German news articles. We first motivate and define statements as a temporally-aware extension of quotations and present the three categories of statements: (1) direct, (2) indirect, and (3) mixed-style statements. We use a combination of machine learning and heuristics based on dependency parses to tackle all three types of statements. The quality of our extraction approach is compared to related work in quotation attribution showing that rules based on syntactic structures increase the extraction quality compared to lexical patterns. In addition, we apply the system on a corpus of German news articles and show that it is able to extract statements with high precision (82.4%).

1 Introduction

> So betonte Merkel im TV-Duell [...]: "Mit mir wird es eine Maut für Autofahrer im Inland nicht geben."[1]

Statements reveal the general attitude of people and groups towards a topic. In the political context, these "attitudes" are referred to as *policy positions* and represent an important factor in political decision-making processes (e.g., Klüver (2009)) and help voters, for instance, to judge their political alignment with parties and groups. In a world of constantly growing amounts of news in different media, it is very hard to track policy positions of individual people over time. While there are publicly accessible protocols of statements in political debates, statements are also uttered outside of the parliament, and there is no repository of the policies of, for instance, influence groups.

[1] source: http://spon.de/ad4xR

Statements consist of direct quotes that are quite easy to extract from texts but there are also more subtle, indirect utterances. While there are many systems to extract quotations from English texts, there is only very few work on German texts. Furthermore, the existing systems for German only extract quotations, neglecting statements that fall under a more broader category of utterances.

In this paper, we present the first system for extracting a broad variety of statements from unstructured, German news articles at a large scale based on both machine learning and heuristics, as well as syntactic dependency relations (Section 4). In order to take the dynamics of statements into account, we incorporate a temporal dimension into the definition of statements and include it into our extraction process. To our knowledge, this is the first system to apply dependency-based rules to extract temporally-aware statements for German.

Having motivated the need for statements and contrasting statements with the task of quotation attribution in the next section, we present our approach in Section 4 and evaluate the system in Section 5. We compare the extraction quality of our system against lexical patterns presented in related work to answer the question whether using syntactic relationship enhances the accuracy of statement extraction. In addition, we apply our approach to a manually assembled German news corpus and measure precision on the statement level. Finally, we will conclude our findings and present our ongoing work in Section 6.

2 Background and Definition of Statements

2.1 Statements vs. Quotations

The task of *quotation attribution* is to extract quotations of people from unstructured text (e.g., Pouliquen et al. (2007)). Following the definition of Pareti (2012), the task of quotation attri-

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 13–21, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

bution involves three components: extracting the *source*, the *cue*, and the *content*. The *source* represents the person or organisation that a quotation is attributed to, the *cue* is a verb or indicator for a quotation (e.g., "sagte") and *content* contains the actual quotation being uttered. Usually, systems distinguish between direct and indirect quotations and usually cover common verbs indicating a quotation (e.g., "sagte").

With this narrow definition of quotation attribution, statements that should be extracted are missed. Most systems restrict themselves to utterances with a small set of specific verbs that are associated with explicit quotations. The sentence "Angela Merkel kritisierte das EU-Abkommen zur…", for instance, reports about a statement that does actually not directly represent a quotation. Thus, despite being a noteworthy statement for the extraction of policy positions, it would not be extracted by existing systems for German quotation extraction.

(1) (a) Wie eine Sprecherin (am Montag)$_{timestamp}$ sagte, …
(b) Westerwelle kritisiert (NSA-Spähaktion)$_{target}$ …

Another aspect is the inherent temporal dimension of statements: in order to capture the evolution of statements (and thus, policies) over time, it is important to determine when a statement was made. While a common approach like using the publication date of an article might work for many cases, statements can be explicitly time-stamped and thus override the article timestamp as example 1(a) shows. We capture this in our definition of statements as well as during statement extraction. Finally, statements may have an explicit target, meaning they are directed against a specific person, topic (illustrated in example 1(b)) or organization. These targets are valuable for modeling policy positions (Van Atteveldt et al., 2008).

Definition of statements. Taking the above considerations into account, we extend the definition of quotations to develop a broader concept of utterances, called **statements**, as our extraction target. Overall, quotation extraction can be seen as a subpart of statement extraction.

We define a statement as a tuple

$$statement = \langle source, cue, content, target, timestamp \rangle$$

where *source*, *cue*, and *content* are identical to the definition of quotations (see Section 2.1). We add an optional element *target* representing the target of a statement. In addition, each statement is time-stamped. Note that the content can be identical to the cue, as illustrated below.

In the following, we present the differences between three types of statements and discuss how the elements of a statement defined above are realized.

2.1.1 Direct Statements

The typical case for direct statements is a reporting verb (*cue*) and the actual content of the statement in quotes, for example:

(2) "Die Kämpfe dauern unvermindert an", sagt$_{cue}$ (ein Sprecher)$_{source}$.

For direct statements, the *content* is represented by the quoted text passage. In this example, the direct statement is embedded into the sentence and thus the *cue* and *source* are realized within the same sentence. There are also scenarios where the sentence containing the content precedes or succeeds the cue and source. Note that quoted text passages in isolation do not always hint towards a statements but are also used to highlight text passages, for instance. We will deal with these cases in Section 4.1.

2.1.2 Indirect Statements

There are different ways of reporting statements indirectly. We will just present two representative examples showing commonly used sentence structures:

(3) (a) (Angela Merkel)$_{source}$ sagte$_{cue}$, dass dies nicht akzeptabel sei.
(b) (Angela Merkel)$_{source}$ kritisierte$_{cue}$ (die Wahlen in der Ostukraine)$_{target,content}$.

In example 3(a), the content is expressed in a subordinate clause governed by the cue "sagte". The mood of the subordinate clause is subjunctive. The conjunction for the subordinate clause "dass" is optional. Example 3(b) represents a statement that is not covered by a strict definition of quotation but nevertheless expresses an important statement. In this case, the content (i.e., the direct object of the cue) is actually identical to the target of the statement. Thus, a system for statement extraction should not only be able to extract phrasal elements but also complex noun phrases as statement contents.

2.1.3 Mixed Statements

Oftentimes, direct and indirect statements are mixed within the same sentence to combine summaries of statements with direct quotes. There are different variations of mixed statements (e.g., coordinated statements), the following representing a very common pattern:

(4) (Angela Merkel)$_{source}$ bezeichnete$_{cue}$ [(die Wahlen in der Ostukraine)$_{target}$ als "illegitim"]$_{content}$.

As the example shows, the elements of a statement are actually realized similarly to indirect statements. The only difference is that passages in direct speech are embedded into the content. This needs to be taken into account when extracting direct utterances: while the content "illegitim" in isolation is not meaningful by itself, the complete predicative construction is required. Note also that in this case the target is part of the content.

2.1.4 Multi-Sentence Statements

Up until now, we only investigated single sentences as statements. Naturally, a statement can consist of a sequence of sentences. There are again variations of multi-sentence statements. The following example illustrates a frequently occurring pattern:

(5) Angela Merkel bezeichnete die Wahlen in der Ostukraine als "illegitim". *Die Bundesregierung würde diese nicht akzeptieren.*

The sentence in italics in isolation can only be understood as a statement when context – in this case from the previous sentence – is used for interpretation, because the sentence itself does not contain any cue or source. A sentence in subjunctive mood succeeding a sentence containing a statement, however, is a reliable predictor that the statement in the previous sentence is continued.

3 Related Work

As indicated above, the task of statement extraction is highly related to the extraction and attribution of quotations. Thus, we build on similar strategies from related work in this area. While most systems performing quotation extraction focus on languages other than German, some of the fundamental strategies can be applied across different languages as well in an adapted form.

The systems performing quotation extraction can be broadly classified into two categories: rule-based or heuristic and supervised machine learning-based extraction. A comprehensive survey of related work can be found in (O'Keefe, 2014).

Heuristic quotation extraction. Sarmento et al. (2009) propose a simple heuristic system to extract quotes and corresponding sources based on pattern matching of 19 flat patterns from Portuguese news articles. With their simple method that neglects coreference resolution, they are only able to extract quotes for about 5% of all articles. Similarly, Pouliquen et al. (2007) use a small set of simple patterns involving utterance verbs to extract quotations in 11 different languages. Due to the limited number of patterns, they also aim at high precision at the loss of recall. Krestel et al. (2008) employ a two-step approach: after searching for reporting verbs (cues), 6 lexical patterns are applied to extract the content and source of a statement. Their evaluation on just 7 articles from the Wall Street Journal yields a good precision (99%) with reasonable recall (74%).

Overall, while flat, lexical patterns work well for languages with a relatively fixed sentence construction (like English), we will show that a dependency-based approach captures the nature of languages with more variations in sentence construction like German or French in a better way. While de La Clergerie et al. (2011) employ syntactic patterns to extract quotes like we do, they always rely on the complement of an utterance verb to extract the statement content. In contrast, we encode possible realizations of statement elements for each verb individually.

Supervised approaches. Pareti et al. (2013) implement a supervised system for extracting quotes in three steps. First, a classifier predicts whether a verb is a valid cue for a statement based on a set of 20 features. To extract the content of a quote, they employ a token-based as well as a constituent-based classifier. The former performs sequence tagging using Conditional Random Fields to predict for each token whether it is part of a quote. In the constituent-based approach, each syntactic constituent is classified with a MaxEnt classifier, followed by a final post-processing step to unify individual predictions. Their evaluation on the Penn Attribution Relations Corpus (Pareti, 2012) shows that sequence tagging on the token-level performs

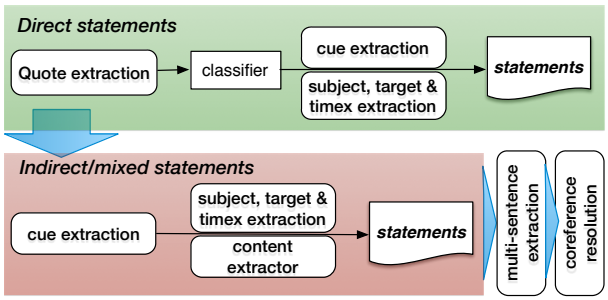


Figure 1: Workflow of statement extraction.

better than classifying constituents or rule-based approaches. The fact that dependency structures yield no improvement might, again, be due to the relatively fixed English sentence structure.

There are only two systems that extract quotes from German texts: as illustrated above, Pouliquen et al. (2007) present a multi-lingual approach suffering from low recall due to the low number of employed patterns. Ploch (2015) have recently presented a system to extract direct, indirect and mixed quotations based on lexical patterns. They only cover a relatively small number (25) of utterance verbs, yielding a low coverage. We, in contrast, follow a broader definition of statement to greatly extend coverage. In addition, instead of rule-based filtering of direct quotations, we use machine learning to filter out misleading quoted text passages. Finally, we employ syntactic dependency relations instead of lexical patterns to extract elements of a statement.

To the best of our knowledge, we present the first system that extracts a broad variety of temporally-aware statements from German texts while explicitly taking into account the target of a statement.

4 Methods

After preprocessing, each document is passed through a sequential pipeline illustrated in Figure 1 that first extracts direct statements using a machine learning-based approach (Section 4.1). For the remaining sentences, indirect and mixed statements are searched (Section 4.2). In a final step, multi-sentence statements are detected and coreference resolution is performed to resolve the sources of statements.

Preprocessing. We implement a modular preprocessing pipeline based on UIMA[2]. After PoS-tagging with the TreeTagger (Schmid, 1999), we perform a morphological analysis using Morphisto (Zielinski and Simon, 2009). To extract the timestamp of a statement, temporal expressions are extracted with HeidelTime (Strötgen and Gertz, 2013). Named Entities are extracted using StanfordNER (Finkel et al., 2005) and a German model (Bingel and Haider, 2014). Finally, the text is parsed with ParZu (Sennrich et al., 2009) to obtain dependency structures and processed with CorZu (Klenner and Tuggener, 2011) to resolve coreferences.

4.1 Direct statement extraction

Extracting direct statements based on quotation marks and heuristics to filter out quoted text passages that do not denote actual statements yields reasonable performance (Pouliquen et al., 2007). There are, however, cases when simple heuristics fail. Many heuristics for filtering quoted strings take into account the length of the text passage. While the quoted string in example (6) consists of only 3 tokens, it is nevertheless a valid statement. Conversely, there are many instances, especially for mixed statements, where longer quoted text passages themselves are only part of a complex statement.

(6) “Deutschland ist Wachstumsmotor”

Instead of manually defining a complex rule set to distinguish between direct statements and highlights etc., we use machine learning to derive rules and thresholds automatically.

As mentioned in Section 2.1.1, the content of direct statements equals text passage enclosed by quotation marks. To extract the remaining elements of a statement, we determine heuristically which sentence reports the statement, i.e., which sentence contains the cue and the other elements of the statement. We first check whether the direct statement is embedded in a sentence.[3] Otherwise, we determine whether the preceding sentence is the reporting sentence by searching for a cue or a colon at the end of the sentence. Finally, the reporting sentence is processed as described in the following section.

[2] https://uima.apache.org/

[3] Quoted passages consisting of multiple sentences are also taken into account.

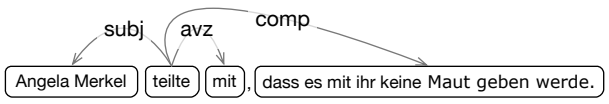


Figure 2: Dependency relations for statement cues.

4.2 Statement Extraction using Dependency Trees

While there are annotated corpora for statement extraction in English (e.g., Pareti (2012)), this is not the case for German. To our knowledge, there exists just one data set from Ploch (2015) that only captures a small amount of statements. Due to the lack of training data and the fact that a rule-based system allows for fine-grained control of encoding context information, we implement a rule-based approach to extract elements of statements. In addition, while we are currently working with news articles from the politics section, switching to another category (such as sports or economy) might require additional statement cues or phrases and thus – in the case of machine learning – additional training data. As our rules are completely independent from the code base, switching to a different domain can easily be done by extending the rule set.

Our approach makes use of dependency relations between the cue verb and other elements of a statement. Figure 2 shows a dependency representation of a sentence containing a statement. The dependency relations governed by the cue (*teilte*) represent the different elements of a statement: the *subject* represents the source and the statement content is realised by the *comp* node.

Syntactic dependency structures present multiple advantages over lexical patterns. In contrast to purely string-based patterns – such as a heuristic regarding everything in a sub-sentence beginning with "dass" as the content of a statement – rules based on dependency relations can be much more fine-grained with respect to elements that should be included in the content string. In addition, syntactic structures are better able to identify the source of a statement by automatically extracting the subject of sentences.

Lexicon of statement cues. To recognize possible cues of statements, we compiled a lexicon consisting of more than 200 verbs indicating a statement by looking at suitable synsets for utterance verbs in GermaNet (Hamp and Feldweg, 1997). In addition to the lemma of each verb, we encoded how elements of statements are realised with respect to dependency relations. Details are explained below.

Extracting the cue. The extraction process begins with extracting the main verb (head) of each sentence. Auxiliary and modal constructions are also taken into account and resolved (e.g., "wollte betonen"). As indicated in Figure 2, particle verbs are also correctly handled. This is necessary to disambiguate verbs that have different meanings with different verb particles. In the example above, "teilen" can only be used as a statement cue in combination with the particle "mit", resulting in "mitteilen". Without proper handling of verb particles, either coverage or quality would suffer due to missed or erroneous statements, respectively. If a match in our verb lexicon is found for the head of the sentence, the remaining elements of the statement are extracted next.

Extracting the source. In most cases, the subject of the cue is identical to the source of a statement, but there are exceptions. In the sentence "Er zitierte Angela Merkel", for instance, the subject of the cue does not actually represent the source. The source is realised as the direct object. These exceptions are encoded in our verb lexicon. Additionally, we normalize passive constructions like "Angela Merkel wurde von Horst Seehofer kritisiert" resulting in the source of a statement being expressed as a prepositional construction. Attributes of the source are not extracted, and if a source partially matches a named entity of type `PERSON` predicted by StanfordNER, we link the statement to the named entity and optionally correct the span of the respective source so that it aligns with the named entity.

Extracting the content. As the content for direct speech statements is already defined by the quoted text passage, content extraction is only performed for indirect statements. For each verb, we encode how the content of an utterance is typically expressed in terms of dependency relations. While the content is oftentimes expressed as a phrasal complement (such as in Figure 2), there are cases where different patterns need to be employed: the verb "bezeichnen", for example, does not take phrasal complements but employs a predicative construction to express the statement content.

Extracting the target and timestamp. Statements often directly refer to a person or a specific topic. Some utterance verbs directly encode the target, such as "kritisieren" or "verlangen". For verbs with an explicit syntactic slot for a target, we encode the corresponding dependency relation in our lexicon. "verlangen", for example, can represent the target of a statement as a prepositional object with the preposition "von".

To extract the timestamp of a statement, we use the output of the temporal tagger HeidelTime: we check whether a temporal expression fills a modifier position of the cue (as in Example 7(a)). In contrast, temporal expressions within the utterance *content* are neglected, as they do not indicate the point in time of a statement (see Example 7(b)). If no timestamp can be found, the timestamp of the article is used.

(7) (a) Wie eine Sprecherin (am Montag)$_{timestamp}$ sagte, ...
(b) Wie eine Sprecherin sagte, war am (am Montag)$_{timestamp}$ ...

Mixed statements. As explained in Section 4.1, quoted strings that are part of a mixed statement should be filtered out by our classifier. As mixed statements are usually governed by the same cues as indirect statements, mixed statements are tackled as indirect statements and the quoted text passage is simply integrated into the content string.

4.3 Multi-sentence Statements and Coreference

After all components have been passed through, we iterate over all sentences to determine whether a sentence should be attached to an already existing statement as explained in Section 2.1.4. If a sentence in subjunctive mood directly succeeds a statement, it is attached to the previous statement. We rely on the morphology component to determine whether a verb is in subjunctive mood, taking into account the subject of the verb for ambiguous verb forms. For continued statements, the cue, source, target and timestamp are often not explicitly mentioned again. In this case, we adopt all four elements from the immediately preceding statement.

Coreference Resolution. The source of statements is often represented indirectly by pronouns or noun references. In order to attribute statements to the correct person, we cluster all mentions of an entity in a text using a modified[4] version of CorZu to perform coreference resolution.

5 Experiments and Evaluation

To evaluate our system, we use two different approaches and data sets: we first assess the quality of extracted quotes by measuring token-based precision and recall of statement components. In addition, we compute the precision of our system with respect to extracted statements.

In the next section, we will first present the two data sets that are used for evaluation, followed by experiments and results for filtering direct quotations in Section 5.2. Finally, we perform a token-based (Section 5.3) and statement-based (Section 5.4) evaluation of our system.

5.1 Data Sets.

Token-based evaluation. In Ploch (2015), two annotators manually annotated the source, cue and the content of quotes in 287 random news articles. For the source, coreference should be resolved and the fully specified name be used as the speaker. The annotation resulted in 383 quotations. We use this data set to compare our approach to the system by Ploch (2015). While the annotations are suitable for evaluating the correctness of *extracted* quotations – i.e., how well the predicted spans match manual annotations – there are two issues: first, only quotations are covered, meaning a sub-set of all statements. Second, due to the focus on token-based accuracy, only a fraction of all quotations are annotated in the data set, meaning it cannot be used to evaluate performance on the statement level.

NSA data set. We manually created an additional data set covering a specific topic – the NSA spying scandal. We chose a well-defined topic to be better able to judge statements in contrast to random news articles. We collected all articles related to the topic between 06-2013 and 12-2014 from two major German news sites, *Spiegel Online*[5] and *FAZ*[6], resulting in about 1200 articles.

5.2 Filtering Direct Quotations

To train a random forests classifier (Breiman, 2001) predicting whether quoted strings are valid statements, we manually annotated 1000 quoted text

[4] We extended the list of proper names and corresponding gender assignments.

[5] www.spiegel.de/

[6] www.faz.net/

	Precision	Recall	F_1
Baseline	71.3	94.7	81.4
Random Forest	96.9	98.0	**97.5**

Table 1: Results of predicting direct statements using machine learning compared to a rule-based baseline approach.

passages from 80 documents. For each text passage, an annotator should decide whether the text passage represents a direct statement. Thus, highlights and misused quoted strings were annotated as erroneous. As mentioned in Section 2.1.3, quoted text passages within mixed statements in isolation are also not sufficient and were thus marked as erroneous, too.

We used a combination of 12 features. Besides the literal string that is quoted, we determine whether (2) the quoted string contains a verb and (3) the preceding sentence ends with a colon. The length of the quoted string is counted in (4) token and (5) characters and both counts are also estimated (6,7) relative to the sentence length. Finally, we check whether the (8) preceding and (9) proceeding sentences also contain a quoted string or an (10,11) utterance verb. A (12) *type* feature determines whether the quotation spans the whole sentence or it is embedded.

Baseline. To check if our machine learning approach performs better than a simple rule set, we compare the trained models to a baseline approach filtering all quoted text passages that consist of at most 3 tokens and do not contain any verb.

Results. To measure the impact of training data size, we first fitted the model on 500 quoted strings and then gradually increased the number to 1000, measuring performance differences with 10-fold cross-validation. For more than 800 instances, the results differed only marginally. The results of 10-fold cross-validation for predicting direct quotations in Table 1 show that the classifier outperforms the baseline by far. The drastic increase in precision of about 15% shows that using simple rules to filter out quoted strings yields many more false positives erroneously being predicted as statements. Overall, with a relatively low number of manual annotations, good results can be achieved.

5.3 Token-Based Evaluation of Quotations

The quality of extracted statements with respect to the number of tokens correctly annotated is summarised in Table 2. We report the numbers of Ploch (2015) alongside our results. For all elements of direct statements, our approach clearly outperforms the baseline system. Regarding indirect and mixed statements, our system improves recall of the cue and source extraction at a slight loss of precision, resulting in a higher F-Score. The performance of content extraction is especially promising with a high increase of precision and recall for all statement types. The F-Score of content extraction for indirect statements, for example, increased by more than 9 percentage points from 76.4% to 85.5%.

Error analysis. Manual error inspection revealed that the loss of precision for the source is often due to erroneous coreference resolution or named entity recognition. It seems that simple rules to resolve pronoun coreference might perform better than applying full-fledged coreference resolution. Errors in the content string are mostly caused by parser errors, especially for complex sentence structures. While errors like these are obviously inherent to a method that relies on syntactic dependencies and could be resolved by a sequence tagging approach, complex cases and sentence structures will probably require many more training data. Another issue are sentence constructions that could be resolved by individual rules but occur very rarely, such as the one given in example (8) where the content of a mixed statement in subjunctive mood precedes the cue and source and the content is referred to by a pronoun.

(8) Dieser Aufdruck sei “kein Wegwerfdatum [...]”. Das sagte [...] Ilse Aigner [...]

Overall, the evaluation showed that an approach based on dependency relations performs better than lexical pattern matching. We believe that this is mostly due to the power of dependency relations being able to capture discontinuities that often occur in German sentences.

5.4 Evaluating Precision of Statement Extraction

As fully annotating a complete data set with statements to measure precision and recall was not feasible, we ran the system on the entire NSA data set described above and focus on evaluating precision of statement extraction. While recall can

		cue			*source*			*content*		
		Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
direct	Ploch (2015)	91.4	67.9	77.9	79.1	67.2	72.7	89.0	89.5	89.2
	our	91.6	86.3	**88.9**	79.1	75.0	**77.0**	96.4	93.4	**94.9**
indirect	Ploch (2015)	98.9	81.5	89.3	85.2	59.6	70.1	74.7	78.2	76.4
	our	90.3	89.9	**90.1**	77.1	76.0	**76.5**	83.2	88.6	**85.8**
mixed	Ploch (2015)	85.2	76.7	80.7	72.2	65.3	68.8	91.3	50.5	65.0
	our	87.1	81.3	**84.1**	75.0	69.1	**71.9**	92.6	77.1	**84.1**

Table 2: Evaluation of token-based precision and recall. The gray entries are the results reported in Ploch (2015).

	direct	indirect	mixed	all
statements	2415	3701	928	7044
persons	43.1%	40.9%	39.8%	41.5%
targets	8.2%	10.7%	9.3%	9.7%
time stamp	14.4%	15.8%	17.0%	15.5%
precision_{50}	97.2%	72.1%	81.8%	82.4%

Table 3: Data set statistics of the entire NSA data set as well as precision of extracted statements for 50 randomly sampled documents.

be compensated with redundancy in the data set, extracting erroneous statements should be avoided. Table 3 shows how many statements were extracted, as well as the fraction of statements for which persons (named entities of type PERSON) as sources, targets and explicit timestamps could be extracted. The numbers support our hypothesis that it is worth integrating the timestamp and target into the extraction process as, for instance, 17% of all mixed statements encode an explicit timestamp.

To measure how many of the extracted statements are actually valid, we randomly sampled 50 documents from the NSA data set and checked how many of the extracted statements represent true statements, regarding overlapping annotations as correct. The resulting precision for each statement type is given in Table 3. Overall, over 82% of all extracted statements are correct, thus showing that our system achieves high precision combined with a higher coverage due to a significantly enhanced lexicon of utterance verbs.

6 Conclusion and Ongoing Work

This paper presented an approach to extract statements from unstructured German news articles. Our temporally-aware definition of statements allows for creating and exploring a timeline of statements over time. We presented a two-stage approach consisting of a machine learning-based system to extract direct statements and a heuristic component based on a large lexicon of utterance verbs and corresponding dependency relations to extract statement components. Comparing our approach to the only other existing system for quotation extraction in German revealed that dependency relations are better suited to extract quotes and statements than lexical patterns. In addition, our extended database of statement cues extends coverage of statements while simultaneously maintaining high precision.

We are currently working on a bootstrap implementation to realize sequence tagging for statement extraction despite the low number of training instances. The final version of our system will be published as an open source project.

Acknowledgments

We would like to thank Danuta Ploch for sharing the annotated data set of quotes with us, making it possible to compare our system to hers. In addition, we thank the anonymous reviewers for their helpful remarks and suggestions.

References

Joachim Bingel and Thomas Haider. 2014. Named Entity Tagging a Very Large Unbalanced Corpus: Training and Evaluating NE Classifiers. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Éric de La Clergerie, Benoît Sagot, Rosa Stern, Pascal Denis, Gaëlle Recourcé, and Victor Mignot. 2011. Extracting and Visualizing Quotations from News Wires. In Zygmunt Vetulani, editor, *Human Language Technology. Challenges for Computer Science and Linguistics*, Lecture Notes in Computer Science, pages 522–532. Springer.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Birgit Hamp and Helmut Feldweg. 1997. Germanet – a Lexical-Semantic Net for German. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

Manfred Klenner and Don Tuggener. 2011. An Incremental Entity-Mention Model for Coreference Resolution with Restrictive Antecedent Accessibility. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2011)*, pages 178–185.

Heike Klüver. 2009. Measuring Interest Group Influence Using Quantitative Text Analysis. *European Union Politics*, 10(4):535–549.

Ralf Krestel, Sabine Bergler, and Ren Witte. 2008. Minding the Source: Automatic Tagging of Reported Speech in Newspaper Articles. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Tim O'Keefe. 2014. *Extracting and Attributing Quotes in Text and Assessing them as Opinions*. Ph.D. thesis, University of Sydney.

Silvia Pareti, Timothy O'Keefe, Ioannis Konstas, James R Curran, and Irena Koprinska. 2013. Automatically Detecting and Attributing Indirect Quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 989–999.

Silvia Pareti. 2012. A Database of Attribution Relations. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*.

Danuta Ploch. 2015. Intelligent News Aggregator for German with Sentiment Analysis. In *Smart Information Systems*, pages 5–46. Springer.

Bruno Pouliquen, Ralf Steinberger, and Clive Best. 2007. Automatic Detection of Quotations in Multilingual News. In *Proceedings of Recent Advances in Natural Language Processing*, pages 487–492.

Luis Sarmento, Sergio Nunes, and E Oliveira. 2009. Automatic Extraction of Quotes and Topics from News Feeds. In *4th Doctoral Symposium on Informatics Engineering*.

Helmut Schmid. 1999. Improvements in Part-of-Speech Tagging with an Application to German. In Susan Armstrong, Kenneth Church, Pierre Isabelle, Sandra Manzi, Evelyne Tzoukermann, and David Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*, volume 11 of *Text, Speech and Language Technology*, pages 13–25. Springer Netherlands.

Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. 2009. A new hybrid dependency parser for german. In *Proceedings of the German Society for Computational Linguistics and Language Technology*, pages 115–124.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298.

Wouter Van Atteveldt, Jan Kleinnijenhuis, and Nel Ruigrok. 2008. Parsing, semantic networks, and political authority using syntactic analysis to extract semantic relations from dutch newspaper articles. *Political Analysis*, 16(4):428–446.

Andrea Zielinski and Christian Simon. 2009. Morphisto – An Open Source Morphological Analyzer for German. In *Proceedings of the 2009 Conference on Finite-State Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP 2008*, pages 224–231.

Fast or Accurate? – A Comparative Evaluation of PoS Tagging Models *

Tobias Horsmann **Nicolai Erbs** **Torsten Zesch**
Language Technology Lab
Department of Computer Science and Applied Cognitive Science
University of Duisburg-Essen, Germany
{tobias.horsmann,nicolai.erbs,torsten.zesch}@uni-due.de

Abstract

We perform a comparison of 22 PoS tagger models for English and German offered by 9 different implementations. By evaluating on a mix of corpora from different domains, we simulate a black-box usage where researchers select a tagger (because of popularity, ease of use, etc.) and apply it to all sorts of text. We find the expected trade-off between fast models with relatively low accuracy and slower models with higher accuracy. The choice of the model, even for the same tagger, does matter and the model should always be chosen for the task at hand. Our evaluation provides researchers with a basis for selecting taggers according to their needs.

1 Introduction

Part-of-Speech (PoS) tagging is one of the most important steps in Natural Language Processing (NLP). Consequently, researchers can choose from a wide range of available PoS taggers, popular choices include TreeTagger (Schmid, 1995), Stanford Tagger (Toutanova et al., 2003), or ClearNLP (Choi and Palmer, 2012). The decision for a certain tool is mainly influenced by tagging accuracy, but other practical issues like ease of use, speed, applicability to target language and domain, or availability for a certain hardware platform might also play a role.

In this paper, we focus on tagging accuracy vs. speed and perform a comparative evaluation of 22 tagging models for English and German, offered by 9 different PoS tagger implementations. We evaluate on a range of English and German corpora from three different broad domains (formal writing, speech transcripts, and social media).

To our knowledge, this is the most comprehensive evaluation to date. Giesbrecht and Evert (2009) compared German models of five PoS taggers and Miguel and Roxas (2007) compared four Tagalog taggers on a single corpus.

PoS tagging A PoS tagger is an application that assigns the word class (i.e. the PoS tag) to each token in a sentence. PoS taggers can loosely be categorized into unsupervised, supervised, and rule-based taggers.

Unsupervised taggers (Goldwater and Griffiths, 2007; Biemann, 2006; Das and Petrov, 2011) analyze large quantities of plain text and group words by their context similarity. The assumption is that words that are grouped together share the same word class. However, this word class is not made explicit in this case, which is why unsupervised taggers are rarely used on their own but usually added as features in a supervised setting (Ritter et al., 2011).

Supervised taggers are machine learning applications that require manually annotated training data. The tagger takes the annotated text and extracts text properties (so called *features*) that are provided to the machine learning classifier which learns a model that maps the feature representation of tokens to the corresponding PoS tags. When running the tagger, the same feature representation is extracted from the raw input text and the trained model is applied to select a tag for every token based on the feature values. A model is thus best applied to input text that is as similar as possible to the training data. In case of a mismatch, e.g. a model trained on newswire applied to speech transcripts, the extracted feature values might not match with the expected ones. As a consequence, the tagging accuracy is considerably reduced.

* An earlier version of this paper used an evaluation subcorpus that turned out to be machine tagged instead of manually labelled. As this artificially increases some results, we decided to remove the problematic corpus. Using the refined evaluation dataset, all general conclusions still hold with one exception: the rule-based tagger does not outperform all other taggers anymore.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 22–30, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

Rule-based taggers utilize sets of patterns or rules to assign tags. In principle, they are very similar to the supervised taggers, only that the underlying model is not automatically learned but hand-curated.

Research question In this paper, we focus on supervised and rule-based taggers, and ask the question: which is the best tagger? However, as we have learned above, supervised taggers are machine learning applications that use a tagging model. Thus, many taggers come with several models that are optimized for different domains or offer trade-offs between accuracy and speed. Thus, the statement *Tagger X performs well* needs to be rephrased as *Tagger X using model Y performs well on corpus Z*.

As the performance of a tagger relies on a complex mix of machine learning, feature representation, and the applied external resources, we cannot analytically decide which tagger is the best. Instead, we perform an empirical evaluation that will provide researchers with a sound basis for their choice of a PoS tagger.

2 Experimental setup

In our experiment, we want to evaluate the tagger models of various PoS tagger implementations against a large number of corpora from various text domains. We base our experiments on the DKPro Core framework (Eckart de Castilho and Gurevych, 2014) that is based on UIMA (Ferrucci and Lally, 2004). DKPro Core provides wrappers for a wide range of taggers shielding the user from the intricate details of installing and invoking the taggers and offering simple, unified usage by providing a shared interface. A UIMA workflow follows a pipeline principle where documents are passed through and processed by an arbitrary number of processing components.

2.1 Processing pipeline

In our setup, each corpus is read and transformed into the internal representation of DKPro Core which is based on stand-off annotations. The tagging is done by a wrapper-component that encapsulates the PoS taggers and allows for using all taggers over a common interface. The wrapper transforms the internal representation of the text into the format which the tagger requires and transforms the tagged text back into the internal representation for further processing. A final evaluation component compares the assigned tags to the gold tags from the corpus.

Directly before and after the tagger component, we inject time measuring components in order to ensure that only the actual time spent for tagging is measured. However, our measuring includes the time that the wrapper needs to feed the data to the underlying tagger implementation. In case of Java taggers, this is usually just a method call, but in case of wrapped C binaries there might be a considerable overhead. Thus, the runtime reported in this study might differ than when running a tagger without the wrapper.

A further issue that might affect the time measurement is document size. Some taggers are fastest when fed with small chunks of data, while others are optimized for processing large documents as a whole. In order to account for this difference, we run all experiments twice: (i) with each sentence as a unit of processing, and (ii) the entire corpus as a unit of processing. We then report the run that takes less time.[1]

2.2 Tagger implementations and models

We now describe the PoS taggers and their models used in this study (see Table 1 for an overview). If available, we provide information about the domain of the training data that were used to train the models.

Arktools (Owoputi et al., 2013) is tailored to tag social media messages. Three models are available of which we use the one trained on annotated Tweets by (Ritter et al., 2011) which uses an extended PTB tagset. The remaining two models are omitted as their training data are part of our evaluation set, a model trained on the data by Gimpel et al. (2011) and IRC chat data by (Forsyth and Martell, 2007);

ClearNLP (Choi and Palmer, 2012) provides a model trained on a mixture of text from various genres that is mostly news-related.

Hepple (Hepple, 2000) is a rule-based tagger similar to the Brill-Tagger (Brill, 1992).

HunPos (Halácsy et al., 2007) is an open-source reimplementation of the TNT tagger (Brants, 2000). Newswire models are available for English trained on the WSJ and for German trained on the Tiger corpus.

LBJ (Roth and Zelenko, 1998) provides a model

[1] Note that the accuracy in both cases is always equal, as the same sentences are tagged.

Tool	Language	Trained on	Modelname	Tagset	Domain	Abbr.
Ark	en	Ritter	ritter	PTB-RIT	social	Ark
ClearNLP	en	OntoNotes	ontonotes	PTB	news	Clear
Hepple	en	*rule-based*		PTB	-	Hepple
HunPos	en	WSJ	wsj	PTB	news	Hun
	de	Tiger	tiger	STTS	news	
Mate	en	CoNLL2009	conll2009	PTB	mixed	Mate
	de	Tiger	tiger	STTS	news	
Lbj	en	WSJ	-	PTB	news	Lbj
OpenNLP	en	*unknown*	maxent	PTB	*unknown*	O-1
		unknown	perceptron	PTB	*unknown*	O-2
	de	Tiger	maxent	STTS	news	O-3
		Tiger	perceptron	STTS	news	O-4
Stanford	en	WSJ	bidirectional-distsim	PTB	news	St-1
		WSJ	caseless-left3w.-distsim	PTB	news	St-2
		unknown	fast	PTB	*unknown*	St-3
		WSJ	wsj-0-18-caseless-left3w.-distsim	PTB	news	St-4
	de	Negra	dewac	STTS	news	St-5
		unknown	fast-caseless	STTS	news	St-6
		Negra	fast	STTS	news	St-7
		Negra	hgc	STTS	news	St-8
TreeTagger	en	*unknown*	le	PTB-TT	news	Tree
	de	*unknown*	le	STTS	news	

Table 1: Tagger models used in our experiments.

for English trained on newswire text.

Mate (Björkelund et al., 2010) provides an English model trained on CoNLL2009 (Hajič et al., 2009) and a German model trained on the Tiger newswire corpus.

OpenNLP is an Apache project that provides a wide range of NLP tools including a tagger.[2] It provides models for English and German based on two different classifiers (Maximum Entropy and Perceptron). The German models are trained on the Tiger corpus. We could not find any information about the training data of the English models.

Stanford (Toutanova et al., 2003) provides several English and German models for their tagger. The models differ with respect to lowercasing of all tokens, adding distributional knowledge, or using a bidirectional model. We excluded two social media models trained by Derczynski et al. (2013) [3] as they use training data which is part of our evaluation set. The origin of some models is unknown.

TreeTagger (Schmid, 1994; Schmid, 1995) provides an English model trained on the Penn-Treebank and further proprietary resources as well as a German model for which little information is available.

2.3 Tagsets

A tagset is a collection of labels which represent word classes. A coarse-grained tagset might only distinguish main word classes such as adjectives or verbs, while more fine-grained tagsets also make distinctions within the broad word classes, e.g. distinguishing between verbs in present and past tense.

Many English models are trained on corpora annotated with the PTB tagset, which distinguishes 48 tags (Marcus et al., 1993). Some models add additional tags to the PTB in order to distinguish further language phenomena. Schmid (1994) assigns the inflection forms of the words *be, do, have* an own tag instead of the default verb tags. Likewise, the word *that* is tagged with an own tag if it occurs as preposition. Ritter et al. (2011) added four additional tags to label the phenomenons that frequently occur in Twitter messages like hashtags or URLs. Forsyth and Martell (2007) prefix PTB tags with an extra character in case the word-form is misspelled.

Other tagsets used in the evaluation corpora are Brown (Nelson Francis and Kuçera, 1964) and C5 (BNC) as well as the coarse-grained Gimpel tagset

[2] https://opennlp.apache.org

[3] https://gate.ac.uk/wiki/twitter-postagger.html

	Domain	Corpus	Tokens in (10^3)	Tagset
en	written	BNC-News	100	C5
		Brown	1,100	Brown
		GUM-News	9	PTB-TT
		GUM-Voyage	9	PTB-TT
		GUM-HowTo	13	PTB-TT
	spoken	BNC-Conversation	100	C5
		GUM-Inverview	13	PTB-TT
		Switchboard	2,100	PTB
	social	Gimpel	27	Gimpel
		NPS-Chat	32	PTB
de	written	Tüba-DZ	1,500	STTS
	social	Twitter-Reh	20	STTS

Table 2: Corpora used in our experiments.

with 25 tags specialized on social media. In German, the *Stuttgart-Tübingen-TagSet* (STTS) with 54 tags is exclusively used.

If a model trained on a corpus with a certain tagset is evaluated on a corpus using a second tagset, this mismatch will result in artificially low accuracy. Thus, we map the fine-grained tags to the coarse grained *universal tagset* (Petrov et al., 2012) as implemented by DKPro Core. Obviously, subtle distinctions between similar tags will be lost in the process, but for many downstream applications fine-grained distinctions between sub-tags of the same word class are not important anyway. Thus, the coarse-grained accuracy gives a good approximation of the expected tagging quality.

2.4 Corpora

Table 2 gives an overview of the corpora used in our evaluation. We partition the English corpora into three broad domains: (i) formal writing, (ii) speech transcripts, and (iii) social media. We choose this partitioning to challenge the taggers with inherent different contents. For German, we could only find corpora for the written and social media domains.

English The first set of corpora contains formal writing, e.g. news articles, travel reports and how to's. We use subset of the newswire text from the British National Corpus[4], the Brown corpus (Nelson Francis and Kuçera, 1964) which contains American English of the 1960's and three subsections of the GUM (Zeldes, 2016) corpus. The second set contains transcripts of spoken language. We use the Switchboard (Marcus et al., 1993) corpus (telephone conversations), a subset of the British National Corpus with spoken language, and one section with interviews taken from the GUM corpus. The third set contains social media messages that combine properties of written and spoken language. Social media is characterized by its high vocabulary heterogeneity and many domain-specific tokens as emoticons, URLs, or email addresses which are likely to be out-of-vocabulary for most tagger models. We use an IRC Chat corpus by Forsyth and Martell (2007) as well as annotated Twitter messages by Gimpel et al. (2011).

In order to avoid testing on the training data, we exclude other available PoS-annotated corpora like the WSJ corpus (Marcus et al., 1993) or the Twitter corpus by Ritter et al. (2011), as many of the models have been trained using those corpora. As the provenance of some models is unknown, their results should be treated with caution as we might still be testing on the training data here.

German We use the STTS-annotated Tüba-DZ corpus (Telljohann et al., 2004) based on the German newspaper *die tageszeitung* and the Twitter-Reh corpus (Rehbein, 2013) of German Tweets annotated with an Twitter-specific extension of STTS following Ritter et al. (2011). We exclude the Tiger corpus (Brants et al., 2004) and the Negra corpus (Skut et al., 1998) as all German models are trained on one of the two.

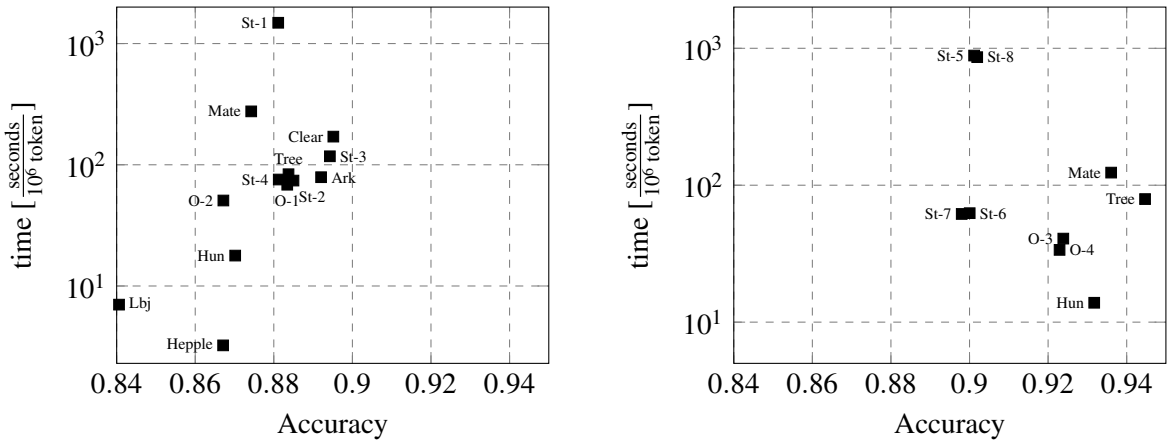
3 Results and Analysis

After evaluating all tagger models on all corpora we obtain the results shown for English in Figure 1a and for German in Figure 1b. The x-axis shows the macro-averaged tagging accuracy based on the coarse-grained universal tagset. As discussed above, we cannot use fine-grained tags for evaluation, because of frequent mismatches between the tagset used by the tagger and the tagset used in the evaluation corpus. The y-axis shows the normalized processing time in seconds per million tokens. Of course the hardware[5] will influence the absolute time spent on the task, but the relative differences between the models are of greater importance here.

In general, we observe the expected trade-off between (i) high-accuracy taggers that invest a lot of processing into feature extraction or more sophisticated classifiers and are thus slower, and (ii) high-speed taggers that can process much more tokens in the same time at the cost of accuracy.

[4] http://www.natcorp.ox.ac.uk/

[5] In our case: Intel Core i5 2.9 GHz CPU, 16GB RAM, single core execution.



(a) English

(b) German

Figure 1: Macro-averaged results over all corpora.

For example on the English corpora, *Hepple* is extremely fast, but reaches only a low accuracy while *St-3* or *Clear* yield a much better accuracy (about 3 points), but are an order of magnitude slower.

On the models that are available for German, we see the same trade-off like for English, with the HunPos tagger being quite fast, but not as accurate as TreeTagger or Mate. Interestingly, none of the Stanford models is competitive for German.

Summarizing the overall results: Even the most accurate English models stay below an accuracy of 90%. While the choice of the model does matter for the accuracy to be expected, the difference in runtime is the most salient difference. As a consequence, researchers need to choose according to their needs. A digital humanities scholar with a couple of hundred documents to tag, may safely select the most accurate tagger, while a social media analyst looking for trends in the full Twitter stream might be better off with one of the faster alternatives.

So far, we have only considered the macro-averaged performance over all corpora. This simulates the usage scenario in which the tagger is treated as a black-box and applied to all sorts of data without caring much about the domain. In the next section, we investigate how well the models perform in different domains.

3.1 Domain-specific results

Figure 2 gives a graphical overview of the evaluation results per domain for English, while Table 3 shows the exact values. As expected, some models that are especially trained for a certain domain perform well in that domain, but not in another. One such example is the *Ark-3* model, a model specialized for social media that is among the best and fastest models on that domain, while it does not perform well on the other domains. However, there are also counter-examples like the *Clear* model that not only performs well on formal writing, but also on the speech transcripts and social media. In general, the differences between the domains are smaller than expected. The absolute accuracy values are best for written, followed by spoken, and worst for social media which fits the expectations.

When looking at the German domain-specific results (Figure 3 and Table 4), we see a similar distribution as for English with little differences between domains. An interesting exception is the *TreeTagger* that is quite fast on written data (reflecting its popularity for tagging German), but rather slow on social media. As *TreeTagger* is not open-source, we could not further investigate the reasons for this difference.

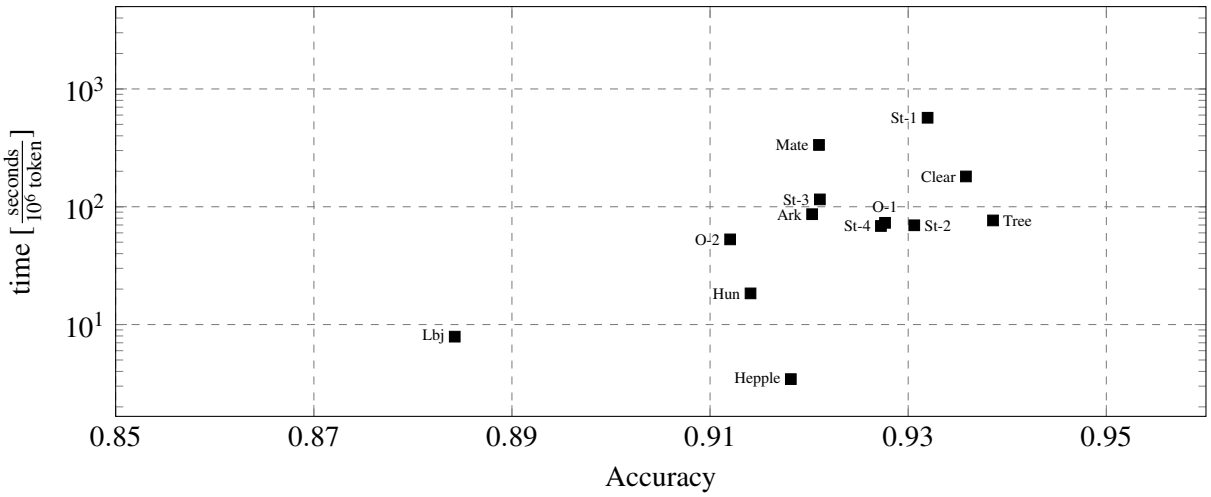
4 Conclusions and future work

In this work, we evaluated a large set of PoS tagging models on a wide range of English and German data from different domains. We find that researchers need to choose between accuracy and speed depending on their needs. The comprehensive results in this paper offer some guidance in this respect.

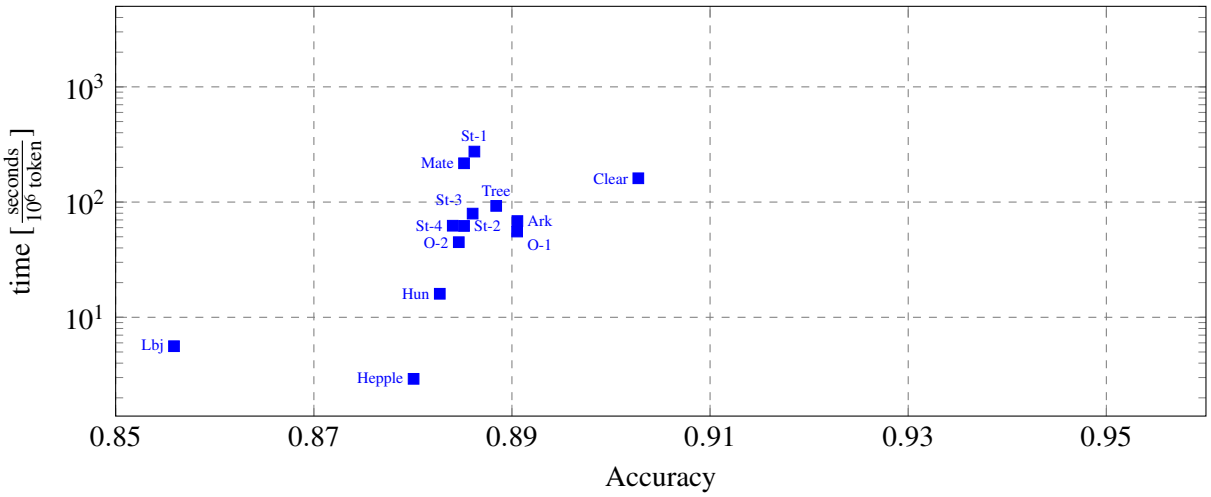
We make our full experimental framework available which will enable researchers to easily extend our analysis to other languages and taggers or compare taggers under different conditions.[6]

[6] https://github.com/zesch/pos-tagger-evaluation.git

(a) Formal / written



(b) Spoken



(c) Social

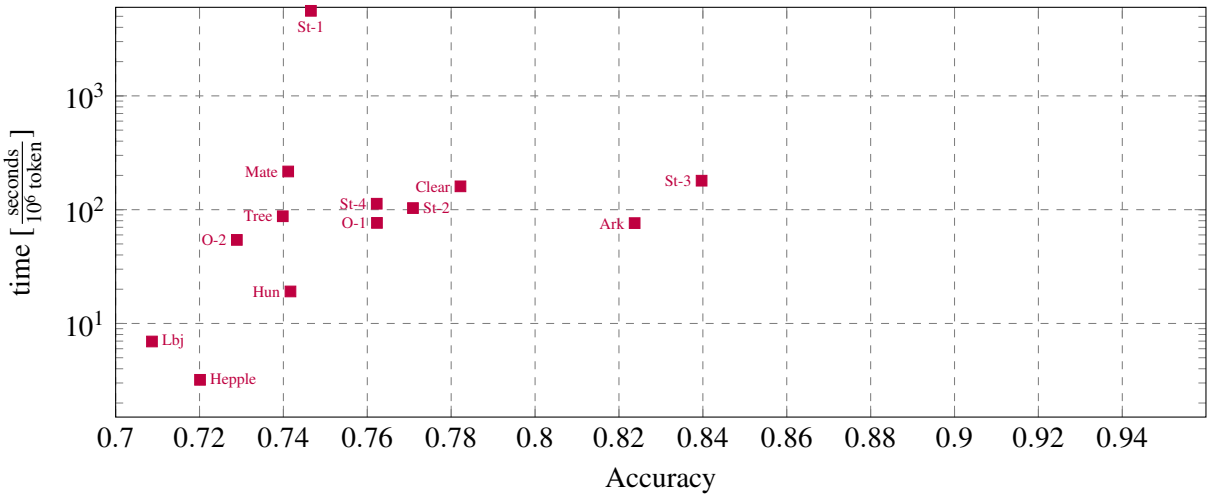
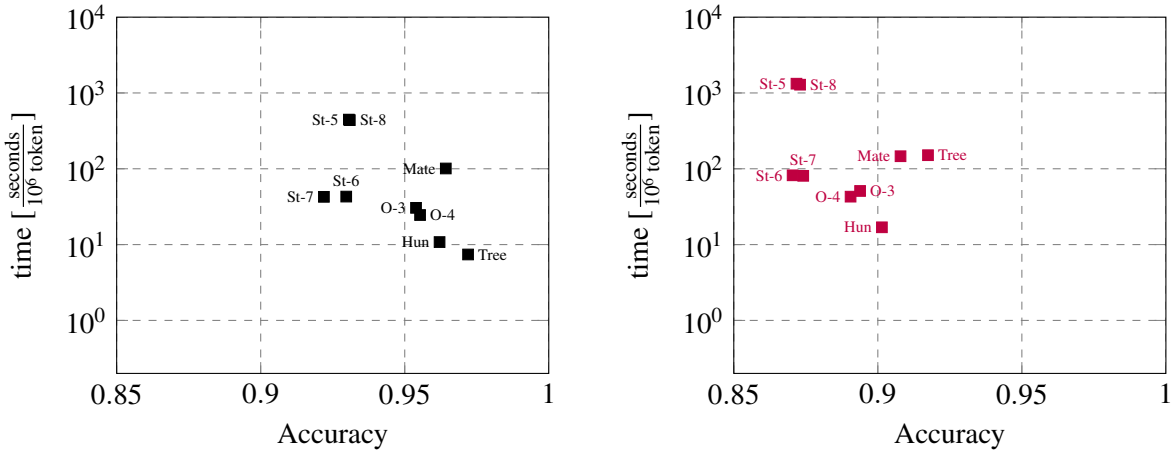


Figure 2: English results per domain.

	Written		Speech transcripts		Social media		Macro-Average	
	accuracy	time	accuracy	time	accuracy	time	accuracy	time
	∅ %	∅ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	∅ %	∅ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	∅ %	∅ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	∅	∅ ($\frac{\text{seconds}}{10^6 \text{ token}}$)
Ark	92.0	86	89.1	68	91.1	76	89.2	79
Clear	93.6	181	**90.3**	161	89.5	160	**89.5**	171
Hepple	91.8	3	88.0	3	84.1	3	86.7	3
HunPos	91.4	18	88.3	16	86.4	19	87.0	18
Lbj	88.4	8	85.6	6	83.0	7	84.1	7
Mate	92.1	335	88.5	217	86.2	217	87.4	276
O-1	92.8	73	89.1	56	87.3	77	88.3	68
O-2	91.2	53	88.5	45	84.3	54	86.7	51
St-1	93.2	570	88.6	274	87.1	5589	88.1	1485
St-2	93.1	70	88.5	62	88.0	103	88.5	74
St-3	92.1	115	88.6	79	**93.6**	180	89.4	118
St-4	92.7	69	88.4	62	87.1	113	88.1	76
Tree	**93.9**	77	88.8	93	86.6	88	88.4	84

Table 3: English tagging accuracy and execution time. Highest accuracies per domain in bold face.



(a) Written

(b) Social

Figure 3: German results per domain

	Written		Social media		Macro Average	
	accuracy	time	accuracy	time	accuracy	time
	∅ %	∅ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	∅ %	∅ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	∅ %	∅ ($\frac{\text{seconds}}{10^6 \text{ token}}$)
Hun	96.2	11	90.1	17	93.2	14
Mate	96.4	101	90.8	146	93.6	124
O-3	95.4	31	89.4	51	92.4	41
O-4	95.5	25	89.1	43	92.3	34
St-5	93.1	445	87.2	1325	90.1	885
St-6	93.0	43	87.0	82	90.0	62
St-7	92.2	43	87.4	81	89.8	62
St-8	93.1	438	87.3	1285	90.2	861
Tree	**97.2**	7	**91.7**	151	**94.5**	79

Table 4: German tagging accuracy and execution time. Highest accuracies per domain in bold face.

5 Acknowledgement

We would like to thank Richard Eckart de Castilho for his valuable input and for his incredible work on the DKPro Core framework.

References

Chris Biemann. 2006. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the COLING/ACL 2006 Student Research Workshop*, pages 7–12. Association for Computational Linguistics.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, pages 33–36, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. Tiger: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2(4):597–620.

Thorsten Brants. 2000. Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 152–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jinho D. Choi and Martha Palmer. 2012. Fast and robust part-of-speech tagging using dynamic model selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers – Volume 2*, ACL '12, pages 363–367, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 600–609, Stroudsburg, PA, USA. Association for Computational Linguistics.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014*, pages 1–11, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

David Ferrucci and Adam Lally. 2004. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10(3-4):327–348, September.

Eric N. Forsyth and Craig H. Martell. 2007. Lexical and discourse analysis of online chat dialog. In *Proceedings of the International Conference on Semantic Computing*, ICSC '07, pages 19–26, Washington, DC, USA. IEEE Computer Society.

Eugenie Giesbrecht and Stefan Evert. 2009. Is part-of-speech tagging a solved task? an evaluation of pos taggers for the German web as corpus. *Proceedings of the Fifth Web as Corpus Workshop*, pages 27–35.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers – Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 1–18, Stroudsburg, PA, USA.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Hunpos: An open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 209–212, Stroudsburg, PA, USA.

Mark Hepple. 2000. Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *COMPUTATIONAL LINGUISTICS*, 19(2):313–330.

Dalos D Miguel and Rachel Edita O Roxas. 2007. Comparative Evaluation of Tagalog Part-of-Speech Taggers. *4th National Natural Language Processing Research Symposium Proceedings*, pages 74–77.

W. Nelson Francis and Henry Kuçera. 1964. Manual of information to accompany a standard corpus of present-day edited american English, for use with digital computers.

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Ines Rehbein. 2013. Fine-Grained POS Tagging of German Tweets. In *Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, pages 162–175.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA.

Dan Roth and Dmitry Zelenko. 1998. Part of speech tagging using a network of linear separators. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Wojciech Skut, Hans Uszkoreit, Thorsten Brants, and Brigitte Krenn. 1998. A linguistically interpreted corpus of german newspaper text. In *Proceedings of the 10th European Summer School in Logic, Language and Information (ESSLLI'98). Workshop on Recent Advances in Corpus Annotation, August 17-28*, Saarbrücken, Germany.

Heike Telljohann, Erhard Hinrichs, Sandra Kübler, Ra Kübler, and Universität Tübingen. 2004. The tüba-d/z treebank: Annotating german with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2235.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology – Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amir Zeldes. 2016. The gum corpus: creating multilayer resources in the classroom. *Language Resources and Evaluation*, pages 1–32.

GermaNER:
Free Open German Named Entity Recognition Tool

Darina Benikova[1] Seid Muhie Yimam[1] Prabhakaran Santhanam[2] Chris Biemann[1]
(1) FG Language Technology, CS Dept., TU Darmstadt, Germany
benikova@aiphes.tu-darmstadt.de, {yimam,biem}@cs.tu-darmstadt.de
(2) IIT Patna, Dept. of CS and Eng., India
prabhakaran.cs11@iitp.ac.in

Abstract

With this paper, we release a freely available statistical German Named Entity Tagger based on conditional random fields (CRF). The tagger is trained and evaluated on the GermEval 2014 dataset for named entity recognition and comes close to the performance of the best (proprietary) system in the competition with 76% F-measure test set performance on the four standard NER classes. We describe a range of features and their influence on German NER classification and provide a comparative evaluation and some analysis of the results. The software components, the training data and all data used for feature generation are distributed under permissive licenses, thus this tagger can be used in academic and commercial settings without restrictions or fees. The tagger is available as a command-line tool and as an Apache UIMA component.

1 Introduction

Named Entity Recognition (NER) is the detection and classification task of proper names in continuous text. NER is used in information extraction, question answering, automatic translation, data mining, speech processing and biomedical science (Jurafsky and Martin, 2000). Moreover, it is a pre-processing step for deeper linguistic processing such as syntactic or semantic parsing, and co-reference resolution.

Despite German being a wide-spread and comparatively well-resourced language, German NER has not received a lot of attention. To the present day only three notable datasets exist, namely CoNLL-data (Tjong Kim Sang and De Meulder, 2003), an extension of this dataset to user-generated content by Faruqui and Padó (2010) and the NoSta-D NE dataset (Benikova et al., 2014b). So far, there has been no freely available German NE tagger. NER for German is especially challenging, as not only proper names, but all nouns are capitalized, which renders the capitalization feature less useful than in other Western-script languages such as English or Spanish. A baseline established on capitalized words therefore fails to show even moderate accuracy levels for German. This is reflected in previous results, e.g. from the CoNLL-2003 challenge, where German NER systems scored in the range of 70%-75% F-measure, as opposed to a recognition rate of 90% for English (Tjong Kim Sang and De Meulder, 2003).

We present GermaNER, a generic German NE tagger that can be easily executed from a command line or integrated into an NLP application. This paper presents the mechanism of the tagger, including the creation and experimental evaluation of the utilized features. The evaluation of the feature performance is accomplished using the F-measure, precision, and recall.

The tagger identifies the four default coarse named entity classes LOCation, PERson, ORGanisation, and OTHer. We have pragmatically excluded other NER subclasses and nested NERs from the GermEval 2014 task.

1.1 Free permissive licensing

Our most important contribution is the availability of GermaNER under a permissive license that allows academic and commercial use without licensing fees. The software components are mixed-licensed under modified BSD and ASL 2.0 licenses, the training and feature data is licensed under CC-BY. Unfortunately, these strict conditions on the permissiveness of licenses are not easy to meet. While it would have been possible to use more and better preprocessing components, more and better word lists for feature generation and possibly a better classifier, we had to exclude

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 31–38, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

the most part of them since many components are only free for academic use. We believe that placing these restrictions on software and data from publicly funded projects is hampering the development of language technologies as a whole, and German language processing in particular. The challenge of not being able to use standard preprocessing like part-of-speech tagging, however, led us to incorporate the output of several unsupervised methods that model required structural characterization in alternative ways.

2 Related Work

So far, two datasets were used for German NER in the academic community. The CoNLL-data by Tjong Kim Sang and De Meulder (2003) had flaws due to its inconsistencies in the training data, which were probably due to the circumstance that the annotators were non-native speakers (Leveling and Hartrumpf, 2008). Systems participating in the CoNLL 2003 contest achieved F-measure between 70%-75%. Faruqui and Padó (2010) have extended this data for evaluation purposes, and made available a German NER module for the Stanford NER tagger (Finkel et al., 2005) which is however, only free for academic use.

The NoSta-D NE data set by Benikova et al. (2014b) was used for the GermEval 2014 NER shared task (Benikova et al., 2014a). But the setting of the task is different to the one used for this project. In the GermEval 2014, the NE annotation were performed on nested-layers, so that entities like 'Madrid' in 'Real Madrid', were also detected. Moreover, in the shared task, derivations like 'German' and parts of NEs, such as 'Germany' in 'Germany-wide' are annotated.

The three best systems at GermEval 2014, ExB (Hänig et al., 2014), UKP (Reimers et al., 2014) and MoSTNER (Schüller, 2014) perform in a range of 73%-79% F-measure on the default set of four NER types (Metric 3 first-level spans) (Benikova et al., 2014a). All these systems implemented machine learning methods that make use of interdependencies among data points, such as Conditional Random Fields (CRFs) and Neural Networks.

While most participants used POS-level, character-level and gazetteer-based features, each of the three best performing systems (Reimers et al., 2014; Schüller, 2014; Hänig et al., 2014) operated with high-level semantic features, such as similarity clusters or word embeddings. These features were created using unsupervised learning methods on large corpora and successfully address the vocabulary problem and sparsity issues through vocabulary clusters (Schüller, 2014; Hänig et al., 2014) or dense vector representations (Reimers et al., 2014).

As previously shown, the use of such simple semantic generalization features improves the recall for NER (Biemann et al., 2007; Finkel and Manning, 2009; Faruqui and Padó, 2010). Moreover, the ExB system applied well-curated NE-specific suffix lists, containing entries such as '-stadt', '-hausen', or '-ingen' for locations.

3 Machine Learning Approach

There are different approaches to NER, including handcrafted rule-based algorithms, supervised machine learning, unsupervised machine learning and semi-supervised algorithms (Nadeau and Sekine, 2007). While a rule-based NER approach usually produces a high precision, it covers a single domain and fails to perform well when new entity types appear in a document (Petasis et al., 2001). Machine learning approaches perform more robustly and are more accurate if sufficient training data and adequate features are incorporated. Supervised NER approaches mainly depend on large collections of texts that are syntactically annotated to systematically recognize NEs based on syntactic patterns (Nadeau et al., 2006).

In our work, we focus on the development of a supervised machine learning NER system that can 1) be readily used from command line or integrated into any NLP applications to automatically tag NEs, and 2) be used as reasonable baseline system to further expand training data sets using active learning and adaptive annotation approaches, and 3) is not subject to license restrictions, thus can be freely downloaded and used by anyone.

3.1 Conditional Random Field (CRF)

While there are plenty of machine learning algorithms for sequence tagging, we choose to integrate a CRF (Lafferty et al., 2001) as it is highly accurate, scalable and easy to use as the training data can be prepared without the need of machine learning experts (Hoefel and Elkan, 2008). We have specifically integrated CRFsuite (Okazaki, 2007), a fast implementation of Conditional Random Fields, into a clearTK UIMA framework

(Bethard et al., 2014) to make training, feature annotation, classification and entity extraction more convenient.

The NER system is highly configurable, which allows users to either use the built-in model that is already optimized with our feature set, or train it with new training data and features sets. In order to make the NER system usable for both low-end and high-end machines, it provides a technique of data chunking, where users with high-end machines can use larger data chunks while users with low-end machines can still run the system on their laptop computer with smaller data chunks.

4 The NER system pipeline

The NER tagger pipeline consists of different components integrated into an UIMA (Ferrucci and Lally, 2004) pipeline written in the Java programming language. We have designed the NER tagger in such a way that each of the components can be replaced or modified easily. The first component of the system obtains the training and testing data and applies segmentation and tokenization that is stored in a UIMA CAS for further processing. The next component is the feature extraction process that internally annotates the documents accordingly. Feature extractors obtain different features either from the token and surrounding tokens, such as word and character n-grams, or the features are supplied from external sources, such as gazetteer lists or lists induced by unsupervised methods. The training component produces a CRFsuite classifier model based on the annotated features. The final component is a classifier component where unseen documents, which get feature-annotated in a similar way as the training file, are subject to prediction of NEs. Figure 1 shows a diagram of the GermaNER tagger system pipeline.

5 Data

5.1 Training Data

For training the NE-Tagger, we use the NoSta-D NE dataset. It consists of 31,300 sentences and more than 37,000 named entity span annotations that are used for training. The original dataset contains more annotations such as partial NEs and NE derivates and nested annotations, which were used in GermEval 2014, but have been excluded in GermaNER. The classes that are used in the training are LOCation, PERson, ORGanisation, and OTHer. All derivation and part classes that are contained in the original dataset were treated like unannotated tokens, because the task of GermaNER is the tagging of the four default coarse NE classes. However, these classes have been used for training and testing for the purpose of comparing the results of GermaNER to the systems that participated in the GermEval 2015 as shown in Table 2.

The training, development and test set were divided just as in the GermEval setting: 24,000 sentences for training, 2,200 for the development and 5,100 sentences for the test set. We optimized feature combinations on the development set and report evaluation scores for the same settings as in the GermEval 2014 challenge.

The final model of GermaNER as included in the GermaNER distribution was trained on the concatenation of training, development and test set. While we cannot assess the quality of the model, the test set performance as reported here can serve as a lower-bound estimate.

5.2 Data Input and Output Format

The input of GermaNER is a file, similar to the CoNLL format, which contains one token per line. Sentences should be separated by a blank line. The output of the tagger is a tab-separated file. The first column is the same as in the input file. The second column holds the predicted NE-tag. The NE-tags are similar to those employed in the training dataset, which made use of the BIO-scheme[1]. An exemplary output sentence of the tagger is presented below.

[1] "The **BIO** scheme suggests to learn classifiers that identify the **B**eginning, the **I**nside and the **O**utside of the text segments."(Ratinov and Roth, 2009)

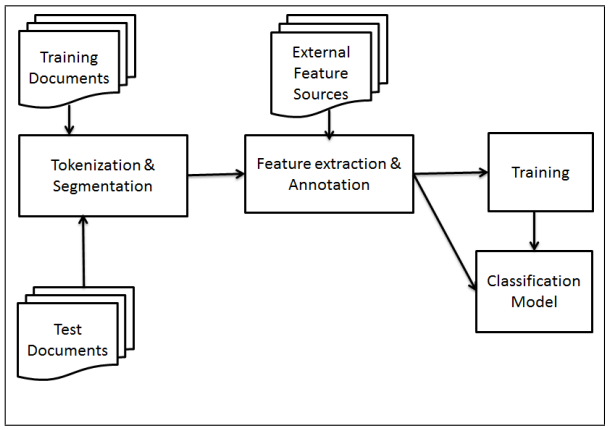


Figure 1: The German NER tagger pipeline

Nehmen	O
Sie	O
die	O
berühmte	O
Rede	O
von	O
Richard	B-PER
Feynman	I-PER
,	O
There	B-OTH
's	I-OTH
Plenty	I-OTH
of	I-OTH
Room	I-OTH
at	I-OTH
the	I-OTH
Bottom	I-OTH
,	O
von	O
1959	O
,	O
die	O
ist	O
damals	O
in	O
der	O
Zeitschrift	O
des	O
California	B-ORG
Institute	I-ORG
of	I-ORG
Technology	I-ORG
abgedruckt	O
worden	O
.	O

Table 1: Exemplary output of GermNER

6 Feature Representation

The creation and selection of features is a crucial part in the development of NER systems. The creation of all included features will be presented in feature groups, as the discussion of every single feature would be redundant. Some of the feature-groups are: 1) n-gram features such as character n-grams, unsupervised POS tag n-grams, and topic cluster n-grams, 2) time-shifted features, i.e. token-based features from surrounding tokens in relative position {-2,-1,0,+1,+2} to the current token 3) combinations of 1 and 2 such as as character n-grams features for one token to the left and right. We now provide details for all features.

6.1 Character and Word Features

This feature group consists of the first and last character uni-, bi- and trigrams of the current token, i.e.. prefixes and suffixes, time-shifted from -2 to +2. Similarly, character category pattern features, which are extracted from the current token based on unicode categories[2] from clearTK are used, and were found to be an influential feature for the system. Further, we use the words themselves as features in a window between -2 and 2.

6.2 NE Gazetteer

This gazetteer feature was created through the assembling of several lists containing NEs. Gazetteers may help to identify NEs that are known to be proper nouns in other contexts. For the *FreebaseList*, several Freebase lists containing proper nouns were merged. Freebase (Bollacker et al., 2008) is an English community-curated data-base containing well-known places, people and things under CC-BY-license. It contains 47 million lists, so-called topics, and 2 billion entities. The entities are ordered into different topics (e.g. Music Album, Family Name, or Continent) which are part of domains (e.g. People, Music, or Location). The largest task relevant lists as well as lists with frequent NEs such as Country or Currency were chosen for the final list. The following lists were incorporated in the gazetteer: Album, Mountain, Book, Musical Group, Book Edition, Organization, Citytown, Person, Country, River, Currency, Stock Exchange, Film Track, Human Language, TV-series-season, Lake, Work of Fiction, and Location.

Only the first column of the lists provided by Freebase was used for this task. The lists were stripped of all entries containing special characters or spaces only. Moreover, double entries of proper names in the same list were removed.

The final *FreebaseList* is a tab-separated file consisting of two columns. The first column contains the proper name and the second column contains the name of the list file it was extracted from. It was used as a look-up table to extract features for every token that was in the table for the corresponding class.

Several other gazetteers were incorporated as features, such as personal name lists extracted with the NameRec tool from ASV Toolbox (Biemann et al., 2008) from large, publicly available corpora. This merged feature group *Gazetteer features* is shown seperatly from the *FreebaseList* in Table 2.

[2] http://www.unicode.org/notes/tn36/

6.3 Parts of Speech

There are several approaches of machine learning for retrieving parts of speech (POS) of words in context automatically. We have incorporated automatically induced POS tags as POS features.

This POS induction is based on the system by Clark (2003), which clusters words into different classes in an unsupervised fashion, based on distributional and morphological information. For this setup, we have used 10 million sentences, which are part of the Leipzig Corpora Collection[3] Richter et al. (2006), and induced 256 different classes.

Additionally, we experimented with classical POS features using the Mate POS tagger (Bohnet, 2010). Our tool will not include this feature as the licenses of the POS tagger and its training data would render our tool unusable for commercial purposes. However, we will provide the possibility to add this feature so that it can be used in an academic setting.

6.4 Word Similarity

This feature group consists of the four most similar words of the current token, obtained from the JoBimText[4] (Biemann and Riedl, 2013) distributional thesaurus database, made available in a window of size 2.

6.5 Topic Clusters

Inspired by the semantic clusters of the ExB system, we have applied LDA topic modelling[5] to above-mentioned JoBimText German distributional thesaurus, using the thesaurus entries as 'documents' for LDA. This results in a fixed number of topic clusters, most of which are quite pure in terms of syntactic and semantic class. We have generated different sets of such clusters, each for all words and for uppercase words only, and use the number of its most probable topic as a token's feature – again, time-shifted in a range of -2 to 2. We experimented with sets of 50, 100, 200 and 500 clusters. In the final version we solely use the set of 200 clusters.

6.6 Other

There are two further features that were implemented: token *Position* and *Case*. *Position* feature is the position of the token in the sentence, while *Case* feature is the case of the token, distinguishing between uppercase and lowercase, the beginning of a sentence, camelCase and all uppercase, time shifted between -2 and 2.

7 Evaluation

In this section, the evaluation metric and other factors influencing the choice of features of the final GermaNER tagger will be presented.

7.1 Methods

For the evaluation of the feature performance, we report scores from the M3.1 metric described in the GermEval 2014 task. It calculates the precision, recall, accuracy and f-measure of the outer layer, which was the only layer of interest for the work described in this paper. To further investigate the issues of the current version of the tagger, the performance on individual classes will be discussed.

In order to determine the optimal feature set, different feature combinations have been tested in order to arrive at a final default feature set. The default feature set contains all features. To determine features that potentially reduce the performance of the NER by overfitting, we performed ablation tests.

7.2 Results

Table 2 shows the results of the previously described evaluation on the development and test data sets. The first line shows results with all features. The scores in boldface indicate the three most influential features, as leaving them out results in the most dramatic drops in tagging quality. The last line displays the performance of the full feature set including supervised POS features from the Mate POS tagger, which is however that is not part of our final system.

Table 2 shows that all features are relevant to the NER tagger, thus the final tagger makes use of all the described features. The best performing feature group are character n-grams. As not only n-grams of the current word, but also n-grams of preceding and following words are used, this features play a role that is on the one hand similar to the detection of prefixes and suffixes that indicate NEs, but are on the other hand similar the detection of words typically preceding or following NEs e.g. prepositions that precede NEs. The

[3] corpora.uni-leipzig.de

[4] http://www.jobimtext.org

[5] http://gibbslda.sourceforge.net/

Model	Precision (%)	Recall (%)	F-measure (%)
All features	83.16	74.32	78.49
no character n-grams	82.18	69.81	**75.49**
no case information	81.93	73.29	**77.37**
no gazetteers	82.75	73.92	78.08
no positions	82.69	74.23	78.23
no Freebase	82.56	73.56	77.80
no char cat. pattern	82.93	72.89	77.58
no similar words	82.29	73.25	**77.50**
no topic clusters	82.48	73.60	77.79
no clark POS induction	82.64	73.34	77.71
with Mate POS tagger	82.65	75.12	**78.71**

Table 2: Results of feature performance evaluation on the development set. Lower F-measure means a high impact of the corresponding feature

second best performing feature group is case information, meaning the classification of words in e.g. uppercase and lowercase. Although, as already mentioned earlier, this feature is not as distinguishing for proper nouns in German as it is in other languages, our experiments show that it still is an important feature in German NER. These two best performing features are standard features in NE detection, the advantage of which is confirmed through our experiment. The third best performing feature is a *similar words*, which is a semantic feature. This not only shows the importance of the semantic layer in this task, but also goes in line with the three best systems participating in the GermEval 2014, that also made use of high-level semantic features. Interestingly, the supervised POS tagger reduces precision and increases recall, resulting in a very modest increase of 0.22% F-score, thus is mostly subsumed by the unsupervised feature groups.

Table 3 reports overall P/R/F-results when training GermaNER on the concatenation of the training and development set and testing it on the official test set and also provides scores of the best GermEval 2014 participants. While the first two results are not directly comparable since the challenge participants were also asked to tag NE derivates and partial NEs, they indicate that GermaNER shows a competitive score to the UKP system and is outperformed by the proprietary ExB system. Interestingly, GermaNER has a comparatively high precision but a lower recall compared to ExB and UKP. To provide a better comparison to the other systems, the GermaNER system was trained on the concatinated training and development set including parts and derivations. The result, which is shown in the last line in Table 3, shows that although the tagging of parts and derivs was not the focus of this tagger, GermaNER is only outperformed by ExB and UKP.

	ExB	UKP	MoSTNER	GermaNER
PER	84.05	85.48	82.54	85.33
LOC	84.05	84.62	80.47	81.39
ORG	76.29	69.60	62.24	68.23
OTH	59.46	49.81	48.38	52.72

Table 4: Test set peformance in % F-measure by NE type for top GermEval 2014 systems

Both Table 2 and Table 3 show that the adjoining of a license resticted supervised POS tagger noticably improves the performance of GermaNER. These examples demonstrate the impact of permissive licensing on the performance of freely available tools.

Finally in Table 4, we provide an F-measure comparison broken down into the four coarse NER classes. Here, it becomes apparent that GermaNER is very strong on PERsons and that there is still some headroom for the other three classes, probably due to the lack of gazetteers for these other classes.

8 Conclusion and Future Work

We have developed GermaNER, a statistical German NER tagger which can be readily used from command line or can be integrated to an NLP application. While the architecture and the features of GermaNER are following common practice in sequence tagging and do not provide much

System	Precision (%)	Recall (%)	F-measure (%)
ExB	80.67	77.55	79.08
UKP	79.90	74.13	76.91
MoSTNER	79.71	67.74	73.24
GermaNER	82.72	71.19	76.52
GermaNER with POS	82.16	72.21	76.86
GermaNER including deriv and part	81.98	69.88	75.45

Table 3: Results of best GermEval 2014 systems and GermaNER on the test set, for different sets of classes from the NER dataset.

methodological novelty, we would like to stress the fact that the tagger is freely available in open source for download[6] under a permissive mixed license, allowing its use as a standalone or as a component in academic and commercial contexts without license restrictions or fees.

In its best configuration, GermaNER performs at an F-measure of 78.49% on the GermEval 2014 dev set and at 76.52% on the test set. The three features with the largest impact are the character n-grams, case information and similar words.

In summary: we provide a freely available German NER tagger for standard categories that comes close to the state of the art. Our largest challenges in creating this tagger were rooted in the fact that many resources and tools for language preprocessing are only available under restrictions. We hope to have advanced German language technology, both in academia and industry, by overcoming these limitations for named entity recognition and would like to see more free components in the future. Of course, everyone is welcome to add features and make high-quality German NER a community effort.

Acknowledgements

The authors would like to thank Martin Riedl for the idea and the implementation of the Topic Cluster feature in Section 6.5.

References

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. GermEval 2014 Named Entity Recognition: Companion Paper. *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition, Hildesheim, Germany*, pages 104–112, 2014a.

Darina Benikova, Chris Biemann, and Marc Reznicek. NoSta-D Named Entity Annotation for German: Guidelines and Dataset. In *Proceedings of LREC*, pages 2524–2531, Reykjavik, Iceland, 2014b.

Steven Bethard, Philip Ogren, and Lee Becker. ClearTK 2.0: Design Patterns for Machine Learning in UIMA. In *Proceedings of LREC*, pages 3289–3293, Reykjavik, Iceland, 2014.

Chris Biemann and Martin Riedl. Text: Now in 2D! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95, 2013.

Chris Biemann, Claudio Giuliano, and Alfio Gliozzo. Unsupervised Part of Speech Tagging Supporting Supervised Methods. In *Proceedings of RANLP-07*, Borovets, Bulgaria, 2007.

Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *Proc. International Conference on Computational Linguistics (COLING 2010)*, pages 89–97, Beijing, China, 2010.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, Vancouver,Canada, 2008. ACM.

Alexander Clark. Combining distributional and morphological information for part of speech induction. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 59–66, Budapest, Hungary, 2003.

Manaal Faruqui and Sebastian Padó. Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. In *Proceedings of KONVENS*, pages 129–133, 2010.

[6] https://github.com/tudarmstadt-lt/GermaNER

David Ferrucci and Adam Lally. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. In *Journal of Natural Language Engineering 2004*, pages 327–348, 2004.

Jenny R. Finkel and Christopher D Manning. Joint Parsing and Named Entity Recognition. In *Proceedings of HLT-NAACL 2009*, pages 326–334, Boulder, CO, USA, 2009.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL*, pages 363–370, Ann Arbor, MI, USA, 2005.

Christian Hänig, Stefan Bordag, and Stefan Thomas. Modular Classifier Ensemble Architecture for Named Entity Recognition on Low Resource Systems. In *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, pages 113–116, Hildesheim, Germany, 2014.

Guilherme Hoefel and Charles Elkan. Learning a Two-stage SVM/CRF Sequence Classifier. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 271–278, Napa Valley, California, USA, 2008.

Dan Jurafsky and James H Martin. *Speech & Language Processing*. Pearson Education India, 2000.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289, Williamstown, MA, USA, 2001.

Johannes Leveling and Sven Hartrumpf. On metonymy recognition for geographic information retrieval. *International Journal of Geographical Information Science*, 22(3):289–299, 2008.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae investigationes: Revue internationale de linguistique française et de linguistique générale*, 30(1):3–26, 2007.

David Nadeau, Peter D. Turney, and Stan Matwin. Unsupervised Named-entity Recognition: Generating Gazetteers and Resolving Ambiguity. In *Proceedings of the 19th International Conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence*, AI'06, pages 266–277, Québec City, Québec, Canada, 2006.

Naoaki Okazaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs), 2007. URL `http://www.chokkan.org/software/crfsuite/`.

Georgios Petasis, Frantz Vichot, Francis Wolinski, Georgios Paliouras, Vangelis Karkaletsis, and Constantine D. Spyropoulos. Using machine learning to maintain rule-based named-entity recognition and classification systems. In *Proceedings of ACL*, pages 426–433, Toulouse, France, 2001.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL*, pages 147–155. Association for Computational Linguistics, 2009.

Nils Reimers, Judith Eckle-Kohler, Carsten Schnober, and Iryna Gurevych. GermEval-2014: Nested Named Entity Recognition with Neural Networks. In *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, pages 117–120, Hildesheim, Germany, 2014.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. Exploiting the leipzig corpora collection. In *Proceedings of IS-LTC*, pages 68–73, Ljubljana, Slovenia, 2006.

Peter Schüller. MoSTNER: Morphology-aware split-tag German NER with Factorie. In *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, pages 121–124, Hildesheim, Germany, 2014.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, pages 142–147, Edmonton, Canada, 2003.

"God Wat Þæt Ic Eom God" — An Exploratory Investigation Into Word Sense Disambiguation in Old English

Martin Wunderlich
Center for Information and Language Processing (CIS)
University of Munich (LMU)

Alexander Fraser
Center for Information and Language Processing (CIS)
University of Munich (LMU)

P. S. Langeslag*
Dept. of English Philology
Medieval English Language and Literature Section
University of Göttingen

Abstract

Natural Language Processing (NLP) of historical languages is an understudied area. Much previous work has focused on the problems of normalization and POS tagging. In contrast, we consider a new problem, word sense disambiguation (WSD). We provide a survey of previous work on processing of historical languages and discuss what we can and cannot apply to the problem of WSD, specifically WSD of Old English (OE). We then annotate a new resource for supervised WSD, which we make available. Finally, we carry out proof-of-concept experiments, followed by a discussion of several promising areas for future work.

1 Introduction

We consider the important task of Word Sense Disambiguation (WSD) for historical languages, which to the best of our knowledge has not been studied extensively yet. WSD is at the heart of many applications in Natural Language Processing (NLP). For instance, in order to correctly translate polysemous or homonymous words in a machine translation system, one needs to disambiguate different word senses. The target language might make a clear lexical distinction where the words in the source language are homographs (Yarowsky, 2010). Examples of polysemy/homonymy would be lexical items, such as "Python" (the snake or the programming language?) in written language, the homophones "Perl" vs. "Pearl" in speech, or — as an example for a homograph — the word "God" in the (synthesized) Old English phrase "god wat þæt ic eom god" ("God knows I'm good").[1]

*Email addresses: martin.wunderlich@campus.lmu.de, fraser@cis.lmu.de, ps@langeslag.org

[1]Note that the latter ambiguity could easily be resolved, if there were a POS tagger for Old English.

We consider WSD for a language that is particularly difficult in this regard but also very interesting. Old English is a Germanic language that was spoken on the British Isles approximately from 450 to 1150 AD, then it gradually transformed into Middle English (ME), particularly under the influence of the conquerors' languages Old Norse and (Norman) French. In the area of NLP, Old English is an under-researched language. It has received relatively little attention — unlike its contemporary variant — and there are few digital resources and tools.

Our contributions are: First, we present a brief survey of the NLP literature on historical languages. Second, we annotate new gold standard training and testing data and make it available for future use. Third, results from a proof-of-concept study are used to show how the problem of WSD for OE can be concretely approached.

The initial results are promising, even with basic techniques. This demonstrates the feasibility of WSD and might motivate work on expanding the inventory of data and NLP tools available for OE, such as POS taggers and stopword lists, which can be applied in WSD and other tasks.

The remainder of this work is structured as follows: In section 2 an overview of the history of Old English is given. In section 3 we present a survey of works on the application of NLP to historical languages. Section 4 briefly summarizes existing work on WSD and details the methods that are being used in the present work. The focus of section 5 is the description of the practical development work that was carried out as part of this project. This section also includes an overview of existing digital resources for NLP as applied to OE, covering both digital corpora/lexica and existing tools. The steps taken for preparing and preprocessing of the digital resources are also described in full detail. Section 6 presents the proof-of-concept evaluation. Finally, section 7

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology, pages 39–48, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

summarizes our findings and discusses several avenues of future work.

2 An Extremely Brief History of Old English

From about 800 BC, the British Isles were populated by Celtic settlements.[2] After initial Roman military expeditions, starting with Julius Caesar in 55 BC, the Roman province of Britannia was established by the year 43 AD. In the 5th century AD, however, the Roman heartland came under pressure and so the troops were withdrawn. Their withdrawal was complete by 410 AD.

This vacuum of power was used in the 5th century by tribes from the north (the Scots and Picts) to push into the southern part of the British Isles, while at the same time Germanic tribes from the European mainland — the Angles, Saxons, and Jutes — likewise made their way to what came to be known as England. The Saxons settled in the south, whereas the Angles settled in the north and the Jutes in Kent.[3] The Anglo-Saxons quickly established rule over Britain and by the end of the 5th century Saxons and Celts lived under the "Rex Anglorum".

The Anglo-Saxons had brought their culture and languages with them, which were quickly adapted and transformed by the local population, giving rise to what is known as the Old English language. This Germanic language retained many grammatical features of its parent languages, which makes it quite distinct from contemporary English. The Old English alphabet was based on Latin and consists of 24 letters:

a **æ** b c d **ð** e f **ȝ**/g h i l m n o p r s/**ſ** t **þ** u **ƿ**/w x y

The language has a case system with five cases (nominative, genitive, dative, accusative, and vestiges of an instrumental) and three numbers (singular, dual and plural). OE is a strongly inflected language, as can be seen from the following two examples:[4] 1) **se** guma geseah **þā** cwēn ("the man saw the woman"); 2) **sēo** cwēn geseah **þone** guma**n** ("the woman saw the man")

The variations for case, gender, and number are clearly visible here in the definite article (highlighted). Also note the suffix for "guma" when used as a direct object in the second sentence (and the lack of such an inflection for "cwēn").

Irish and Roman missionaries introduced the Latin language at large scale (which had left few traces during the previous Roman occupation). Several word borrowings can be traced to Latin roots, such as "missa" – "mæsse" ("Mass"), "presbyter" – "prēost" ("priest") and "calendae" – "calend" ("calendar"). The OE language was further influenced by Old Norse, following several waves of Scandinavian raids and invasions first recorded in 787 and recurring into the late eleventh century. After the Treaty of Wedmore in 886, an area known as the "Danelaw" was established in northeastern England. Names for locations and people can be traced to these Scandinavian roots, such as "Whenby" or "Skewsby", "Jackson" or "Davidson". The initial "sk" in words such as "skirt", "skin" or "skill" has Old Norse roots. Also, common words like "same" or "give", and even some closed-class pronouns can be traced back to Old Norse: The 3rd person plural forms of the pronoun have Scandinavian roots.

The entire OE corpus that survives consists of only approximately 24,000 word types, around 15% of which have remained in Modern English and 3% are loan words (Crystal, 2010, page 27). Most of this corpus is in the West Saxon dialect, since under King Alfred's rule many works were translated from Latin into OE. The two other main dialects are Northumbrian and Mercian. A lack of standardized orthography, combined with sound changes, morphological, and dialectal variations, acted increase the number of word types and gave rise to word variations.[5]

3 Related Work On Historical Languages

As pointed out in the introduction, OE is an under-resourced and under-researched language when it comes to the field of NLP. Nevertheless, a few related studies that cover OE and other historical languages can be found. Sukhareva and Chiarcos (2014) examine the possibility of using data from related languages and dialects to compensate for the sparseness of annotated corpus data in OE and other historical languages. They use parallel biblical texts to train a dependency

[2] The following section is largely based on Crystal (2010, pages 6-29) and Schirmer and Esch (1977, pages 2-20)

[3] At least according to the traditional (and probably simplified) account by the Northumbrian monk Bede; cf. (Crystal, 2010, page 6)

[4] Taken from Crystal (2010).

[5] Such as "wunderlic", "wundarlic", "wundorlic", which might be translated as "peculiar", "strange".

parser and find that annotation projections derived from word alignments allow for cross-language parser adaptation. The authors speculate "[...] that languages separated for 1000 years (OE-IS) or more are too remote from each other to provide helpful background information, but that languages separated within the last 750 years (ME-DE) or less are still sufficiently close."[6] (Sukhareva and Chiarcos, 2014, page 15) In the context of the present work this means that in terms of the temporal distance resources for ME might be useful, but in terms of the relatedness of OE and ME, the languages might be too different, due to the influences described in the previous section.

Pennacchiotti and Zanzotto (2008) evaluate to what extent existing NLP tools for contemporary Italian are suitable for POS tagging applied to fourteenth-century Italian using a corpus of fourteen major Italian literary works, such as Dante Alighieri's *Divina Commedia* from 1321. In addition, the authors test in what manner simple modifications and customizations of the existing tools might improve their application to late medieval Italian. The evaluated accuracy of the POS tagging ranges between 0.54 and 0.90. In conclusion, the authors find that the results "[...] support our initial claim that the dictionary and the Chaos parser for contemporary Italian are insufficient for the analysis of ancient texts, as there exists a significant gap in dictionary coverage between contemporary and ancient texts." (Pennacchiotti and Zanzotto, 2008, page 378) The authors also propose possible improvements, such as manually building a lexicon for each period, leveraging manually annotated corpora or adapting existing models by applying rules to capture morphological variations.

In a similar fashion, Meyer (2011) uses existing NLP resources for contemporary Russian to tag Old East Slavonic texts, by first annotating the modern version and then projecting part of the annotation back onto the corresponding original forms, based on a parallel corpus consisting of old and modern versions of the same texts. Meyer presents a system that goes through steps of sentence alignment, "guessing" of morphological categories, word alignment, creation of hyperlemmata[7] and, finally, annotation projection. The main result of this work is the finding that this method can be used to successfully derive morphosyntactic annotations in a process that is based on the disambiguation of the output of a morphological guesser with the help of aligned Modern Russian word forms and associated tags.

Bollmann (2013) carried out similar work in the area of POS tagging on historical German texts from two corpora: the 15th century Anselm corpus and GerManC-GS with texts from the 17th and 18th centuries. Various steps of normalization and different parametrizations are derived automatically by the Norma tool (Bollmann et al., 2012). POS tagging on the historical texts is evaluated in three different scenarios: first, tagging on the simplified, but otherwise unmodified, original texts; second, tagging on the gold-standard normalizations; and third, tagging on texts which have been normalized automatically. The author reports accuracy results of around 69.6% for Early Modern German texts and POS tagging results of 81.92% for the historical texts when tagging on gold-standard normalizations (vs. 95.74% for modern data) (Bollmann, 2013, page 16).

In a study pertaining to Middle English (Moon and Baldridge, 2007), tags from present day English source texts were projected to Middle English texts using alignments from a parallel Biblical text. The authors report a "[...] tagging accuracy in the low 80's on Biblical test material and in the 60's on other Middle English material." (Moon and Baldridge, 2007, page 390). This work was based on the annotated Penn-Helsinki Parsed Corpus of Middle English, containing texts from from around 1150 to 1500. This corpus contains approximately 1,150,000 words of running text from 55 sources. The texts are provided in three forms: raw, tagged, and parsed. Using a bigram tagger, "[r]esults were improved further by training a more powerful maximum entropy tagger on the predictions of the bootstrapped bigram tagger, and [the authors] observed a further, small boost by using Modern English tagged material in addition to the projected tags when training the maximum entropy tagger" (Moon and Baldridge, 2007, page 398).

As regards Early Modern English, Baron and Rayson (2008) carried out experiments using

[6]The language codes here stand for: Old English (OE), Middle English (ME), Middle Icelandic (IS), and Early Modern High German (DE).

[7]That is, "[...] an artificial label bundling together corresponding lemmata of different diachronic stages." (Meyer, 2011, page 274)

automatic spelling normalization. In the process of this, a tool was created called VARD 2, which could possibly be adapted for OE. Normalizing the spelling across the corpus helps to reduce the spelling variations that derive from the non-standardized orthography and thus reduce the noise that stems from these variations.

As is evident from the works cited above, the primary focus of NLP on historical texts has been the problem of POS tagging and the possibility of applying existing tools for contemporary languages to their historical counter-parts. Detailed studies on WSD for historical languages, particularly on OE texts, seem to be non-existent. Also, the works quoted above focus mainly on annotation projection, an approach which is not applicable for the WSD task since no sense-annotated corpus of a sufficiently closely related language exists to the best of our knowledge. The existing body of work shows that standard classification methods, such as maximum entropy, can be used successfully and that parallel corpora are a useful resource for historical languages, but only if the two languages are sufficiently closely related. For Old English, however, no such parallel corpus exists, so the present work is based on a monolingual body of text.

4 Methodological Background on WSD and Machine Learning Techniques

If the meaning of word is its usage in the language, as Wittgenstein claimed,[8] then it should be possible to derive the meaning by closely examining this usage. One aspect of the usage is the context in which a word appears with a certain meaning or word sense and, consequently, techniques for Word Sense Disambiguation focus on the context of a word to select the most likely word sense from a given "sense inventory" (Yarowsky, 2010). Essentially, WSD is a classification task using the context words in the sentence or paragraph and, possibly, additional information such as their POS tags, as evidence (Yarowsky, 2010). The term "sense inventories" here can mean any form of dictionary-based repository that maps lemmas or lexical items to word senses. The task of WSD consists of a semantic analysis or interpretation with the goal of deriving the meaning of an utterance. In WSD each word can be considered a classification problem in its own right (Cabezas et al., 2001), for the purpose of which each word instance is represented as a collection of feature-value pairs in vector form and the correct category assigned to this training instance in form of a unique sense ID or label.

Supervised machine-learning algorithms can be applied to this WSD classification problem. The ability to distinguish different word senses is "learned" from sense-labeled training examples of polysemous/homonymous words in the context of a sentence or paragraph. The context could, for instance, be a window size of 50 words to the left or right of the target word, which is cited by Yarowsky (2010) as a typical window. This window is then be converted to a bag-of-words feature vector, with either binary values, signifying the presence or absence, or using a more fine-grained representation, such as TF-IDF. Other features, such as the POS of a context word at a given position relative to the target word, might also be used (Yarowsky, 2010). Since an ambiguous word might have more than two meanings, the task can be modeled as a multi-class classification or a binary ("one versus all") classification. In the present work, one of the findings has been that binary classification in general performs better than multi-class classification.

Stevenson differentiates four categories for WSD tasks (Stevenson, 2003): 1) Semantic disambiguation where there are no restrictions as to the number or kinds of senses.[9] 2) Semantic tagging: Also known as the "all-words task" whereby all words have to be annotated with a specific word sense. 3) Sense disambiguation whereby some words (not all) are to be tagged with a specific sense from a lexicographical resource. 4) Sense tagging, whereby all words are to be tagged with lexical senses.

The task in the present work would fall into the third category, since only a selection of polysemous words is being tagged with word sense classes from a lexicon.

When running any kind of machine learning algorithm, it is useful to have a baseline that the results can be evaluated against. Stevenson presents a number of possible baselines in WSD tasks (Stevenson, 2003). However, for our purposes only two of these are relevant: 1) the random selection of a word sense and 2) the selection of

[8] Ludwig Wittgenstein: *Philosophische Untersuchungen*, §43, page 40. Suhrkamp Verlag, Frankfurt a.M., 6th ed., 2013.

[9] Also referred to as "word sense discrimination".

the most frequent word sense (from the training set). The other three baseline metrics proposed by Stevenson build on the Lesk algorithm,[10] which uses lexical overlap between the target word's context and dictionary definitions for classification and is therefore not applicable in our case, since the corpora here are in a language (Old English) that is different from the lexicographic definitions (Modern English). Usually, the Lesk algorithm should be strongly considered as a WSD algorithm and might be reconsidered for the work presented here if or when dictionaries with definitions in OE become available in the future. In terms of classification methods, the present work compares Naïve Bayes with Maximum Entropy, both evaluated against random and most frequent baselines.

5 Old English NLP Resources Used in the Present Work - Selection, Preparation, and Preprocessing

In the following section, the digital resources that formed the basis of our work are described. Statistics on the Old English corpus and lexicon are presented in sub-sections 5.1 and 5.2. The third sub-section (5.3) provides a brief overview of the preparation of the data used for training the machine-learning algorithm. Also, the feature extraction steps that were employed to generate feature vectors from the corpus data are described in that section.

5.1 The "Dictionary of Old English Corpus" (DOE Corpus) and Preprocessing Applied To It

Old English corpora are not as abundant as their contemporary counter-parts, but nevertheless some specimen can be found. In this present work, one main corpus was used, the DOE Corpus[11] or, more accurately, "The Dictionary of Old English Web Corpus", compiled by Antonette di Paolo Healey with John Price Wilkin and Xin Xiang (diPaolo Healey et al., 2009)[12]. The DOE corpus contains the text of at least one manuscript witness for every extant Old English text, including both prose and verse, as well as glosses, glossaries, and inscriptions

A number of preprocessing steps were undertaken, such as tokenization on sentence and word level. Other potentially useful pre-processing steps, such as lemmatization and POS tagging, were not possible, due to the lack of existing tools, but future work might use POS tagged data from the YCOE corpus. The following table 1 lists statistical information on the corpus:[13]

Number of HTML documents	3,037
Token count	3,786,753
Type count	343,135
Ratio of (token count / type count)	ca. 11
Total number of sentences	234113
Average sentence length	5.5
Minimum sentence length	1
Maximum sentence length	263

Table 1: Corpus statistics for the DOE corpus

5.2 Lexicographic Resources

In order to obtain a set of polysemous words, the Dictionary of Old English (DOE)[14] was employed. The DOE provides vocabulary from the first six centuries (600 - 1150 AD) of the English language and list entries for approx. 12,500 terms, currently ranging from letters A through G. The DOE comes in the form of HTML documents. The word counts by initial letter are given in table 2 (with some minor word count differences between the counts on the DOE website and the actual counts in the corpus).

The HTML format was parsed into a Java class structure and from this structure polysemous candidate terms were extracted. The criteria for the extraction were as follows:

- minimum token count 200
- minimum word length 3 characters

[10] For a detailed description of the Lesk algorithm see, for instance, Jurafsky and Martin (2008, pages 680f)

[11] Downloaded from the University of Oxford Text Archive - http://ota.ox.ac.uk; last accessed 2014-12-25

[12] An alternative might have been the York-Toronto-Helsinki Parsed Corpus of Old English Prose (YCOE), a 1.5 million word syntactically-annotated corpus of Old English prose texts (for which a corpus reader is included in NLTK), but the DOE corpus was chosen due to the larger volume. YCOE is a subset of DOE. For details see http://www-users.york.ac.uk/~lang22/YCOE/YcoeHome.htm

[13] The difference between the type count here and the OE type count of 24,000 provided earlier derives from the absence of any normalization and lemmatization in our software. Types are the raw word types exactly as they appear in the DOE corpus. So, for instance, "Fæder" and "fæder" would be counted as two different word types. The motivation for this is that we wanted to provide the generic text data whereas normalization would have lead to a loss of information.

[14] The DOE resources were last accessed and downloaded 2014-12-25 under http://tapor.library.utoronto.ca/doe/andhttp://tapor.library.utoronto.ca/doecorpus/

Letter	Counts on DOE website	Counts in HTML files
A	1,539	1,540
Æ	623	623
B	2,264	2,285
C	1,409	1,418
D	921	927
E	1,480	1,481
F	3,013	3,029
G	1,319	1,322

Table 2: Word counts from DOE

- non-Latin (i.e. no “dictum”, “confundantur”, “magister”...)
- minimum number of dictionary entries 2 (obviously)
- common nouns
- no proper nouns (e.g. no “Egypta”, “Micel”, “Iulianus”...)

The candidates were then reviewed manually to obtain an initial list of ten polysemous terms with sufficiently diverging word senses, as checked against the DOE definitions. From this shortlist of ten terms, we excluded those where the distribution of word senses in the randomly selected concordance matches was too skewed.[15] Table 3 in the appendix gives an overview of the seven remaining terms with their sense labels and definitions.

For the remainder of this present work, we will be focusing on the WSD results for the term “boc” as a representative and sufficiently ambiguous term.[16] Two examples for concordances of the target term “boc” shall serve to illustrate the format of the data (with doc ID and line ID for the DOE corpus):

- Doc ID: ÆGenPref; Line ID: 003800 (117); Ic bidde nu on Godes naman, gyf hwa ðas **boc** awritan wille, ðæt he hi gerihte wel be ðære bysne, for ðan ðe ic nah geweald, ðeah ðe hi hwa <to> woge gebringe ðurh lease writeras, & hit bið ðonne his pleoh na min: micel yfel deð se unwritere, gyf he nele his gewrit gerihtan.[17]
- Doc ID: MtGl (Li); Line ID: 062600 (19.7); dicunt illi quid ergo moses mandauit dari librum repudii et dimittere cuoedon him huæt forðon bebead sella **boc** freodomas & forleta[18].

One major drawback of the DOE is that it seems to have been engineered for use by human scholars and not by machines. There is no downloadable version in (TEI-)XML and there is no API for convenient access by other systems. Therefore, the dictionary had to be processed in HTML form and the information needed to be extracted from raw HTML tables into a structured Java object format.

To generate the training data, 100 concordance sentences for each word were randomly selected from the DOE corpus. Each occurrence was manually labeled with the sense ID of the top-level sense as per the DOE definition. These annotations were then verified by a second annotator. The final distribution is shown in figure 1. For the target word “boc”, the two instances of sense class C were removed. Also, one instance which used the word in the sense of the tree “beech” was removed and two instances could not be classified with sufficient reliability. This left a total of 95 training instances (with the three labels A: 33; B: 20; D: 42).

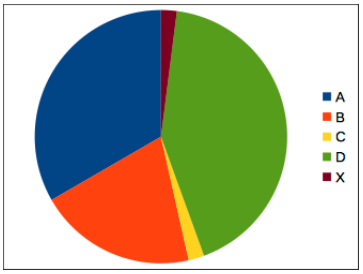


Figure 1: Word sense distribution of “boc”

The effort of the manual annotations was quite considerable, taking at least an hour per word, not including the quality checks and reviews. For a more comprehensive study, it would be possible perhaps to extract sense-labeled training data from the DOE files directly. This, however, could lead to sparsity issues, since on the lowest level

[15] This which would have lead to sparsity problems. The excluded terms with their word sense distribution were: “andlang” (1: 0; 2: 0; A: 16; B: 0); “ban” (A: 88; B: 6; X:6); “eadigen” (1: 21; 2: 4). The numbers do not add up 100 because the labeling was canceled once the skewed distribution became obvious. Sense labels are those from the DOE definitions.

[16] The additional data for the other target terms are available via the following URL (together with links to the code repository): `http://www.cis.uni-muenchen.de/~martinw/`

[17] Translation: “I ask now in the name of God, if anyone desires to copy this book, that he corrects it well by the exemplar, because I have no control if someone brings it to error through lesser scribes, and it is then his peril, not mine. The bad scribe does much evil if he will not correct his errors.” - translated by Brandon W. Hawk: `http://brandonwhawk.net/2014/07/30/aelfrics-preface-to-genesis-a-translation/` - last accessed 2015-05-14.

[18] Note the mixed language example with both Latin and Old English in this second example. The OE here is a gloss to the Latin text (Matthew 19:7) from the Vulgata. In the King James Bible, this verse is translated as: “They say unto him, Why did Moses then command to give a writing of divorcement, and to put her away?”

each meaning definition might have only two or three example sentences. It would be possible to circumvent this problem by merging all sub-level meanings into the top-level, but this procedure would have to be carefully evaluated first for its validity. A threshold for the minimum number of sample sentences could also be introduced, but again this might create sparsity problems.[19]

5.3 From Corpus to Feature Vectors

In order to train the various learning algorithms, the training data needs to be converted to an abstracted representation in the form of feature vectors, one vector per instance of a training word. A feature here is a particular characteristic derived from the context of the target word and the feature vector is a collection of several such features. Ng lists a number of possible feature types (Ng and Zelle, 1997): 1) surrounding words (unordered set within fixed size window or word from the entire sentence); 2) local collocations (short sequence with word order); 3) syntactic relations (e.g. verb-object relations); 4) POS of context words and morphological features.

Cabezas et al. (2001) distinguish between two types of features 1) feature f_{WIDE} will be non-zero, if f appears in wide context of target word w; 2) feature $f_{COLL(x,w)}$ will be the token $\pm$ x positions to the right or left of w.

The full feature set F in this case would then be the union of $f_{WIDE} \cup f_{COLL}$. Following these authors, two initial types of feature vector were obtained: 1) Unordered BoW vector, which comprised all words in the same paragraph as the given target word within a token window of $\pm$ x tokens, where x was varied between 1 and 20. 2) Collocation vector, by creating features for ordered words in a window of $\pm$ 20 words on either side of the target word. The following is an example for such a feature vector (bag-of-words) for a window size of n=5 for the example sentence from section 5.2:

```
godes(9)=1.0
naman(10)=1.0
gyf(11)=1.0
hwa(12)=1.0
ðas(13)=1.0
awritan(14)=1.0
wille(15)=1.0
ðæt(16)=1.0
he(17)=1.0
hi(18)=1.0
```

[20]

6 Evaluation Metrics, Experiments, and Results

6.1 Evaluation Metrics

The assessment of the various classification methods requires solid and pre-defined evaluation metrics. These metrics should then be compared to pre-defined upper and lower bounds, for instance those given by Gale, who lists 75% lower bound and 96.8% upper bound, derived from the agreement of human judges (Gale et al., 1992). During the test runs for each trained classifier the following metrics were calculated for each classification (per target word and per one-vs-all classification as regards the word senses): accuracy,[21] precision, recall, and balanced F1 measure.

6.2 Experiments and Results

We compared different machine-learning techniques for the use of Old English WSD, using two classifier types: Naïve Bayes and Maximum Entropy. Both were provided by the MALLET machine-learning library written in Java (McCallum, 2002).

For each type of learning algorithm, a multi-class classification was compared to the binary classification of creating one-vs-all classifiers per sense class. As the baseline to compare the results against, a random selection and a "most common sense" heuristic were both used.[22]

The two classification algorithms were trained on feature vectors as follows: 1) BoW vector with a token window between 1 and 20 tokens. 2) Collocational vector (i.e. including positional information) with a token window between 1 and 20 tokens. In the appendix, figure 4 presents the results from the baseline classification. Figure 5 presents the results from actual classification for

[19] A different approach that does not rely on hand-labeled data would be the use of clustering techniques, such as graph-based methods or using lexical expansion, to generate sense clusters in an unsupervised manner, as described e.g. in Bordag (2006), Biemann (2012) or Miller et al. (2012). This unsupervised approach is known as *Word Sense Induction* and might be applied to OE in future work.

[20] Adapted from the output of MALLET's PrintInputAndTarget pipeline step.

[21] Also known as the "exact match criterion" (Stevenson, 2003)

[22] Since these baseline classifiers did not exist in MALLET, they were created from scratch as part of this present work and have been accepted into the project as a contribution via GitHub. Accepted on 2015-01-19, see https://github.com/mimno/Mallet/

target term "boc" using Naïve Bayes and Maximum Entropy.

It can be seen from this latter table that the best results in term of classification accuracy for the target term "boc" were achieved for a Naïve Bayes classifier using a bag-of-words model and a binary classification task ("one-vs-all") for sense ID "D". The same combination also gave the best values for precision (0.85), recall (0.83), and F1 (0.82). Overall, from the two types of classification methods, Maximum Entropy yielded a slightly better average accuracy of 0.734 (as compared to Naïve Bayes with 0.729). Naïve Bayes scored slightly higher in terms of overall average F1 measure with 0.666 (MaxEnt: 0.658), but the differences are probably negligable.

7 Conclusion and Potential Future Work

This present work has tried to demonstrate in which manner modern methods of statistical text processing can be used for the purposes of word sense disambiguation on an under-resourced language like Old English, provided that corpora and dictionary resources exist in digital form.

In the future, more tools for processing Old English texts might become available, such as POS taggers and NE extractors, which could be used to generate richer feature vectors.[23] Also, such tools would be useful in the preprocessing steps and could reduce words to their lemmas, which might help improve classification results. The features provided by the different window sizes could be analyzed closely for sparsity issues and a form of count-based cutoff might be implemented to try to be more robust. Other methods of dimensionality reduction, for instance knowledge-free stemming (Porter-stemming for OE, simple learned stemming, or simple truncation) could also be applied to reduce the sparsity of features. Also, it could be possible to use existing data of Modern English to train ML algorithms for WSD, although one might have reservations about the prospects, since OE and ModE are syntactically and lexically very different languages.[24]

In this work we focused on the use of Naïve Bayes and Maximum Entropy as classification methods. Other common machine-learning techniques that have been applied for WSD could also be used, such as Bayesian networks (Bruce, 1995), content vector models in combination with clustering techniques and Singular Value Decomposition (Schütze and Pedersen, 1995), (Schütze, 1998), or Artificial Neural Networks (Veronis and Ide, 1990).

Future work could also apply the classification methods in combination with bootstrapping techniques,[25] especially when the set of sense-labeled training data is relatively sparse (cf. Ng and Zelle (1997)). Since at present there is no single best WSD method, it might also make sense to combine several different classifiers in such a fashion, even in cases where there is a more satisfactory abundance of training data and combine these classifiers in a framework of several WSD sources and systems (Stevenson, 2003).

Acknowledgments

The authors would like to acknowledge the valuable help provided by Winfried Rudolf (University of Göttingen) who helped with translations and general comments. Further, we would like to thank the three anonymous reviewers whose comments led us to improve the paper.

References

A. Baron and P Rayson. 2008. Vard 2: A tool for dealing with spelling variation in historical corpora. In *Online Proceedings of the Aston Postgraduate Conference on Corpus Linguistics*, Birmingham, U.K.

Chris Biemann. 2012. Word Sense Induction and Disambiguation. In *Structure Discovery in Natural Language*, pages 145–155. Springer, Berlin, Heidelberg.

Marcel Bollmann, Stefanie Dipper, Julia Krasselt, and Florian Petran. 2012. Manual and semi-automatic normalization of historical spelling - case studies from early new high german. In *Proceedings of the*

[23] Alternatively, the syntactical and POS information provided by the YCOE corpus might be parsed and applied for WSD.

[24] As Moon et al. note on the difference between ME and ModE: "It is also questionable whether it would still be robust on texts predating Middle English, which might as well be written in a foreign language when compared to Modern English." (Moon and Baldridge, 2007, page 398)

[25] In bootstrapping the available training data is used as an initial seed set to train a classifier. The trained classifier is then applied to a larger corpus and the sense-labeled word instances gained from this are added to the training set. A threshold should be set in advance so that only classifications with a certain confidence score get added. Bootstrapping can also be used to train a second classifier on the results of a first one, a form of multi-engine WSD, as used for instance in the system by Stevenson (2003).

11th Conference on Natural Language Processing (KONVENS 2012), LThist 2012 workshop, pages 342–350.

Marcel Bollmann. 2013. Pos tagging for historical texts with sparse training data. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability in Discourse*, pages 11–18, Sofia, Bulgaria, August. Association for Computational Linguistics.

Stefan Bordag. 2006. Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. In *EACL*. EACL.

Rebecca Bruce. 1995. A statistical method for word-sense disambiguation (phd thesis).

Clara Cabezas, Philip Resnik, and Jessica Stevens. 2001. Supervised sense tagging using support vector machines. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, SENSEVAL '01, pages 59–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

D. Crystal. 2010. *The Cambridge Encyclopedia of Language*. The Cambridge Encyclopedia of Language. Cambridge University Press.

Antonette diPaolo Healey, John Price Wilkin, and Xin Xiang, editors. 2009. *Dictionary of Old English Web Corpus*. Dictionary of Old English Project.

William Gale, Kenneth Ward Church, and David Yarowsky. 1992. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting on Association for Computational Linguistics*, ACL '92, pages 249–256, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall, 2 edition.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Roland Meyer. 2011. New wine in old wineskins? - tagging Old Russian via annotation projection from modern translations. *Russian Linguistics*, 35(2):267–281.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In *COLING*, pages 1781–1796.

Taesun Moon and Jason Baldridge. 2007. Part-of-speech tagging for middle English through alignment and projection of parallel diachronic texts. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 390–399, Prague, Czech Republic, June. Association for Computational Linguistics.

Hwee Tou Ng and John M. Zelle. 1997. Corpus-based approaches to semantic interpretation in NLP. *AI Magazine*, 18(4):45–64.

Marco Pennacchiotti and Fabio Massimo Zanzotto, 2008. *Natural Language Processing across time: an empirical investigation on Italian*, volume 5221, pages 371–382. Springer.

Walter F. Schirmer and Arno Esch. 1977. *Kurze Geschichte der englischen und amerikanischen Literatur*. Dtv ; 4291 : Wissenschaftliche Reihe. Dt. Taschenbuch-Verl., München, 4. edition.

Hinrich Schütze and Jan Pedersen. 1995. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Mark Stevenson. 2003. *Word sense disambiguation : the case for combinations of knowledge sources*. CSLI studies in computational linguistics. CSLI Publ., Stanford, Calif.

Maria Sukhareva and Christian Chiarcos. 2014. Diachronic proximity vs. data sparsity in cross-lingual parser projection. a case study on germanic. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 11–20, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Jean Veronis and Nancy M. Ide. 1990. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 2*, COLING '90, pages 389–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

D. Yarowsky. 2010. Word sense disambiguation. In Alexander Clark, editor, *The handbook of computational linguistics and natural language processing*, Blackwell handbooks in linguistics. Wiley-Blackwell, Oxford, 1. publ. edition.

Appendix - Detailed Results

On the next page, we present detailed results for the term "boc" as discussed in the experimental section.

Target term	Token count	DOE definitions with IDs/labels
Anweald	242	A. power, sovereignty, sway B.1. a sovereign's or lord's dominion: realm, domain, empire; B.2. referring to the world considered as God's dominion: dominion C. the name of the sixth order of angels in the celestial hierarchy: Powers
Are	308	A. honour B. mercy, grace, favour, help C. property, possession(s), goods, resources
Boc	567	A. book B. major division of a larger work C. register, record, list D. legal document
Dryhten	261	1. in poetry and laws: lord, ruler, chief 2. the Lord, God the supreme ruler 3. lord, applied to a pagan god
Fæder	416	A. father (of humans) B. of supernatural beings / abstractions: father
For	955	1. action of going, state of movement, motion 2. journey, trip, voyage; fore geferan / gefremman - to go on / make a journey; 3. armed foray; march of an army 4. rendering accessus, here the approach, access 5. path, course; here figurative: way of life, course of conduct 6. glossing vehiculum means of transport, vehicle, conveyance
Fultum	574	1. help, aid, assistance, support, succour 2. concrete: someone who or something which provides help, support 2.a. supporter (of someone gen.; of a monastery, into and dat.) 2.b. referring to military support in the form of a force, troop, army 2.c. in medical recipes: a remedy

Table 3: APPENDIX — WSD target words from the DOE corpus with labeled definitions (sense labels are the original ones from the DOE).

Training algorithm	**Classification type**	**Vector type**	**Accuracy**		**Precision**		**Recall**		**F1**	
			Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
rnd	A vs. not A	bow	0.55	0.13	0.49	0.27	0.51	0.30	0.47	0.26
rnd	A vs. not A	coll	0.57	0.14	0.53	0.31	0.57	0.28	**0.50**	0.25
rnd	B vs. not B	bow	**0.66**	0.13	0.55	0.36	0.56	0.36	0.49	0.33
rnd	B vs. not B	coll	0.64	0.17	0.51	0.41	**0.58**	0.39	0.45	**0.38**
rnd	D vs. not D	bow	0.49	**0.22**	0.52	0.28	0.49	*0.26*	0.48	0.23
rnd	D vs. not D	coll	0.53	0.17	0.53	*0.25*	0.53	*0.26*	**0.50**	0.21
rnd	multi	bow	0.38	0.18	*0.38*	0.33	0.37	0.33	0.32	0.29
rnd	multi	coll	0.37	0.13	0.41	0.35	0.41	0.36	0.32	0.28
mostfreq	A vs. not A	bow	0.35	0.16	0.68	0.35	0.50	**0.51**	0.25	0.28
mostfreq	A vs. not A	coll	0.34	0.14	0.67	0.35	0.50	**0.51**	0.25	0.27
mostfreq	B vs. not B	bow	*0.16*	*0.10*	0.58	**0.43**	0.50	**0.51**	*0.14*	*0.17*
mostfreq	B vs. not B	coll	0.17	0.12	0.59	**0.43**	0.50	**0.51**	*0.14*	0.19
mostfreq	D vs. not D	bow	0.42	0.18	0.71	0.32	0.50	**0.51**	0.29	0.31
mostfreq	D vs. not D	coll	0.50	0.14	0.75	0.27	0.50	**0.51**	0.32	0.34
mostfreq	multi	bow	0.37	0.19	**0.79**	0.32	0.43	0.50	0.27	0.36
mostfreq	multi	coll	0.37	0.14	**0.79**	0.31	*0.35*	0.48	0.19	0.28

Table 4: APPENDIX — baseline results for target term "boc" (maximum and minimum values highlighted in bold and italics, respectively)

Training algorithm	**Classification type**	**Vector type**	**Accuracy**		**Precision**		**Recall**		**F1**	
			Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
nb	A vs. not A	bow	0.73	0.13	0.74	0.25	0.77	0.21	0.71	0.16
nb	A vs. not A	coll	0.79	0.14	0.81	0.22	0.73	0.31	0.71	0.26
nb	B vs. not B	bow	0.67	**0.19**	0.69	0.36	0.74	0.30	0.60	0.27
nb	B vs. not B	coll	0.75	0.17	0.71	0.35	0.65	0.38	0.61	**0.36**
nb	D vs. not D	bow	**0.84**	0.10	**0.85**	*0.15*	**0.83**	*0.18*	**0.82**	*0.12*
nb	D vs. not D	coll	0.82	0.13	0.82	0.20	0.82	0.20	0.80	0.17
nb	multi	bow	0.63	0.16	0.65	0.37	0.62	0.35	0.56	0.33
nb	multi	coll	*0.60*	0.17	*0.64*	0.35	*0.58*	0.37	*0.52*	0.31
me	A vs. not A	bow	0.75	0.12	0.73	0.28	0.73	0.27	0.69	0.25
me	A vs. not A	coll	0.79	*0.09*	0.81	0.20	0.71	0.31	0.70	0.25
me	B vs. not B	bow	0.66	0.17	*0.64*	**0.38**	0.72	0.30	0.58	0.29
me	B vs. not B	coll	0.74	0.14	0.76	0.27	0.63	**0.40**	0.58	0.34
me	D vs. not D	bow	0.81	0.14	0.81	0.23	0.82	0.21	0.78	0.20
me	D vs. not D	coll	0.76	0.15	0.79	0.22	0.77	0.23	0.75	0.18
me	multi	bow	0.65	0.18	*0.64*	0.32	0.67	0.29	0.61	0.28
me	multi	coll	0.71	0.14	0.75	0.29	0.62	0.39	0.57	0.34

Table 5: APPENDIX — detailed results for target term "boc" (maximum and minimum values highlighted in bold and italics, respectively)

Growing trees from morphs: Towards data-driven morphological parsing

Petra Steiner and Josef Ruppenhofer
Institute of Information Science and Natural Language Processing
Hildesheim University
Hildesheim, Germany
{ruppenho,steinerp}@uni-hildesheim.de

Abstract

We present a quantitative approach to disambiguating flat morphological analyses and producing more deeply structured analyses. Based on existing morphological segmentations, possible combinations of resulting word trees for the next level are filtered first by criteria of linguistic plausibility and then by weighting procedures based on the geometric mean.

The frequencies for weighting are derived from three different sources (counts of morphs in a lexicon, counts of largest constituents in a lexicon, counts of token frequencies in a corpus) and can be used either to find the best analysis on the level of morphs or on the next higher constituent level. The evaluation shows that for this task corpus-based frequency counts are slightly superior to counts of lexical data.

1 Introduction

One of the bottlenecks for the automatic processing of German language data is word form productivity. For the specification of concepts, the creation of long compounds and derived forms is very common, e.g. (1).

(1) Oberklassenschlagbohrmaschine
'Premium class hammer drill (machine)'

While constituents of English compounds are often separated by hyphens or spaces, in German the constituents of compounds are written as a single orthographic word. Thus, the word form in (1) could be analyzed (usefully) as *Oberklasse* 'premium class', and *Schlagbohrmaschine* 'hammer drill', but also (uselessly) as *Ober* 'premium', *Klassenschlag* '*class hit', *bohr* 'drill', and *Maschine* 'machine'. Note the interfix *n* between *klasse* and *schlag*. Furthermore, some morphs can be ambiguous. E.g. *Ober* might denote a waiter, while *Schlag* might be related to the verb *schlagen* 'hit, hammer' or the noun *Schlag* 'hit, blow'. Moreover, the spelling conventions of German result in ambiguity concerning morph boundaries. E.g., *Anbaumenge* could be analyzed into the immediate constituents *Anbau* 'cultivation' and *Menge* 'amount' but also to *An* 'at', *Baum* 'tree' and *Enge* 'narrowness'.

Applications in machine translation or multilingual terminology extraction require robust methods for disambiguating and post-processing morphological analyses of German words. While some robust morphological analyzers for German exist (e.g. SMOR (Schmid et al., 2004), Gertwol (Haapalainen and Majorin, 1995), MORPH (Hanrieder, 1991; Hanrieder, 1996), TAGH (Geyken and Hanneforth, 2006)), all of them yield flat structures. However, hierarchical word structures provide important information about a word's meaning and should be taken into account as well.

Some heuristics, such as taking the analysis with the smallest number of constituents, can be used to inform the choice between multiple analyses. However, there is room for refinement and augmentation. Cap (2014) discusses a broad range of approaches to disambiguating compounds. The present contribution, in contrast, aims at dealing not only with compounding but also with derivation and other word formation processes.

Würzner and Hanneforth (2013) tackle the problem of full morphological parsing, but restricted to adjectives. They segment words into lexical units using the TAGH system of (Geyken and Hanneforth, 2006) and then use a probabilistic context free grammar for parsing. The grammar is trained on manually labeled word trees. Our approach is more general in that we cover complex words of any part of speech. It is more limited in that we do not produce a full parse.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 49–57, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

Most importantly, since the morphology system we use, SMOR, produces more segmentations per item than TAGH, we focus on disambiguation of available analyses, for which we also use corpus frequency counts, unlike Würzner and Hanneforth (2013).

Our approach starts from sets of flat analyses, builds all possible combinations of higher-level analyses, and filters these using the geometric mean (*gm*) score. In the calculation of the score, we use either frequencies derived from lexicons or frequencies derived from corpora:

a) all morphs found in a German lexicon with their frequencies within the lexicon,

b) all immediate constituents found in a German lexicon with their frequencies derived from the lexicon,

c) all immediate constituents found in a German lexicon with their frequencies taken from a German corpus.

Section 2 presents the data and their pre-processing and augmentation. Section 3 describes our gold standard. The methods for weighting and filtering the morphological analyses are presented in Section 4, which also shows our approach to handling data sparsity. The results for each of the three datasets are presented in Section 5 and discussed in Section 6. The last section comprises a conclusion with an outlook for future work.

2 Data

2.1 Augmented SMOR Analyses

SMOR is a morphological analyzer based on two-level morphology (Koskenniemi, 1984), implemented as a set of finite-state transducers (Schmid et al., 2004). For German, a large set of lexicons is available. The final version used for the current work comprises a main lexicon with 41,944 entries, proper name lexicons with 15,188 entries and different datasets with other morphological information. These lexicons contain information about inflection, parts of speech and classes of word formation (e.g. abbreviations, truncations). The tag set used is compatible with the STTS (Stuttgart Tübingen tag set, Schiller et al. (1995)).

The output for (1) with information on word formation and inflection is given in Figure 1. Please note that the interfix between *Klasse* and *schlag* has been deleted in these analyses by SMOR. Also, the STTS-like annotation contains some metatags for abbreviations and word-form parts before or between hyphenation as in example (2), which is a hyphenated variant of (1).

(2) {Oberklassen}-<TRUNC>Schlag<NN>bohren <V>Maschine<+NN><Fem><Acc><Sg>

This leaves part of the word unanalyzed and is an unwanted side effect. We therefore reanalyze results with tag <TRUNC> as follows:

a) Hyphens are removed, the letters following the hyphens are transformed to lower case. The copy is used as input of SMOR. If an analysis was found, hyphens and letters are re-inserted.

b) If only an analysis with <TRUNC> is possible, each string between hyphens is reanalyzed separately. For this process, the SMOR lexicons are reused.

This leads to analyses such as (3).

(3) Ober<PREF>Klasse<NN>n<FL>-<HYPHEN> Schlag<NN>bohren<V>Maschine<+NN>

Interfixes are restored from internal SMOR results[1] and annotated with *FL* (filler letter) as a new tag for the interfix. Table 1 summarizes the changes.

Method	*t*	*n*	*a*	*r*
(a) SMOR baseline	105	3	0	0.00
(b) remove hyphens	48	2	58	0.54
(c) reanalyze TRUNC	39	3	66	0.61
(d) combine (b) and (c)	2	2	104	0.96

Table 1: Analyzed hyphenated forms; *t* analyses containing TRUNC; *n*: hyphenated forms without analyses; *a*: correctly pre-analyzed hyphenated word form; *r*: relative frequency of *a*

We used the 1,101 items from our gold standard data (see Section 3). Of the 108 word forms containing one or more hyphens, only two were not covered by any method. The methods (b) and (c) work rather complementarily. The analyses of hyphen-removed forms are especially successful for spelling variants, such as *anti-amerikanisch* 'anti-American'. On the other hand, the lexicon-based analyses cover especially word forms which

[1] Sennrich and Kunz (2014) also add interfixes to SMOR output.

ober<PREF>Klasse<NN>Schlag<NN>bohren<V>Maschine<+NN><Fem><Acc><Sg>
ober<PREF>Klasse<NN>Schlag<NN>bohren<V>Maschine<+NN><Fem><Dat><Sg>
ober<PREF>Klasse<NN>Schlag<NN>bohren<V>Maschine<+NN><Fem><Gen><Sg>
ober<PREF>Klasse<NN>Schlag<NN>bohren<V>Maschine<+NN><Fem><Nom><Sg>
ober<PREF>Klasse<NN>schlagen<V><NN><SUFF>bohren<V>Maschine<+NN><Fem><Acc><Sg>
ober<PREF>Klasse<NN>schlagen<V><NN><SUFF>bohren<V>Maschine<+NN><Fem><Dat><Sg>
ober<PREF>Klasse<NN>schlagen<V><NN><SUFF>bohren<V>Maschine<+NN><Fem><Gen><Sg>
ober<PREF>Klasse<NN>schlagen<V><NN><SUFF>bohren<V>Maschine<+NN><Fem><Nom><Sg>

Figure 1: Output of SMOR for *Oberklassenschlagbohrmaschine*

include abbreviations, e.g. *CO2-Emissionen* 'CO2 emissions'. When hyphenated word forms which cannot be analyzed after removal of the hyphens are processed by the second algorithm, only four hyphenated word forms remain unanalyzed, due to misspelling or unusual forms that were not included in the SMOR lexicon.

Small changes to the lexicon, such as adding proper names or changing restrictions on morph positions inside words, allow for complete coverage of the observed data. The analyses are reduced to the lemma form. The sequence of the morphological information is transformed by using directed acyclic graphs, resulting in output such as (4), giving a surface form and a lexical form of the word analysis, followed by the tags.

(4)

Ober	klasse	n	schlag	bohr	maschine	
ober	Klasse	n	Schlag	bohren	Maschine	
PREF	NN	FL	NN	V	NN	<NN>

2.2 CELEX

The lexical database CELEX contains Dutch, English, and German lexical information (Baayen et al., 1995) combined with frequency information, which for German is based on counts of the *Mannheim Corpus* (Gulikers et al., 1995, 102ff.). The morphological part is of special interest for word analyses (Gulikers et al., 1995, 45ff.). The database gives information on word-formation types and provides manually annotated multi-level word structures from which flat as well as complex structures can be extracted. Special characters of German such as *ä* and *ß* are represented as *"a* and *$* in the lexical part of CELEX and had to be changed. Information about orthography is available in the database. However, it is restricted to lemmas. Therefore, the components of morphological analyses had to be adapted heuristically and were manually corrected. All ablauts which occur in irregular verbs were changed manually.

In total, our modified CELEX dataset for German has 51,727 entries. From it, three datasets with frequency information were extracted:

a) all morphs with their frequencies within the CELEX lemmas,

b) all immediate constituents with their frequencies within the CELEX lemmas,

c) all immediate constituents within the CELEX lemmas with their frequencies as found in the *Mannheim Corpus*.

For example, the lemma *Sprachwissenschaft* increments the frequencies for each of the morphs *sprech* (*Sprache* is a derivative of *sprechen*), *wissen* and *schaft* by 1. Likewise, the frequencies of its immediate constituents, *Sprache* and *Wissenschaft*, are incremented by 1. For the dataset of the text frequencies, 13 is added for each of the immediate constituents, as this is the lemma's corpus frequency. This leads to 13,419 entries for the morphs and their frequencies, and 21,406 entries for the constituents and their frequencies within the lexicon and the corpus.

The first dataset is used to choose among the best morph-level analyses, the other frequency data provide input for higher-level analyses.

3 Gold Standard

The gold standard used is based on Cap (2014, 95), who uses part of the test set of the 2009 workshop on statistical machine translation.[2] Of these 6,187 tokens, 1,101 were analyzed by human annotators, as in (5).

(5) 10-Jahres-Prognosen 10|Jahr|Prognosen
'10-year forecast'

These compounds are input for the analyses of morphological structure. The analysis of the lemmatized form with hyphens and interfixes in Cap (2014) included forms like (6), which made it necessary to create a new gold standard for our evaluation (cf. Section 5).

[2] http://www.statmt.org/wmt09/translation-task.html

(6) 10|-|Jahr|es-|Prognose

4 Methods

4.1 Word structures as Integer Compositions

The combinatorial structure of morphological analyses of a word with *n* parts is isomorphic to the permuted integer partitions of *n*. For instance, a word which is analyzed into three noun stems can be described in four different ways (7a–d). While (7a) shows an analysis of a syntagmatic compound (German: Zusammenrückung), in (7b) and (7c) the immediate constituents *Drahtseil* and *Seilakt* are identified. (7d) interprets the three-stem analysis as incorrect and amalgamates them to a monomorphemic word. The correct analysis for immediate constituents is (7c), for the smallest units (morphs) it is (7a).

(7) *Drahtseilakt* 'High-wire act'

a. [['Draht'], ['seil'], ['akt']]
b. [['Draht'], ['seilakt']]
c. [['Drahtseil'], ['akt']]
d. [['Drahtseilakt']]

The isomorphic structure of integer compositions shows the number of elements in the subsets of the sequential elements of each morphological analysis (cf. (8)). The algorithm for processing the combinatorially possible analyses makes use of this analogy.

(8) Integer compositions corresponding to the analyses in (7) above

a. 1-1-1
b. 1-2
c. 2-1
d. 3

The number of all integer compositions for *n* equals 2^{n-1} for integers $>= 1$. For *Oberklassenschlagbohrmaschine* with *n* = 5 this gives 16 compositions. The interfix does not count as a relevant morph.

However, some compositions can be ruled out as linguistically implausible, e.g. compositions starting with a suffix or ending with a prefix. This does not only reduce the number of combinatorially possible analyses but also splits the set into subsets marked by affix boundaries. E.g. some compositions for *abwechslungsreich* 'rich in variety' yield impossible subcomponents such as **ungsreich* 'SUFFIX FL full' in (9b) and (9f).

(9) Compositions of *abwechslungsreich*

a. [['ab'], ['wechsl'], ['ung', 's'], ['reich']],
b. [['ab'], ['wechsl'], ['ung', 's', 'reich']],
c. [['ab'], ['wechsl', 'ung', 's'], ['reich']],
d. [['ab'], ['wechsl', 'ung', 's', 'reich']],
e. [['ab', 'wechsl'], ['ung', 's'], ['reich']],
f. [['ab', 'wechsl'], ['ung', 's', 'reich']],
g. [['ab', 'wechsl', 'ung', 's'], ['reich']],
h. [['ab', 'wechsl', 'ung', 's', 'reich']]

As prefixes and verb particles form a natural boundary within morphological analyses, the combinatorial path has to be pruned. For instance, if *Benutzerunterstützung* 'user support' is analyzed as in (10) - other analyses are possible - *unter* marks a boundary. After building all combinations for each of the subsets {*'Be' 'nutz' 'er'*} and {*'unter' 'stütz' 'ung'*}, the Cartesian product of the resulting combinations has to be produced. The final sets of morphs and morph combinations are input for the weighting procedures.

(10)

Be	nutz	er	unter	stütz	ung
VPREF	V	NNSUFF	VPART	V	NNSUFF
be.pref	use	er.suff	below	support	ung.suff

4.2 Geometric Mean Score

Cap (2014, 67) uses the geometric mean as a quality measure for the analysis of German compounds. She uses the logarithmic transformation which is based the model of Koehn and Knight (2003). We use the non-transformed geometric mean as in (11) as the log-transformation preserves the ordering of the non-transformed value.

$$\left(\prod_{i=1}^{n} x_i\right)^{1/n} \textit{for} \quad x_i ... x_n, \tag{11}$$

For the morph analysis of *Anbaumenge* to *An|bau|Menge* the respective morph frequencies are x_1 = 845 for *an*, x_2 = 168 for *bau* and x_3 = 8 for *Menge*, resulting in a *gm* score of 104.33. However, it is possible that the analyzed part *An|bau* is actually wrong and *Anbau* is the smallest unit that could be found. The same could hold for an analysis to *An|Baumenge*. However, the frequencies for these alternatives are lower than those of the first analysis (see Table 2).

4.3 Data Sparsity

The last example showed a case where a low frequency was consistent with linguistic reality. If the frequency of an element, whether a simple morph

or an amalgamated form, is very small or 0, this can have two reasons: a. the form does not exist, or b. the form exists but is not present in the lexicon or in the underlying corpus. For example, the analysis of *10-Jahres-Prognosen* into its three lexical morphs would be impossible as numbers are not included in the lexicon. In both cases, the geometric mean would be undefined. However, especially for the second case, it is sensible to assign a small value to the element. Here, we chose 0.1. For a set of analyses which consists exclusively of unknown parts, this has the effect that the analysis with the smallest number of elements is chosen. This heuristic filters out longish pseudo-analyses which consist of highly frequent short words.

4.4 Heuristics for Parts of Speech

As surface and lexical forms of the two-level morphology might differ, we look up each morph or constituent candidate in both representations. For morphs which are the first part of the analysis, the lower case version has to be looked up, the opposite is necessary for nouns whose lexical form is represented with upper case, while their surface form might have lower case, if the noun is a non-initial component. SMOR produces the infinitive as the output for verbal morphs on the lexical level. However, for noun derivations with suffixes, the surface form of the verb stem is more relevant. After hyphens, the surface forms can start with a capital letter. Still other restrictions hold for abbreviations. A simple look-up heuristic deals with these different conditions.

5 Outcome and Evaluation

The following evaluation comprises qualitative and quantitative parts for each of the lexicons used. In the qualitative part, we consider cases of non-trivial analyses. The quantitative part presents results in terms of recall against the gold standard.

The qualitative test set covers three problems of disambiguation: a. ambiguity of morph boundaries, b. unknown parts of the analysis and c. ambiguous word structure.

For a. we choose the word forms

- *Anbaumenge* with the analyses *An|bau|menge* '(at|build|amount)' and **An|Baum|Enge* '(at|tree|narrowness)'
- *Benzinverbrauch* with the analyses *Benzin|ver|brauch* '(petrol|(PREF)|use)' and **Benzin|Verb|Rauch* '(petrol|verb|smoke)'
- *Aufbewahrungsorten* with the analyses of the immediate constituents **Aufbewahrung|sorte* '(storage|class)' and *Aufbewahrung|s|orte* '(storage|(FL)|places)'; *Aufbewahrung* as a derived form can be analyzed as a complex multi-prefixed and suffixed form.

For b. we choose

- *10-Jahres-Prognosen* with the analysis '(10|-|Jahr|es-|Prognose)' where *10* is unknown.

As an example for c., ambiguous structures, we choose

- *Arzneimittelverkaufs* with the noun constituents (*Arznei|mittel|verkauf)* '(medicine|means|sale)'. However, the next level of the morphological tree could either be *((Arznei|mittel)|verkauf)* '(medicine means|sale)' or *(Arznei|(mittel|verkauf))* '*(medicine|means sale)'

For the quantitative evaluation, 50 percent (initial letters A to L) of the output of the system across the testset was evaluated by two humans. The data comprises 1,290 analyses of 572 word forms. These analyses are the ones representing the compositions with the highest score for a given item. No lower-ranking analyses are taken into account. For these analyses with largest scores, we annotated three cases:

***** for wrong segmentations (false positive)

? for segmentations which were correct but on the "wrong level" (meaning higher-constituent analyses for morph analyses, or morph analyses instead of constituent analyses) (weak positive)

- for a correct segmentation (true positive, Recall)

We only considered the segmentation of the strings and ignored dubious tag assignments. However, if two analyses for the same word form got the same highest score and one of them was wrong, we marked this with *.

5.1 Morph Frequencies

5.1.1 Qualitative Analysis

Table 2 presents the output of the analyses. The morph analysis of *Anbaumenge* yields five different (flat) analyses from SMOR which can be combined into 16 plausible complex constructions. For each of the five SMOR results, the combinatorial analysis with the highest *gm* score is chosen.

word	gm	# of analyses	lexical analysis	surface analysis	tag structure
Anbau-	104.33	2	an|bauen|Menge	An|bau|menge	(VPART)(V|NNSUFF)(NN)
menge	104.33	4	an|bauen|Menge	An|bau|menge	(VPART)(V)(NN)
	12.66	4	an|baumen|eng	An|baum|eng	(VPART)(V)(ADJ|NNSUFF)
	9.19	4	an|baumenEnge	An|baumenge	(VPART)(V|NN)
	0.89	2	Anbau|Menge	Anbau|menge	(NN)(NN)
Benzin-	12.00	4	Benzin|Verb|Rauch	Benzin|verb|rauch	(NN)(NN)(NN)
verbrauch	0.63	2	Benzin|Verbrauch	Benzin|verbrauch	(NN)(NN)
Auf-	9.35	2	auf|bewahrenung|Sorte	Auf|bewahrung|sorte	(VPART)(V|NNSUFF)(NN)
bewahrungs-	15.53	4	auf|bewahrenung|s|Ort	Auf|bewahrung|s|ort	(VPART)(V|NNSUFF|FL)(NN)
orten	8.25	4	auf|bewahrenungsorten	Auf|bewahrungsorten	(VPART)(V|NNSUFF|FL|V|NNSUFF)
10-Jahres-	2.56	4	10|-|Jahr|es-|Prognose	10|-|Jahr|es-|Prognose	(PREF|HYPHEN)(NN|FL|HYPHEN)(NN)
Arznei-	80.52	4	Arznei|Mittel|ver|kaufen	Arznei|mittel|ver|kauf	(NN)(NN)(VPREF)(V|NNSUFF)
mittel-	3.35	4	Arznei|Mittel|verkaufen	Arznei|mittel|verkauf	(NN)(NN)(V|NNSUFF)
verkaufs	3.35	4	Arznei|Mittel|Verkauf	Arznei|mittel|verkauf	(NN)(NN)(NN)
	56.01	4	Arznei|mittel|ver|kaufen	Arznei|mittel|ver|kauf	(NN)(ADJ)(VPREF)(V|NNSUFF)
	2.07	4	Arznei|mittel|verkaufen	Arznei|mittel|verkauf	(NN)(ADJ)(V|NNSUFF)
	62.07	4	Arznei|mittel|Verkauf	Arznei|mittel|verkauf	(NN)(ADJ)(NN)
	22.36	2	Arzneimittel|ver|kaufen	Arzneimittel|ver|kauf	(NN)(VPREF)(V|NNSUFF)
	0.10	2	Arzneimittel|verkaufen	Arzneimittel|verkauf	(NN)(V|NNSUFF)
	0.10	2	Arzneimittel|Verkauf	Arzneimittel|verkauf	(NN)(NN)

Table 2: Output of morph analyses with *gm* score, number of compositions, lexical analysis, surface analysis and tag structure

It can easily be seen that the wrong analysis with *An|baum|enge* in the third line has a far lower score than the correct analysis. Another analysis based on the verb *baumen* 'to sit on a tree' and *Enge* 'narrowness' also gets a low score. The immediate constituents have a very low score, as *Anbau* is not part of the set of known morphs and only gets the back-off value of 0.1. The output for *Benzinverbrauch* faces the problem that SMOR does not segment the derived form *Verbrauch*. However, this word form is not part of the morph lexicon so that the analysis wrongly gives the best score to **Benzin|Verb|Rauch* '(petrol|verb|smoke)' For *Aufbewahrungsorten*, the correct SMOR analysis gets the highest score. However, the score for the incorrect analysis **Auf|bewahrungsorte* '(VPART| keeping class)' is surprisingly high, which is due to the high frequency of the verb particle *auf* which is multiplied by the sparse data value 0.1. The word form with the unknown number, *10-Jahres-Prognosen*, is correctly analyzed out of four different compositions. Due to the sparse data value, the *gm* score for each of these compositions can be calculated and compared. Finally, the example for ambiguous structures *Arzneimittelverkaufs* yields 9 SMOR analyses with 30 plausible combinatorial analyses. As can be seen from the last block of Table 2, the correct morph analysis gets the highest score. Note that the segmentation in line four with the second-largest score is also correct. However, it is based on an incorrect POS-assignment: *mittel* is analyzed as an adjective ('middle') instead of a noun.

5.1.2 Quantitative Analysis

For 572 word forms, we found 38 wrong segmentations and 70 cases which were correct annotations, though not on the expected morphological level. This leads to a recall of 81.11 percent. The number of different combinatorial analyses available was not taken into account.

About a third of the incorrectly analyzed word forms are of the type *An|passungsmechanismus*, where a high-frequency prefix determines the high score of a mostly unanalyzed word form.

Regarding the weak recall, some morph analyses are simply not feasible as the SMOR output does not always yield the smallest lexical units.

5.2 Frequencies of Constituents

5.2.1 Qualitative Analysis

The constituent analysis of *Anbaumenge* yields the same best analysis as the morph analysis. The numbers are slightly different but the score for the segmentation *Anbau|menge* is outweighed by that for *An|bau|Menge* due to the high frequencies of *an* and *bau*. The first part of Table 3 presents the output of the analyses.

The output for *Benzinverbrauch* is shown in the second part of Table 3. As with the morph-

word	gm	# of analyses	lexical analysis	surface analysis	tag structure
Anbau-	59.80	2	an|bauen|Menge	An|bau|menge	(VPART)(V|NNSUFF)(NN)
menge	53.70	4	an|bauen|Menge	An|bau|menge	(VPART)(V)(NN)
	8.54	4	an|baumen|eng	An|baum|eng	(VPART)(V)(ADJ|NNSUFF)
	6.05	4	an|baumenEnge	An|baumenge	(VPART)(V|NN)
	4.90	2	Anbau|Menge	Anbau|menge	(NN)(NN)
Benzin-	6.51	4	Benzin|Verb|Rauch	Benzin|verb|rauch	(NN)(NN)(NN)
verbrauch	4.00	2	Benzin|Verbrauch	Benzin|verbrauch	(NN)(NN)
Auf-	5.30	2	auf|bewahrenung|Sorte	Auf|bewahrung|sorte	(VPART)(V|NNSUFF)(NN)*
bewahrungs-	12.10	4	auf|bewahrenung|s|Ort	Auf|bewahrung|s|ort	(VPART)(V|NNSUFF|FL)(NN)
orten	6.11	4	auf|bewahrenungsorten	Auf|bewahrungsorten	(VPART)(V|NNSUFF|FL|V|NNSUFF)
10-Jahres-	2.29	4	10|-|Jahr|es-|Prognose	10|-|Jahr|es-|Prognose	(PREF|HYPHEN)(NN|FL|HYPHEN)(NN)
Prognosen					
Arznei-	43.4	4	Arznei|Mittel|ver|kaufen	Arznei|mittel|ver|kauf	(NN)(NN)(VPREF)(V|NNSUFF)
mittel-	12.80	4	Arznei|Mittel|verkaufen	Arznei|mittel|verkauf	(NN)(NN) (V|NNSUFF)
verkaufs	12.80	4	Arznei|Mittel|Verkauf	Arznei|mittel|verkauf	(NN)(NN) (NN)
	29.50	4	Arznei|mittel|ver|kaufen	Arznei|mittel|ver|kauf	(NN)(ADJ)(VPREF) (V|NNSUFF)
	7.65	4	Arznei|mittel|verkaufen	Arznei|mittel|verkauf	(NN)(ADJ) (V|NNSUFF)
	7.65	4	Arznei|mittel|Verkauf	Arznei|mittel|verkauf	(NN)(ADJ) (NN)
	10.6	2	Arzneimittel|ver|kaufen	Arzneimittel|ver|kauf	(NN)(VPREF) (V|NNSUFF)
	0.84	2	Arzneimittel|verkaufen	Arzneimittel|verkauf	(NN)(V|NNSUFF)
	0.84	2	Arzneimittel|Verkauf	Arzneimittel|verkauf	(NN)(NN)

Table 3: Output of constituent analyses with gm-score, number of compositions, lexical analysis, surface analysis and tag structure

based analysis, the constituent analyses wrongly gives the best score to **Benzin\|Verb\|Rauch* (petrol\|verb\|smoke). The respective frequencies of the constituents within the CELEX lexicon are (4, 3, and 23) vs. (4 and 4), so the segmentation into three parts is preferred. The third part of Table 3 presents the results for *Aufbewahrungsorten*. While the highest rank remains the same, there is an increase in the scores as the immediate constituent counts increment the number of longer words at the cost of the shorter ones. The analysis of *10-Jahres-Prognosen* is the same as for the morphs, as all constituents are monomorphemes. Due to the different counts, the *gm* score differs slightly. The analysis of ambiguous structures for *Arzneimittelverkaufs* can be seen in the last part of Table 3. Though closer to each other, the scores are in the same order as for the morph analyses. This unwanted result is caused by the low frequency for the constituent *Arzneimittel* in CELEX.

5.2.2 Quantitative Analysis

We found 22 wrong segmentations and 86 weak positive ones. The word forms concerned were the same as for the morphs, which results in the same overall recall. Sometimes the sequence of the *gm* scores was a bit closer to the correct order, however this frequency count shows that constituent counts from a lexicon of an acceptable size are not good enough for analyzing these cases. It is of some linguistic irony that the weak positive annotated analyses are mostly good analyses for the morph level or another low-level description, e.g. the *Fuß\|ball\|national\|team* 'national football team' was correctly analyzed.

Among the wrongly analyzed forms we encounter the above-described effect of too dominant prefixes. Segmentations such as **Benzin\|Verb\|Rauch* are rather an exception.

5.3 Corpus Frequencies

5.3.1 Qualitative Analysis

As the frequency counts for the corpora are higher than those for the lexicons, the scores also become larger and tend to differ more significantly. Table 4 shows the results for *Anbaumenge*. Obviously, the ranks are determined by the high frequencies of prefixes and verb particles.

Benzinverbrauch is analyzed correctly, though the tag structure reveals that the analysis was produced by merging *Verb* with *rauch* to *Verbrauch*. *Aufbewahrungsorte* and *10-Jahres-Prognosen* yield good results too. *Arzneimittelverkaufs* is perfectly analyzed for the morph level. In general, the corpus-based analyses of constituents and morphs produce more interpretable results.

gm	analyses	lexical analysis	surface analysis	tag structure
6075.87	2	an|bauen|Menge	An|bau|menge	(VPART)(V|NNSUFF)(NN)
6075.87	4	an|bauen|Menge	An|bau|menge	(VPART)(V)(NN)
209.14	4	an|baumen|eng	An|baum|enge	(VPART)(V)(ADJ|NNSUFF)
85.27	4	an|baumenEnge	An|baumenge	(VPART)(V|NN)
114.60	2	Anbau|Menge	Anbau|menge	(NN)(NN)

Table 4: Output of word-form analysis for *Anbaumenge* with gm-score, number of compositions, lexical analysis, surface analysis and tag structure

5.3.2 Quantitative Analysis

Scores for the word-form frequencies differed from the lexical frequencies and the results were improved. The errors that remain include syntagmatic compounds such as *50-jährig* which are erroneously segmented as endocentric compounds, e.g. *50*|-|*jährig* '50|-|year+suffix'. Even ambiguous forms on the morph level (*Benzinverbrauch*) are segmented correctly on the string level – though their morphological analyses remain erroneous.

Table 5 gives an overview of the evaluation with the overall recall and the recall for the weakly consistent analyses.

dataset	*	?	overall recall	weak recall
morphs	38	70	81.11	93.34
constituents	22	86	81.11	96.15
word forms	15	88	88.02	97.38

Table 5: Overall recall and weak recall for three frequency sources

6 Discussion

The approach we presented shows how different frequency counts can lead to different specific constituent segmentation analyses. The corpus frequencies in particular lead to better segmentation on the morph level.

Ideally, we would derive counts from corpus data that match the register and domain of the lexical units that are to be analyzed. Here, frequencies derived from a corpus of 6.0 million words (Gulikers et al., 1995, 102) were too small to yield reliable counts for non-monomorphic constituents. Larger sets of well-annotated corpus data should be used. Moreover, the analysis process can be augmented by other linguistic characteristics: parts of speech, position of constituents in words, and the text specificity of words.

As our approach builds on the output of a morphological segmentizer, it is dependent on the prior segmentation, for better or for worse. Starting with analyses of different morphological tools might help to avert, or compensate for, gaps in the lexicon or systematic weaknesses in the tools. In general, morphological data should be analyzed from many sides.

The use of the geometric mean should be analyzed from a quantitative point of view. In particular, the distribution of values should be investigated to make statements about their relevance. When weighting alternatives of n vs. $n+1$ constituents, in most cases the geometric mean for the variant with more constituents is larger than that for the variant with fewer constituents. This is owed to the facts that a) shorter morphs and lexical units are more frequent than longer ones and b) corpus frequencies for compounds or derivates are still relatively small compared to the frequencies of their constituents. Restricting the cotext to smaller units than the corpus, such as the document or paragraph, could help, although in that case the data might become too sparse.

7 Conclusion

This investigation has shown that ambiguous flat structure results from morphological analyses can be disambiguated by using additional statistical methods, especially on the morphological level.

What could not be analyzed on a lower level should be re-analyzed by using the same combinatorial approach on a higher-level analysis. The current state of work already shows some more complex morphological structures. However, as the geometric mean is not a good indicator for the concatenation of morphs, the weighting measure(s) should be derived carefully. In future work, we will explore probabilistic models. In combination with such models, the sets of integer compositions for the analyses of one word form can be processed in transition networks. Also, we will focus on building up more levels of the word-structure trees and exploiting the statistical dependencies between morphs and their parts of speech.

Acknowledgments

The authors were partially supported by the German Research Foundation (DFG) under grant RU 1873/2-1. We would like to thank especially Helmut Schmid for his valuable advice and updates of the SMOR lexicons and Jasper Brandes for his help in the evaluation task.

References

Harald Baayen, Richard Piepenbrock, and Léon Gulikers. 1995. The CELEX lexical database (CD-ROM).

Fabienne Cap. 2014. *Morphological processing of compounds for statistical machine translation*. Ph.D. thesis, Universität Stuttgart.

Alexander Geyken 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, pages 55–66. Springer.

Léon Gulikers, Gilbert Rattink, and Richard Piepenbrock. 1995. German Linguistic Guide. In Harald Baayen, Richard Piepenbrock, and Léon Gulikers, editors, *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, Philadelphia, PA.

Mariikka Haapalainen and Ari Majorin. 1995. GERTWOL und morphologische Disambiguierung für das Deutsche. In *Proceedings of the 10th Nordic Conference on Computational Linguistics, Helsinki, Finland*.

Gerhard Hanrieder. 1991. Robustes Wortparsing. Lexikonbasierte morphologische Analyse (komplexer) deutscher Wortformen. Master's thesis, Universität Trier.

Gerhard Hanrieder. 1996. MORPH - Ein modulares und robustes Morphologieprogramm für das Deutsche in Common Lisp. In Roland Hauser, editor, *Linguistische Verifikation Dokumentation zur Ersten Morpholymics 1994*, pages 53–66. Niemeyer, Tübingen.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 187–193. Association for Computational Linguistics.

Kimmo Koskenniemi. 1984. A general computational model for word-form recognition and production. In *Proceedings of the 10th international conference on Computational linguistics*, pages 178–181. Association for Computational Linguistics.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1995. Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, and Seminar für Sprachwissenschaft, Universität Tübingen.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Rico Sennrich and Beat Kunz. 2014. Zmorge: A German Morphological Lexicon Extracted from Wiktionary. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014*, pages 1063–1067.

Kay-Michael Würzner and Thomas Hanneforth. 2013. Parsing morphologically complex words. In Mark-Jan Nederhof, editor, *Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing, FSMNLP 2013, St. Andrews, Scotland, UK, July 15-17, 2013*, pages 39–43. The Association for Computer Linguistics.

Rule-based Dependency Parse Collapsing and Propagation for German and English

Eugen Ruppert and **Jonas Klesy** and **Martin Riedl** and **Chris Biemann**
FG Language Technology, Computer Science Department
Technische Universität Darmstadt
{eugen.ruppert,riedl,biem}@cs.tu-darmstadt.de, jonas.klesy@googlemail.com

Abstract

We present a flexible open-source framework that performs dependency parsing with collapsed dependencies. The parser framework features a rule-based annotator that directly works on the output of a dependency parser. Thus, it can introduce dependency collapsing and propagation (de Marneffe et al., 2006) to parsers that lack this functionality. Collapsing is a technique for dependency parses where words, mainly prepositions, are elevated into the dependency relation name. Propagation assigns syntactic roles to all involved items in conjunctions. Currently, only the Stanford parser features these abilities for the English language. Here we introduce a rule-based collapsing engine that can be applied on top of the output of a dependency parser and that was used to re-engineer the rules of the English Stanford parser. Furthermore, we provide the first dependency parser with collapsing and propagation for German. We directly compare our collapsing for English with the one from the Stanford parser. Additionally, we evaluate collapsed and non-collapsed syntactic dependencies extrinsically when used as features for building a distributional thesaurus (DT).

1 Introduction

Dependency parsing is a major pre-processing step for many applications like similarity computation, machine translation or semantic parsing. In de Marneffe et al. (2006), a technique called dependency collapsing was introduced into the Stanford parser. Collapsing is the process of reducing the number of dependencies, by inserting mostly function words into dependency relation names. Considering the sentence *They sit in the car.*, a dependency parser produces two dependency arcs between *sit* and *car* (cf. Figure 1). The first dependency arc connects *sit* to *in* – `prep(sit,in)`, the second connects *in* to *car* – `pobj(in,car)`. These relations offer only limited information; you can *sit in* something, and you can do something *in cars*. Collapsing makes the relation more informative, `prep_in(sit,car)`, indicating that you can *sit in a car*. Due to the enriched information and larger disambiguation capability of the relation, collapsed dependency parsing is often used for word sense disambiguation (Lin, 1997) and for computing similarities between terms (Biemann and Riedl, 2013).

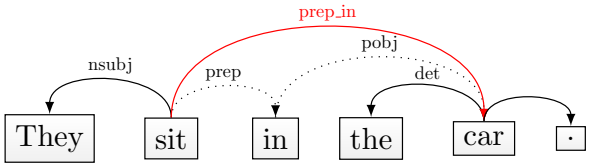


Figure 1: Dependency parser output of the sentence *They sit in the car.* Solid and dotted black lines indicate the standard dependencies, red lines indicate collapsed dependencies. Dotted black lines represent dependencies that are removed after collapsing.

Still, most work regarding collapsing has been done for English and only as variations of the Stanford parsing. Here we introduce a rule-based dependency collapsing framework. We contribute a ruleset for English and also, to our best knowledge, the first German collapsing ruleset.

The collapsing is performed on top of the parser output using our collapsing engine. Due to the rule-based nature of our engine, we can also generate rules for more complex collapsing strategies like dependency propagation. Propagation of dependencies is used to transitively propagate dependencies using conjunctions (de Marneffe et al., 2006). Words that are connected by a conjunction receive

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 58–66, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

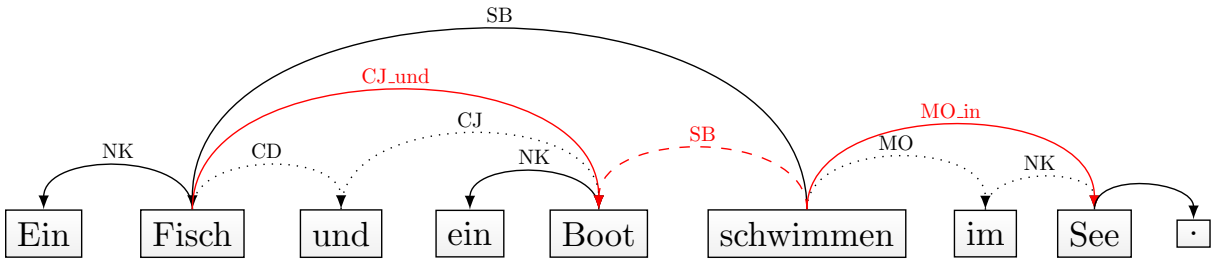


Figure 2: Depencency parser output of the sentence *Ein Fisch und ein Boot schwimmen im See. (A fish and a boat are swimming in the lake.)* Solid and dotted black lines indicate the standard dependencies, solid red lines indicate collapsed dependencies. Dotted black lines represent dependencies that are removed after collapsing. Dashed red line represents the propagated dependency, where the subject (SB) dependency was propagated to *Boot*, as it is connected via the conjunction *und* with *Fisch*, the subject identified by the parser.

the same dependencies, e.g. in Figure 2, *Fisch (fish)* and *Boot (boat)* are both subjects (SB) of the verb *schwimmen (to swim)*, even though only *Fisch* is directly connected to the verb in the original parse.

Furthermore, this approach allows rulesets to be adapted to the tagsets and dependency relation types of different parsers, allowing to apply the functionality to different parsers. We demonstrate the language independence of our framework by transforming the English ruleset to be applicable to German dependency parses. The impact of collapsing and dependency propagation is shown extrinsically based on a distributional similarity evaluation for different corpus sizes. Our collapsing technique is applied on uncollapsed Stanford output to demonstrate the quality of our collapsing rules and to compare the results with its built-in collapsing rules.

2 Related Work

Dependency parsing can help many linguistic applications, especially if they are geared towards semantic representation of text. Since sentences are represented as a lattice of dependencies, mostly originating from the verb, different surface forms of a sentence can result in the same dependency graph. This is important for languages with a variable word order (Dubey and Keller, 2003). In German, the sentence *I have seen the dog*, can have two surface forms: *Den Hund habe ich gesehen (the dog have I seen)* and *Ich habe den Hund gesehen (I have the dog seen)*, even though the first form is marked.

Klein and Manning (2003) introduced the Stanford parser, a dependency parser that extracts dependencies from probabilistic context free grammar (PCFG; Johnson, 1998) constituent parses. Dependency collapsing for this parser was added in (de Marneffe et al., 2006). The Stanford dependency representation (de Marneffe and Manning, 2008) connects two words with a directed and typed dependency relation.

Dubey and Keller (2003) introduced the first PCFG parser for German. The Mate-tools parser (Bohnet, 2010; Seeker and Kuhn, 2012) currently offers one of the best dependency parsing performances for German. However, it does not feature dependency collapsing. Therefore we introduce collapsing for German on top of the Mate-tools parser output.

Our collapsing engine is based on the UIMA framework (Ferrucci and Lally, 2004; Ogren and Bethard, 2009). While there exist generic frameworks to apply rules to UIMA annotations such as UIMA Ruta (Kluegl et al., 2014), we have opted to develop our own processing, in order to be able to tailor our framework specifically for the needs of dependency collapsing and propagation.

Dependency parsing is often used to generate features for information extraction tasks. E.g. in the event extraction task of BioNLP'09 (Kim et al., 2009), 80 % of the participants and all of the top performing teams used dependency features. The best team (Björne et al., 2009) directly worked on the dependency graph to identify relations and extract events. This could be facilitated with our dependency processing framework, where lists of trigger words and protein names can be identified with their relations.

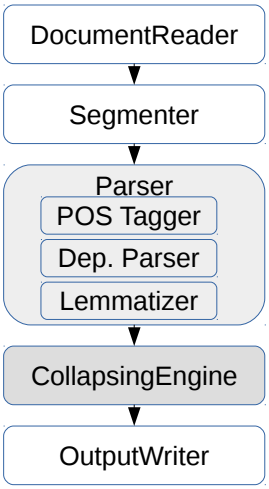


Figure 3: Processing pipeline in the UIMA framework, with exchangeable annotator components

3 Method

3.1 Overview

Our framework works with parsers for any language that produce uimaFIT (Ogren and Bethard, 2009) annotations and transforms the results according to the custom defined rules given in the rules file. We use the framework to perform dependency collapsing and propagation for the output of the Stanford parser and the German Mate-tools parser.

The UIMA framework allows to create flexible annotation pipelines, where each component can be exchanged by other components. This enables users to exchange components like tokenizers and parsers to create individual processing pipelines. In Figure 3 we show the pipeline used in our framework. Documents are read by a document reader and first segmented into sentences and tokens. Afterwards, we run a dependency parser component (that can include a part-of-speech tagger and a lemmatizer) and apply the collapsing engine on the dependency parser annotations. Last, the collapsed dependencies are written to the disk i.e. in the CoNLL format. The pipelines for English and German are freely available for download[1].

3.1.1 Collapsing Engine

The main contribution of our collapsing framework is the collapsing engine that operates on UIMA annotations produced by dependency parsers[2]. It uses rule files that define collapsing triggers and the instructions for dependency collapsing[3]. A collapsing trigger can be defined on word, lemma and POS level and the relationship of multiple tokens. If a match is found by the collapsing engine, the system applies the operations defined in the rule file to the parse, most commonly creating a new dependency with a specified name.

Since we have found it advantageous to perform collapsing and propagation in several stages, we have introduced the capability to create different stages that are executed in succession. Each stage matches dependencies and creates new ones, as specified in a rule. The matched dependencies can be kept for later stage or marked to be removed after the stage is finished.

3.2 Rule Format

3.2.1 Stages

A rule file consists of multiple stages that are executed sequentially. Each stage consists of a set of rules that are applied synchronously. Every rule has access to the dependency annotations that were present when the previous stage finished. Multiple stages are needed, if an already processed annotation should be further processed. For example, in one stage the collapsed dependency `prep_such` is generated from the phrase *animals such as birds*. In the next stage, this should be refined to `prep_such_as`, which creates a direct dependency relation between *animals* and *birds*.

3.2.2 Rules

A rule has the following scheme:

```
##<Rule name>
<element><ID>:<regex>
<r,d><FromID>_<ToID>:<regex>
<element><ID>:<regex>
...
from:<FromID>
to:<ToID>
relationName:name
```

It defines a collapsing match and the new dependency annotation that is created from this match. A collapsing match is defined between multiple `element` items and their relations. An element

[1] We provide the framework with rulesets for English and German under the permissive ASL 2.0 license at `http://jobimtext.org/dependency-collapsing/`

[2] The DKPro Core framework (Eckart de Castilho and Gurevych, 2014) provides UIMA wrappers for a large number of NLP components.

[3] Dependency propagation is also defined in the rulesets. The framework allows to create and modify custom operations on the dependency graph. For brevity, we only mention dependency collapsing in the rule description.

can be matched by word (w), POS tag (p) or a lemma (l). Matching of elements is performed using regular expressions and thus allows any matching, i.e. lists or string matches. Each element also contains an integer ID that is used to identify relations and to create new ones. Relations between elements (specified by ID), are matched by a relation name, which can also be defined as a regular expression.

If all of the conditions are met, then a new relation with a custom relationName is created between the elements specified in from and to. A matched relation will be removed after a stage, if it is identified as a removable relation (r). Durable relations (d) are available in subsequent stages.

Dependency Collapsing Here, we present an exemplary rule file for dependency collapsing:

```
STAGE:CollapsePrepositions
{
  ##prep_in Rule
  w1:.*
  r1_2:prep
  p2:IN|TO|VBG
  r2_3:pobj
  w3:.*
  relationName:prep_{w2}
  from: 1
  to: 3
}
```

In this example ruleset that collapses prepositions, only one stage is defined and in this stage only one rule exists. This rule will match all dependencies that have the dependency type prep, originate from any word (w1:.*) and end in a word with any of the specified POS tags (p2:IN|TO|VBG). Additionally, there should be another pobj dependency from p2 to any other word (w3:.*).

If such a match is found, the collapsing engine will create a new relation between w1 and w3, collapsing the preposition word into the relation name (relationName:prep_{w2}). And as the matched dependencies are marked as 'removable' (*r*1_2:prep, *r*2_3:pobj), they will be deleted after the stage is finished. To keep the dependency for later stages, the relation should be specified as durable, e.g. *d*1_2:prep.

Dependency propagation A rule set for propagation is shown below. It matches any words (wN:.*) with N=1,2,3 that form the following dependency constellation: there is a dobj or nsubj relation between words 1 and 2; additionally there should be a conjunction (conj) between words 2 and 3. If these conditions apply, the same relation as between words 1 and 2 (relationName:d1_2) is propagated as a new relation between w1 and w3.

```
STAGE:Propagation
{
  ##subj/obj propagation
  w1:.*
  d1_2:dobj|nsubj
  w2:.*
  d2_3:conj.*
  w3:.*
  relationName:{d1_2}
  from:1
  to:3
}
```

3.3 Rulesets

We have compiled rulesets for English and German, each with and without dependency propagation. For re-engineering the collapsing rules for English, we compared the collapsing from the Stanford parser to our collapsing rules and stopped after we achieved sufficient overlap. Starting from preposition collapsing, we addressed the largest error class in each rule writing stage to identify more complex dependency matchings. Propagation rules were added in an additional stage.

German rules are corresponding to the English rules, with adjusted dependency types, and – for lexicalized rules – translated words. In few cases, the parsers produce slightly different dependency structures (e.g. switch of governor / dependent). Also, some rules only apply for English and are thus left out of the German ruleset.

Overall, the rulesets now feature 3 stages for collapsing and 4 stages for collapsing with propagation. For English, we have compiled 43 rules for collapsing (46 with propagation). For German, there are 26 rules for collapsing (29 with propagation).

4 Evaluation

To evaluate our collapsing tool and the rulesets, we performe two kinds of evaluation. First, we intrinsically compare to the English Stanford parser[4]. Second, we use the dependency parser output for similarity computation (see Section 5). This allows us to show the positive impact of collapsing and dependency propagation on semantic tasks.

[4] As there exists no parser that performs collapsing for German, we cannot compare to a German parser intrinsically.

Table 1: Number of dependencies of our method and Stanford dependency parsing

	both	not in our method	not in Stanford
collapsed dep.	27,602	405	664
all dep.	165,594	1,067	1,850

Table 2: Number of dependencies of our method and Stanford dependency parsing, with dependency propagation

	both	not in our method	not in Stanford
propagated dep.	29,271	611	566
all dep.	168,201	7,377	4,036

4.1 Intrinsic Evaluation of the English Ruleset

As there is no evaluation set for collapsing available, we perform an intrinsic evaluation. We compare directly to the Stanford parser on a sample of 10,000 unannotated English sentences. On these sentences, we perform dependency parsing using 1) the Stanford dependency parser with collapsing and 2) the Stanford dependency parser without any collapsing, where we apply the collapsing using our tool.

From these sentences, 927 do not contain any collapsed dependency. The number of dependency relations is reduced from 200,541 to 194,669 with collapsing from the Stanford dependency parser. For dependency collapsing, we observe that our method only misses 1.45 % of the collapsed dependencies (see Table 1). Additionally, we observe that our collapsing engine performs collapsing more often than the Stanford collapsing, e.g. we always collapse prepositions, which is often not performed by the Stanford collapsing. On the other hand, we most commonly miss collapsed dependency relations which collapse more than one word into the relation name, like `prep_away_from` or `prep_out_of`.

In a further analysis we checked the different dependency types for each sentence. For this, we extracted all differences between the Stanford collapsed dependencies and our collapsed dependencies and counted how often they occur. Manually checking the most frequent 100 of these differences (occurring in 592 sentences), we figured out that in 27 of these discrepancies (represented by 32 sentences) we could not decide which system performs better, when checking instances manually. For 72 dependency patterns (204 sentences), the Stanford collapsing performs better, whereas our system yields better results for 78 patterns (168 sentences).

On the basis of sentences, this gives a balanced picture of our system, with Stanford being correct in 50.6 % of the cases and 41.6 % where our system is correct. Our system often misses collapsed dependency relations that collapse more then one word into the relation name. On the other hand, the Stanford collapsing often misses collapsing the construct *according to* into the dependency type, which is performed more often with our system.

We also analyzed the performance of dependency propagation, as shown in Table 2. Here we miss slightly more dependencies and we also observe that the total number of dependencies increases. This is because most propagated dependencies are added to the dependency graph, so that mostly no other dependencies are deleted by propagation.

4.2 Extrinsic Evaluation on Similarity Computation

We follow Riedl et al. (2014) to evaluate the performance of collapsing extrinsically: We use dependency features to build distributional thesauri (DT) and evaluate the thesauri against the lexical resources WordNet (Fellbaum, 1998) and GermaNet (Hamp and Feldweg, 1997; Henrich and Hinrichs, 2010). Evaluation is performed by the WordNet-based Path measure (Pedersen et al., 2004). The Path measure is the inverse path length between two words in WordNet / GermaNet. We use a word list of 1000 frequent and 1000 infrequent nouns for each language. The word lists were sampled from the corpora with the requirement that they are also present in WordNet / GermaNet. For English, we use the same words that were also used in Weeds and Weir (2003). From the DT entries of these words, we extract the top 5 similar words and calculate the average inverse path length. Since the similarities in a DT entry should be high, this means that the path length should remain relatively low, e.g. synonyms (length 1, same synset), hyponyms/hypernyms (length 2, term – hypernym / hyponym) or co-hyponyms (length 3, term – hypernym – related term). The Path measure

Table 3: Extracted context features for *Mary and John sit in the car*, using the Stanford parser, without collapsing

Term	Context Feature	
Lemma	Lemma	Dependency
Mary#NP	sit#VB	-nsubj
Mary#NP	John#NP	conj
Mary#NP	and#CC	cc
and#CC	Mary#NP	-cc
John#NP	Mary#NP	-conj
sit#VB	Mary#NP	nsubj
sit#VB	in#IN	prep
in#IN	sit#VB	-prep
in#IN	car#NN	pobj
the#DT	car#NN	-det
car#NN	the#DT	det
car#NN	in#IN	-pobj

Table 4: Extracted context features for *Mary and John sit in the car*, using the Stanford parser, with collapsing and propagation (propagated dependencies in italics)

Term	Context Feature	
Lemma	Lemma	Dependency
Mary#NP	John#NP	conj_and
Mary#NP	sit#VB	-nsubj
John#NP	Mary#NP	-conj_and
John#NP	*sit#VB*	*-nsubj*
sit#VB	Mary#NP	nsubj
sit#VB	*John#NP*	*nsubj*
sit#VB	car#NN	prep_in
the#DT	car#NN	-det
car#NN	the#DT	det
car#NN	sit#VB	-prep_in

is an established way for relative comparison of DTs against lexical-semantic networks. Since the measure is heavily dependent on the structure and coverage of the semantic network, scores are not comparable across languages.

5 Experimental Settings

5.1 Context Feature Representations

To evaluate the performance of our custom collapsing rules, we create distributional thesauri (DT) using dependency parse relations as context features. We use the PCFG Stanford parser (Klein and Manning, 2003) to parse English. As the Stanford parser has options for collapsing and propagation (de Marneffe et al., 2006), we can directly compare DTs that are computed with different Stanford parser settings (collapsing, propagation) with the performance achieved when using our collapsing rules. For German, we use the Mate-tools parser (Bohnet, 2010; Seeker and Kuhn, 2012). As the Mate-tools parser does not feature collapsing, we cannot directly compare results. In fact, the lack of collapsing for German was the main motivation of our work. Instead, we can measure the quality improvements that collapsing introduces for similarity computation.

For higher precision, words are lemmatized leading to context features as shown in Tables 3 and 4. For the purpose of computing semantic similarity, we model dependencies in both directions by adding 'inverse' dependencies (e.g. `-nsubj`). As dependency relations are directed, not using inverse dependencies affects similarity computations of words commonly used as dependents in dependencies (e.g. *Mary* in Table 3).

Tables 3 and 4 demonstrate the impact of collapsing and propagation. Even though Table 3 contains more dependency relations, they are less discriminative than the collapsed dependencies in Table 4. Dependency propagation adds a dependency from *John* to *sit*, leading to a higher recall in the similarity computation.

5.1.1 Trigram Baseline

As a baseline system, we use a context representation of trigrams. This is an unsupervised, language-independent feature extractor that uses the left and right neighboring words as a combined context feature, e.g. extracting the term *likes* and the context feature *Mary_@_John* from the phrase *Mary likes John*. This feature representation is language-agnostic and thus it can be used for most languages with an established tokenization. For a fair comparison, we run this baseline in two configurations: first, without any linguistic processing and second, for an analysis of the impact of dependency features, we lemmatize all words.

5.1.2 Stanford Parser

stanford_basic Dependency parsing without collapsing or propagation

stanford_collapsed Dependency parsing with built-in collapsing

stanford_collapsed_prop Dependency parsing with built-in collapsing and propagation

stanford_basic_custom_collapsing Dependency parsing without built-in collapsing, applying our English collapsing rules afterwards

stanford_basic_custom_collapsing_prop
Dependency parsing without built-in collapsing, applying our English collapsing rules with propagation afterwards

5.1.3 Mate-tools Parser

matetools Dependency parsing without collapsing (no built-in collapsing available)

matetools_custom_collapsing Dependency parsing without collapsing, applying our German collapsing rules afterwards

matetools_custom_collapsing_prop Dependency parsing without collapsing, applying our German collapsing rules with propagation afterwards

5.2 Data

We chose datasets of different sizes for German and English. Both of these sets consist of randomly sampled sentences from news articles from the Leipzig Corpora Collection (Richter et al., 2006).

To assess the impact of training data size, we have taken samples of different sizes. These samples were taken from the full sets and include the following sizes: 0.1M, 1M and 10M sentences.

5.3 Similarity Computation

The similarity computation is performed using the JoBimText framework (Biemann and Riedl, 2013). It incorporates UIMA annotators for feature extraction, so we can add the collapsing annotator to the feature extraction pipeline on top of the parser output. We use the settings from Riedl and Biemann (2013). Terms and their context features are extracted from the input text. In our experiments, we use neighboring words and dependency parse features. We calculate the corpus frequencies for terms, context features and the term–context feature combinations. Using these frequencies, we compute the Lexicographers Mutual Information significance measure (Evert, 2005) between terms and contexts. We prune context features that occur with more than 1000 words and only keep the top 1000 most significant context features per word. Similarity between words is computed by counting the number of context features that two words share. The result is a distributional thesaurus where for each word we obtain up to 200 similar words, of which we evaluate on the top 5 only.

6 Results

Tables 5 and 6 show the evaluation results. In line with previous results, a larger corpus size results in more accurate similarities. Corpus size has a larger impact on rare nouns, where more input text is required to get sufficient 'signals' for similarity calculation. This becomes most apparent for German, where, due to the more complex morphology and noun compounding, we find about twice as many different word forms as in English (2.8M vs. 1.4M words) in the 10M corpora.

Overall, a larger corpus size leads to better DTs, which is expected. Compared to the baselines, dependency path features also improve the DTs. This is due to the structured, more accurate features and the fact that with dependency parsing, we obtain several context features for most words in a sentence (cf. Table 3).

Of the structural alternatives, collapsing brings a large boost and propagation usually adds a small improvement on top, especially for rare nouns. Therefore, we conclude that collapsing and propagation help improve the similarity computation for both languages. Some example DT entries can be seen in Table 7. Even though the DT entries do not change much, using collapsing and propagation puts the more similar terms on top (e.g. *office, bank* or *Bahnhof (railway station)*), while less similar terms like *student or Straße (street)* are ranked lower. Collapsing and propagation do not only lead to more accurate similarities, they also improve the recall. For rare German nouns, the distributional thesaurus contains similarities for only 592 out of 1000 test nouns, when computed using the Matetools parser alone. Collapsing improves the recall to 700 nouns and propagation offers an additional increase to 703 words.

The extrinsic comparison of our collapsing engine with the custom rules shows a comparable performance to Stanford parsing with collapsing. The scores show almost no difference for collapsing and up to 0.002 score points difference for propagation, indicating – as in Section 4.1 – that the dependencies produced by our rulesets are very similar to Stanford parser dependencies.

7 Conclusion

In this paper, we have presented a flexible framework for collapsing, which can be applied on top of arbitrary dependency parser outputs. We release, to our knowledge, the first dependency collapsing and

Table 5: Average WordNet path scores for the top 5 most similar words in a DT, considering different corpus sizes and word lists of frequent / rare English nouns

Method	Corpus Size (freq. nouns) 0.1M	1M	10M	Corpus Size (rare nouns) 0.1M	1M	10M
baseline	0.178	0.228	0.280	0.044	0.129	0.190
baseline_lemma	0.187	0.240	0.280	0.065	0.137	0.194
stanford_basic	0.210	0.272	0.302	0.096	0.183	0.229
stanford_collapsed	*0.225*	*0.292*	*0.322*	0.100	0.193	*0.241*
stanford_collapsed_prop	0.222	0.291	0.319	*0.106*	*0.200*	*0.241*
stanford_basic_custom_coll	0.224	0.291	0.321	0.101	0.193	*0.241*
stanford_basic_custom_coll_prop	0.224	0.290	0.319	0.104	0.195	*0.241*

Table 6: Average GermaNet path scores for the top 5 most similar words in a DT, considering different corpus sizes and word lists of frequent / rare German nouns

Method	Corpus Size (freq. nouns) 0.1M	1M	10M	Corpus Size (rare nouns) 0.1M	1M	10M
baseline	0.127	0.165	0.229	0.000	0.009	0.056
baseline_lemma	0.128	0.183	0.254	0.000	0.009	0.062
matetools	0.144	0.208	0.265	*0.001*	0.017	0.081
matetools_custom_coll	*0.149*	*0.217*	0.273	*0.001*	0.022	0.103
matetools_custom_coll_prop	0.147	*0.217*	*.0.274*	*0.001*	*0.023*	*0.104*

Table 7: Top 5 most similar words for English and German nouns, and their average Path scores

English: *branch*			German: *Bahnhofsplatz (station square)*		
stanf_basic	stanf_coll	stanf_coll_prop	matetools	matetools_coll	matetools_coll_prop
student	office	office	Innenstadt (city)	Bahnhof (station)	Bahnhof (station)
area	bank	bank	Bahnhof (station)	Straße (street)	Straße (street)
official	company	company	Hauptbahnhof (station)	Innenstadt (city)	Innenstadt (city)
bank	project	group	Straße (street)	Hauptbahnhof (station)	Hauptbahnhof (station)
business	director	director	Stadtteil (district)	Schulhof (school yard)	Platz (square)
0.152	0.153	0.170	0.152	0.169	0.219

propagation mechanism for German. This framework was used to implement collapsing and transitive propagation of dependencies over conjunctions for English and German. We have shown that collapsing improves the quality of distributional models. Also, our English ruleset offers comparable performance to the built-in Stanford collapsing rules. Since it is a separate component, it can be used to add collapsing functionality to any other English dependency parser. The parser framework is freely available for download and is accessible under a permissive license. We supply extensible rulesets for English based on the Stanford dependency tagset and for German based on the TIGER (Brants et al., 2002) tagset.

Since we have demonstrated the utility of dependency collapsing and propagation for semantic tasks such as distributional similarity, we believe that the German collapsing and propagation rules are a valuable addition for German NLP. For the future, we would hope that other groups might add collapsing rules for more languages.

Acknowledgment

This work has been supported by the German Federal Ministry of Education and Research (BMBF) within the context of the Software Campus project LiCoRes under grant No. 01IS12054.

References

Chris Biemann and Martin Riedl. 2013. Text: Now in 2D! a framework for lexical expansion with contextual similarity. *Language Modelling*, 1(1):55–95.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proc. Workshop on Current Trends in Biomedical NLP: Shared Task*, pages 10–18, Boulder, CO, USA.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proc. International Conference on Computational Linguistics (COLING 2010)*, pages 89–97, Beijing, China.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proc. Workshop on Treebanks and Linguistic Theories*, pages 24–41, Sofia, Bulgaria.

Marie-Catherine de Marneffe and Christopher D Manning. 2008. The Stanford typed dependencies representation. In *Proc. International Conference on Computational Linguistics (COLING '08)*, pages 1–8, Manchester, UK.

Marie-Catherine de Marneffe, Bill MacCartney, and Christoper D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. Language Resources and Evaluation (LREC 2006)*, pages 449–454, Genova, Italy.

Amit Dubey and Frank Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *Proc. Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 96–103, Sapporo, Japan.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proc. Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014*, pages 1–11, Dublin, Ireland.

Stefan Evert. 2005. *The statistics of word cooccurrences: word pairs and collocations*. Ph.D. thesis, IMS, Universität Stuttgart.

Christiane Fellbaum. 1998. *Wordnet. An Electronic Lexical Database*. MIT Press, Cambridge, MA.

David Ferrucci and Adam Lally. 2004. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering 2004*, 10(3-4):327–348.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet – a lexical-semantic net for German. In *Proc. ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications (ACL-EACL '97)*, pages 9–15, Madrid, Spain.

Verena Henrich and Erhard Hinrichs. 2010. GernEdiT – the GermaNet editing tool. In *Proc. Language Resources and Evaluation (LREC 2010)*, pages 2228–2235, Valletta, Malta.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Compututational Linguistics*, 24(4):613–632.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 shared task on event extraction. In *Proc. Workshop on Current Trends in Biomedical NLP: Shared Task*, pages 1–9, Boulder, CO, USA.

Dan Klein and Christoper D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 423–430, Sapporo, Japan.

Peter Kluegl, Martin Toepfer, Philip-Daniel Beck, Georg Fette, and Frank Puppe. 2014. UIMA Ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering*, pages 1–40.

Dekang Lin. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proc. Meeting of the Association for Computational Linguistics and Conference of the European Chapter of the ACL (ACL-EACL '97)*, pages 64–71, Madrid, Spain.

Philip Ogren and Steven Bethard. 2009. Building test suites for UIMA components. In *Proc. Workshop on Software Engineering, Testing, and Quality Assurance for NLP (SETQA-NLP 2009)*, pages 1–4, Boulder, CO, USA.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41, Boston, MA, USA.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. 2006. Exploiting the Leipzig Corpora Collection. In *Proc. IS-LTC 2006*, pages 68–73, Ljubljana, Slovenia.

Martin Riedl and Chris Biemann. 2013. Scaling to large^3 data: An efficient and effective method to calculate distributional thesauri. In *Proc. Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 884–890, Seattle, WA, USA.

Martin Riedl, Irina Alles, and Chris Biemann. 2014. Combining supervised and unsupervised parsing for distributional similarity. In *Proc. International Conference on Computational Linguistics (COLING 2014)*, pages 1435–1446, Dublin, Ireland.

Wolfgang Seeker and Jonas Kuhn. 2012. Making ellipses explicit in dependency conversion for a German treebank. In *Proc. Language Resources and Evaluation (LREC 2012)*, pages 3132–3139, Istanbul, Turkey.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proc. Empirical methods in Natural Language Processing (EMNLP 2003)*, pages 81–88, Sapporo, Japan.

Systematic Acquisition of Reading and Writing: An Exploration of Structure in Didactic Elementary Texts for German

Kay Berkling, Rémi Lavalley
Cooperative State University
Karlsruhe, Germany
remiL@singularity.fr
berkling@dhbw-karlsruhe.de

Uwe Reichel
Institute of Phonetics and Speech Processing
University of Munich, Germany
reichelu@phonetik.uni-muenchen.de

Abstract

The work presented here is part of a larger study that looks at the impact that structured teaching materials have on improving orthographic skills. The first step is an analysis of the current status of linguistic structures in common primary readers (primers) since their key purpose is to lead children along a systematic learning path towards becoming proficient readers and writers. Using text processing tools detecting morphemes, syllables and sentence structures, several popular primers are automatically examined with respect to their systematic approach to language at the phonics/syllable, word and sentence levels. It can be shown that there is little apparent structure in today's school texts at any of the examined levels when compared to some older schoolbooks (1877, 1904) that have a systematic and explicit progression at all examined levels.

1 Introduction

Over the last decade the number of children in Germany referred to language pathologists (Logopädie) has increased manifold according to WiDO AOK (health insurance) (2012; 2013) to reach an all-time high of 25% for boys aged around six. The cause for this phenomenon has not been well studied. Our hypothesis is that an improved structure in learning materials at the word and sentence level may improve orthographic skills. As a first step towards answering this question, this paper looks at the given structure in popular first grade readers (primers). To study the actual impact of structure on performance is beyond the scope of this paper but has been covered in part by other publications in this series of work (Berkling and Pflaumer, 2014; Berkling et al., 2015). This line of research is further motivated by results stemming from a large-scale analysis of orthographic skills in a corpus from first grade until eighth grade. In order to understand why certain spelling errors persist, more analysis is clearly needed to show the impact of teaching methods on acquisition of orthography.

Much work has been done in studying various approaches for English reading and writing acquisition. The National Reading Panel (Donald N. Langenberg et al, 2000) and more up-to-date studies by Galuschka (2014) have published meta-reviews of major comparative and quantitative studies, realizing that phonics[1] is a vital component in reading instruction. It has been shown in the Anglophone research by Stahl (1989) that relying solely on the whole-word approach is to the detriment of the weaker students. It has also been shown by Steffler (2001) that making structure explicit leads to efficient learning results. To our knowledge, no comparative work has been done on the German language. Phonics in its complexity is also not known as a method for reading/writing acquisition in Germany. In that sense, phonics is not apparent in the elementary readers while other methods prevail.

Currently, in Germany, there are mainly 2-3 popular methods in use. One such method is called "Lautiermethode" and refers to the theory that words can be sounded out one letter at a time (Reichen, 2008; Brügelmann, 2014). By synthesizing the sounds, the word is supposed to be read in its entirety. This works approximately for words like "Oma" or "Banane" that is an imported word but deemed "lautgetreu", meaning one grapheme corresponds to one phoneme. However, this approach fails to generalize to typical German structures in which a single grapheme <e> can stand

[1] The explicit and systematic (including clear sequence and scope) instruction of patterns in phoneme-grapheme correspondence)

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 67–76, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

for at least three semantically distinct pronunciations as a function of its position within syllable and word.

Another popular method says that reading acquisition takes place at the syllable level as this is more natural for children to bridge from the spoken language to the written representation (Röber-Siekmeyer, 2004). However, if the syllables are not taught well, then the learner pronounces the unstressed syllable in a stressed manner, thereby not recognizing the sound of the resulting word, for example by reading "Mutter" as /mʊt.tɐr/ instead of /mʊtɐ/.

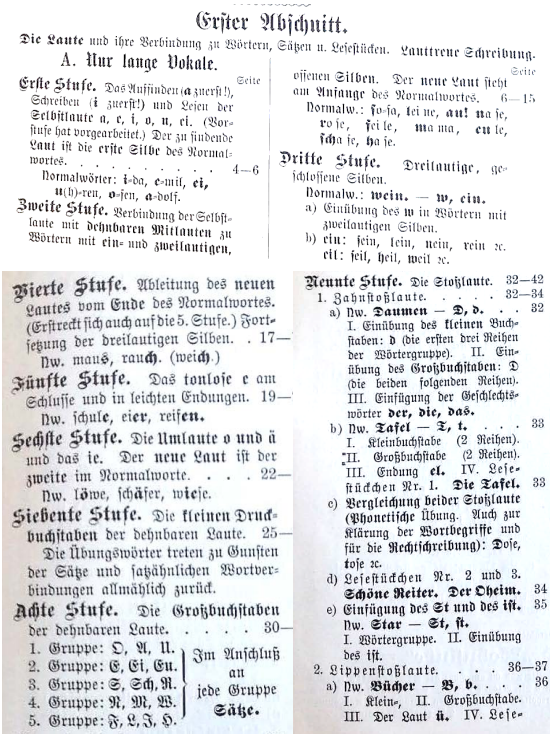
Finally, the whole word approach is widely used, even if not explicitly stated. It assumes that the child, given enough practice, will memorize the words as a unit. In general, experts say that first grade learning and teaching is based on a mix of all these methods and their respective effectiveness is child dependent.

Primers (*Fibel* in German) for first grade traditionally have the primary purpose of leading children in a systematic manner towards learning to read and write. As a first step towards understanding these methodologies and their progression, the goal of this work is to analyze and document selected structures and their progression in primary readers. The analysis looks at the lexical level through syllable and morpheme occurrence, word structures at the syllable level and complexity progression at the sentence level. Texts are analyzed for a number of books, including two older ones from 1877 and 1904 that show explicit structure unlike any of the ones used today.

The rest of the paper proceeds as follows. Section 2 motivates the proposed levels of analysis. After a brief overview of German phonics system and sentence complexity levels in Section 3, Section 4 will explain the tools used for analysis. Section 5 will introduce the data and Section 6 reports on the results, comparing the various different texts. Section 7 discusses the results.

2 Motivation

Elementary primers have traditionally had the purpose of serving as a guide for learners of a reading and writing system. In this function, they should naturally display a thought-out systematic approach on how they move from simple towards increasingly complex materials, considering the problem of inert knowledge (what children theo-


Erster Abſchnitt.

Die Laute und ihre Verbindung zu Wörtern, Sätzen u. Leſeſtücken. Lauttreue Schreibung.

A. Nur lange Vokale.

Erſte Stufe. Das Auffinden (a zuerſt!), Schreiben (i zuerſt!) und Leſen der Selbſtlaute a, e, i, o, u, ei. (Vorſtufe hat vorgearbeitet.) Der zu findende Laut iſt die erſte Silbe des Normalwortes. . . . . 4—6
Normalwörter: i-da, e-mil, ei, u(h)-ren, o-fen, a-dolf.

Zweite Stufe. Verbindung der Selbſtlaute mit dehnbaren Mitlauten zu Wörtern mit ein- und zweilautigen, offenen Silben. Der neue Laut ſteht am Anfange des Normalwortes. 6—15
Normalw.: ſo-ſa, lei ne, au! na ſe, ro ſe, fei le, ma ma, eu le, ſcha fe, ha ſe.

Dritte Stufe. Dreilautige, geſchloſſene Silben.
Normalw.: wein. — w, ein.
a) Einübung des w in Wörtern mit zweilautigen Silben.
b) ein: fein, lein, nein, rein ꝛc. eil: feil, heil, weil ꝛc.

Vierte Stufe. Ableitung des neuen Lautes vom Ende des Normalwortes. (Erſtreckt ſich auch auf die 5. Stufe.) Fortſetzung der dreilautigen Silben. . 17—
Nw. maus, rauch. (weich.)

Fünfte Stufe. Das tonloſe e am Schluſſe und in leichten Endungen. 19—
Nw. ſchule, eier, reiſen.

Sechſte Stufe. Die Umlaute ö und ä und das ie. Der neue Laut iſt der zweite im Normalworte. . . . 22—
Nw. löwe, ſchäfer, wieſe.

Siebente Stufe. Die kleinen Druckbuchſtaben der dehnbaren Laute. 25—
Die Übungswörter treten zu Gunſten der Sätze und ſatzähnlichen Wortverbindungen allmählich zurück.

Achte Stufe. Die Großbuchſtaben der dehnbaren Laute. . . . . 30—
1. Gruppe: O, A, U.
2. Gruppe: E, Ei, Eu.
3. Gruppe: S, Sch, R.
4. Gruppe: N, M, W.
5. Gruppe: F, L, J, H.
Im Anſchluß an jede Gruppe Sätze.

Neunte Stufe. Die Stoßlaute. 32—42
1. Zahnſtoßlaute. . . . . 32—34
a) Nw. Daumen — D, d. . . 32
I. Einübung des kleinen Buchſtaben: d (die erſten drei Reihen der Wörtergruppe). II. Einübung des Großbuchſtaben: D (die beiden folgenden Reihen). III. Einfügung der Geſchlechtswörter der, die, das.
b) Nw. Tafel — T, t. . . . 33
I. Kleinbuchſtabe (2 Reihen). II. Großbuchſtabe (2 Reihen). III. Endung el. IV. Leſeſtückchen Nr. 1. Die Tafel. 33
c) Vergleichung beider Stoßlaute (Phonetiſche Übung. Auch zur Klärung der Wortbegriffe und für die Rechtſchreibung): Doſe, koſe ꝛc.
d) Leſeſtückchen Nr. 2 und 3. Schöne Reiter. Der Oheim. 34
e) Einfügung des St und des ſt. 35
Nw. Star — St, ſt.
I. Wörtergruppe. II. Einübung des ſt.
2. Lippenſtoßlaute. . . . 36—37
a) Nw. Bücher — B, b. . . . 36
I. Klein-, II. Großbuchſtabe. III. Der Laut ü. IV. Leſe-

Figure 1: Index pages for a primer around 1900

retically know but are unable to use) which can be reduced by usage in different contexts (Bereiter and Scardamalia, 1985).

When looking at old primers, this type of progression is often made explicit within the index or names of the chapters as shown in Figure 1. Today's books do not explicitly have a progression apart from the order of introduction of the graphemes. The goal of the work presented here is to visualize and render explicit any inherent progression that can be found at the syllable, morpheme or sentence structure level, keeping in mind that a systematic approach, including explicitness is important according to a study by Steffler (2001) to ensure the learners' grasp of linguistic structures. Finally, a look at progression should include a look at the complexity that can be reached.

It is well known, that learning builds on previous knowledge and that each new material should have small consecutive steps building on each other (Martin and Rose, 2005; Leong, 1998). As a consequence, complexity grows incrementally and builds on the knowledge of the preschooler, which is mostly based on syllable structure of the language and progresses from there as described by Siekmeyer (2009).

For the German language, the most common syllabic pattern is the "Trochee" type of word

(2-syllable length, first syllable stressed, second one unstressed). A comparable example for both English and German is the word "**dou**ble" or "doppelt". The top 10,000 German words in newspapers include about 16% of pure 2-Syllable Trochees that follow this pattern. The 100 most used words cover around 45% in a standard text. The rest of the words are constructed words, such as compound words or those containing prefix or suffix attached to words that contain the Trochee pattern, where students have to generalize the pattern to unseen words – these make up about 45% of the top 10,000 words according to an automated analysis by Berkling (2014).

The remaining words, like "Auto", "Banane", "Tiger", "Portemonnaie" are imported vocabulary and follow different orthographic patterns that are not directly comparable to German. This is why the study of the Trochee is at the center of the word-pattern analysis.

Research questions to answer by analyzing the data are therefore the following: 1) Are the patterns of the German language (in form of Trochee) occurring in some structured form? 2) Is there repetition at syllable and morpheme level to support the cognitive process by achieving automation as described by McLaughlin (1990)? 3) Is the vocabulary embedded within sentence structures that exercise grammatical structures, including morpheme endings?

3 The Structure of German Text

In this section, the underlying theory concerning syllable types for the German language is explained. The structures underlying the study of word repetition and sentence complexity are also thereby defined and motivated.

3.1 Syllables

The German language distinguishes between three major classes of Trochee that will be described next.

- **Type 1: C-V-C-red ("b-e-t-en"):** This type of Trochee is the simplest form (phonemes: Consonant-Vowel-Consonant-reduction), comparable in that sense to the "cat", "hat", "mat" vocabulary used in the first steps of English phonics lessons. The 1-syllable form CVC derives directly from the 2-syllable Trochee ("gab" – as past tense of "geben"). In this case, the morpheme boundary is within the Trochee. CVCred can not be reduced to a 1-syllable words when there is no morpheme boundary within the Trochee, examples are "bird" ("Vogel") or "vase" ("Vase"). Identifiable features of this word type are the tense/long vowel (V) followed by a single consonant sound and a reduction syllable that contains the letter <e> pronounced as schwa /ə/.

- **Type 2: C-v-C1-red ("B-e-tt-en"):** The second type of word is one of the most difficult features of the written language to master by learners of the orthographic system (phonemes: Consonant - short vowel - Consonant - reduction). It distinguishes itself from the first type only by the feature tense/lax of the vowel, perceived as shortness in this form. As in the first type, there is only one consonant sound in the center of the word. However, in the orthography the tenseness of the preceding vowel is denoted by duplicating the center consonant letter. Orthographically, there are regularities such as <tt>, <nn> and irregularities such as <ng>, <tz>, <ck>, or <sch> and <ch> that need to be mastered. The 1-syllable form CvC1 derives directly from the 2-syllable Trochee ("Betten" (plural) – "Bett" (singular)). In this case, the morpheme boundary is within the Trochee. Not all CvC1red will reduce to 1-syllable words, such as the word for "rattle" ("Rassel").

  Identifiable features of this word type are the lax/short vowel (v) followed by a single consonant sound and a reduction syllable containing the letter <e> pronounced as schwa /ə/.

- **Type 3: C-v-C1-C2-red ("r-a-s-t-e-n"):** The third type of word is easier than the second one for the learner, as there are two distinct consonant sounds in the center of the word to help denote the tenseness of the preceding vowel. No orthographic particularities need to be mastered. The 1-syllable form CvC1C2 derives directly from the 2-syllable Trochee ("rasten" (verb) – "Rast" (noun)). In this case, the morpheme boundary is within the Trochee. Not all CvC1C2red will reduce to 1-syllable words, such as the word for *turn*: "Wende". Identifiable features of this word

type are the lax/short vowel (v) followed by two consonant sounds and a reduction syllable containing the letter <e> pronounced as schwa /ə/.

- **Other categories** of words include foreign words and high-frequency words as well as constructed words (compound words and those including prefix and suffix). These are not considered separately for the purpose of this study as they decompose either into the aforementioned formats or do not pertain to German phonics rules.

3.2 Sentences

There are numerous publications regarding the definition of sentence readability (Glöckner et al., 2006; Sitbon and Bellot, 2008; DuBay, 2008). Nelson (2012) is an example of a recent overview of such measures. Most of them have been designed for English. Regarding German language, little research has been done on that subject. We can cite (Hancke et al., 2012) where the authors used a number of different features to determine sentence readability: average number of words, characters, syllables, lexical features (noun and verb token ratios, textual lexical diversity, ...), syntactic features (number of noun or verb phrases, average length of a noun phrase, ...), language models (trained with texts for children vs. newspapers) and morphological features (ratio of finite verbs, compounds, ...).

However, the goal of the study presented here is not exactly to measure a sentence readability, but to describe the progression in the structures' complexity (word-level clues are considered separately, as described in Subsection 3.1). Hence, for the purpose of written language acquisition a progression as described by Clahsen (1982; 1988) for L1 acquisition in children is chosen as a first approach. (A more detailed analysis of sentence structures can be found in Berkling (2014)). For the purpose of this study, sentence structures are distinguished at the following levels:

- LEVEL 1: One-word utterances without counting articles, so it covers simple noun phrases such as "Peter" and "eine Katze" (a cat) and imperative verbs "Lauf!" ("Run!").
- LEVEL 2: Two-word utterances without counting articles, so simple noun phrases (as described for level 1) + Verb in present tense ("Peter isst" ("Peter is eating"). "Eine Katze läuft" ("A cat is running") or noun phrases including adjectives not followed by a verb: "die kleine Katze" ("the little cat").
- LEVEL 3: Common sentences, including Adjectives, Adverbs, ... So, noun phrases (including complex ones, such as article + adjective + noun) + Verb with or without Object: "Die kleine Katze." ("the little cat."). "Die kleine Katze läuft" ("The little cat is running"). "Ich gehe nach Hause" ("I am going home").
- LEVEL 4: Verb positions: Conjugated verbs are located in the second position of German sentences. In some circumstances, it increases difficulty in the process of understanding the sentence meaning. The following cases are thus considered as Level 4:
  - past and future tenses, built with a conjugated auxiliary (2^{nd} position) and an infinitive (future) or past participle (past tense) form of the meaningful verb, located at the end of the sentence: "Wir *werden* nächsten Sommer nach Spanien fahren". ("We *will* go to Spain next Summer"). "Du *hast* spät in der Nacht gearbeitet". ("You *have* worked late in the night").
  - modal verbs: used to express obligation, ability or will. These verbs are conjugated and located in the second position in the sentence, then the verb on which the modality applies is located at the end of the sentence: "Du *sollst* morgen nicht kommen" ("You *should* not come tomorrow").
  - compound verbs: conjugated verb is in 2nd position, the particle is at the end of the sentence: "Die Sonne geht immer früher auf" ("The sun rises earlier and earlier") – The compound verb "**auf**gehen" (to rise) has a different meaning than the verb "gehen" (to go).
- LEVEL 5: complex sentences. Depending on the kind of clause, verb positions follow different rules. Thus, we have distinguished two subcategories, to reflect different complexities: the coordinate considered as an

independent sentence, where the verb is located in 2nd position (in both of the clauses). e.g., "Ich bin fertig *und* ich **gehe** jetzt in die Schule" ("I am ready *and* I **go** to the school now"), or in subordinate clauses with the verb located at the end of the subordinate clause: "Ich denke, *dass* er intelligent **ist**" ("I think *that* he **is** smart").

Sentence containing several levels are classified as the highest: "Ich werde ein Buch kaufen, **weil** ich viel *lese*." ("I will buy a book, **because** I *read* a lot.") – is thus considered as Level 5.

4 System

Three software modules as shown in Figure 2 are built to analyze the primary reader texts. A synthesizer, that performs automatic syllabification and morpheme boundary segmentation called Balloon, a word classification system (Berkling and Reichel, 2014) that builds on Balloon output, and a syntactic parser (Petrov et al., 2006) followed by a sentence classifier.

4.1 Balloon and Lexical Analysis

The word counter uses the output of Balloon, which works roughly as follows, for details see (Reichel, 2012).

The Grapheme-phoneme (G2P) conversion is carried out by a C4.5 decision tree (Quinlan, 2003). Syllable boundaries are placed in front of each sonority minimum, and their locations are subsequently adjusted in case German syllable phonotactics is violated. Word stress is again assigned by a C4.5 tree that predicts for each syllable whether it is stressed or not. Part of speech (POS) labels were assigned by a Markov tagger that additionally makes use of information stored in word suffix strings. Relevant suffix strings are extracted by means of an adaptation of the peak and plateau algorithm of Nascimento (1998). As POS inventory the Stuttgart Tübingen tag set (Schiller et al., 1995) is used.

The morphological analysis yields a flat segmentation of a word into morphemes and their morpheme classes. Each word is therefore decomposed into phonemes, syllables, stress markers and morpheme boundaries that are then used by the morpheme and syllable counter. The word counter simply counts new and previously seen syllables and morphemes for each page and then plots new vs. previously seen numbers for each

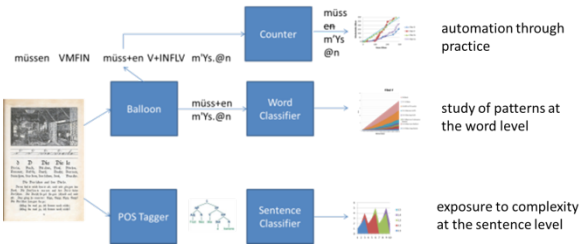


Figure 2: System Modules for calculating Features

page. The word classifier uses syllable and morpheme boundaries to filter out the 2-syllable words that match the Trochee types and their 1-syllable derivatives as described in Section 2 by their unique identifying features in a rule based system.

4.2 Sentence Classifier

The sentence classifier takes its input from the Berkley's parser for German, with *-tokenize* (to use the integrated tokenizer) and *-accurate* (favours accuracy over speed) options. A small set of simple rules - including ones designed to overcome some mistakes made by the parser - is used to assign levels according to Section 3.3 in decreasing order:

- looking for subordinate or coordinate clauses; this information is generally provided by the Berkley's parser. However, to increase recall (some of these clauses where not tagged as such by the parser), it has here been re-inforced by an upper layer using a list of words introducing subordinate clauses (dass, ob, ...) or coordinate clauses (und, oder, ...) and looking for verbs positions around these specific words in order to make the disctinction between "Ich habe einen Bruder und eine Schwester." ("I have a brother and a sister.") and "Ich spiele und ich **arbeite**." ("I play and I **work**.").

- looking for auxiliary in 2nd position and past participle or infinite verb at the end of the sentence for level 4. Or finite verb in second position and particle at the end of the sentence.

- number of words and their part of speech for levels 1 and 2.

5 Data

Various primary texts were transferred into electronic format up to page 50. The primers were chosen for their popularity and opposing methodologies ranging from no apparent methodology towards syllable, grapheme, and whole word approach. The two old primers from 1877 and 1904 have been analyzed only in part. Their makeup is different from today's primers. Separate sections are devoted to grapheme introduction, word-level training and sentence level training within progressively complex texts (as measured by sentence and word structure complexity). These sections have been extracted to demonstrate progression. Table 1 lists the available data.

Readers	Primary Methodology	Number of Words	Number of pages
Primer A	Whole word	1259	First
Primer B	Syllable strict	593	50
Primer C	Phonics 1877	580	pages
Primer D	Syllable	1010	(except
Primer E	Grapheme	1265	old
Primer F	Grapheme	932	primers)
Primer G	None	1328	
Primer H	Phonics 1904	320	

Table 1: Primary Readers Used in Study.

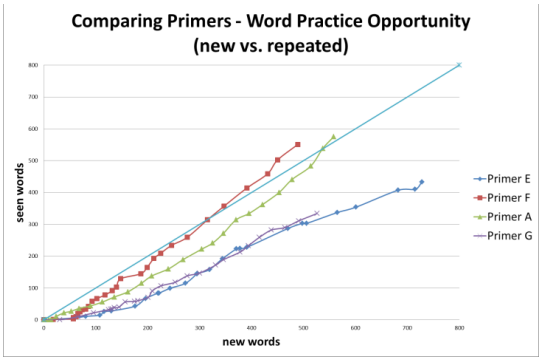


Figure 3: Repetition of words for practice, comparing four primers.

6 Evaluation

Due to limited space, only the results for selected readers are displayed where they show prominent differences. Results are reported for morpheme and syllable repetition, progression of Trochee types, and sentence complexities.

6.1 Ability to Practice on Words and Syllables

Figures 3 and 4 show how many words or syllables are repeated or new for each page. There is quite a large difference in training when comparing Primers F & A with E & G, where there is substantially less repetition.

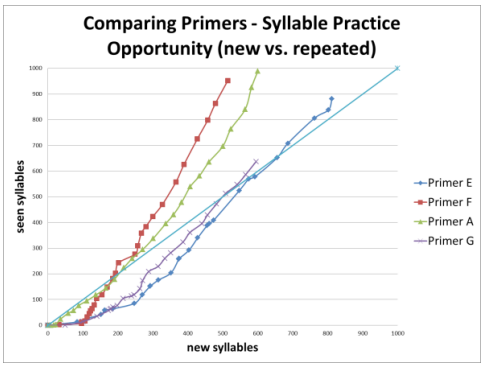


Figure 4: Repetition of syllables for practice, comparing four primers.

6.2 Progression towards Complexity at word level

The three diagrams in Figures 5 and 6 (See Appendix) show three different versions of progression for the different Trochee types and their 1-syllable derivatives. The figures are generated by looking at the distribution across the word types for every 5 consecutive words. Each slice on the x-axis therefore represents five words. The y-axis counts the number of word types for that slice from 1-5.

6.3 Progression at the Sentence Level

As depicted in the Appendix, Figures 7 to 9 show the progression in sentence complexity throughout the primers for three different books. Figure 7 depicts Primer F and some progression in the sentence structure can be seen throughout the reader. At the beginning of the book, there are lots of simple utterances (mainly Level 1, so Noun or Article+Noun), no Level 5 sentences (subordinate and coordinate clauses) and a few Level 3 sentences. As we proceed, we can observe more and more Level 3 sentences and the appearance of Level 4 sentences (verb tricks), while the number of Level 1 sentences decreases significantly. At the end of the book, there are lots of Level 4 and 5 sentences and almost no simple ones.

In contrast, Figure 8 depicts Primer E with no progression at all in the sentences structure. All the levels are merged all along the book, meaning children learn a few words and directly use them in complex sentences. Level 3 is the highest level of complexity reached, significantly diminishing the syntactic exercices that more complex sentences offer.

Figure 9 shows Primer from 1904 depicting the progression taken from the special section on practicing sentence complexity. This progression is visible in the diagram. At the beginning of the analyzed part of the book, it can be observed that most of the sentences belong to Level 3. Then their number is slightly decreasing as we browse to the end of the book. Meanwhile there are an increasing number of Level 4 and 5 sentences. The number of high complexity sentences is significantly larger here than observed in contemporary primers.

7 Conclusion

Looking at primers through a quantitative lens revealed large differences in primary reading material for first graders who are introduced to reading with these materials. It is important to be aware of such differences that may not be apparent immediately. Today, these effects are not studied in quantitative, systematic manner and teachers are not aware of the detailed particularities of their materials. Obviously, these primers are only one component of many materials chosen by teachers to work with the children in training their reading and writing skills. This type of analysis covers only one aspect of the classroom dynamics. However, it would be desirable to see a clear progression in the examined material and an explicit goal of the skills that the students in first grade should be able to reach along with a defined progression in that direction. Based on our work, there is no evident answer as to which methodology has a clearer progression at different levels. None of the contemporary primers show the distinctive marks of the two chosen older versions that contain an apparent progression at the word and the sentence level that takes the elementary student from the simple to the complex. Looking at the older primers, much more complexity was demanded from first graders at the end of the school year. It may be that in order to reach that level more practice opportunities were supplied and explicit progression at various levels was needed.

Future work will have to study whether there is a cause and effect here. The old approach provides a stark contrast to our primers in use today and today's primers differ significantly even among each other regarding practice and progression. None of the modern readers seem to spend the time (when compared to the older primers) on the typical German word structure of the Trochee. This work has shown that there are differences between readers, some of which are not well studied, and that there is a need to study the consequences this material has on student's learning and achievements.

References

Carl Bereiter and Marlene Scardamalia. 1985. Cognitive coping strategies and the problem of "inert" knowledge. *Thinking and learning skills: Current research and open questions*, 2:65–80.

Kay Berkling and Nadine Pflaumer. 2014. Phontasia - a Phonics Trainer for German Spelling in Primary Education. In *Workshop on Child Computer and Interaction WOCCI*, pages 33–38, Singapore.

Kay Berkling and Uwe Reichel. 2014. Der phonologische Zugang zur Schrift im Deutschen. In *Symposium Deutsch Didaktik, Sektion 7, Orthographie.*, Basel, CH.

Kay Berkling, Nadine Pflaumer, and Rémi Lavalley. 2015. German phonics game using speech synthesis: A longitudinal study about the effect on orthography skills. In *Workshop on Speech and Language Technology in Education SLaTE*, pages 167–172, Leipzig, Germany.

Hans Brügelmann. 2014. *Kinder auf dem Weg zur Schrift: Eine Fibel für Lehrer und Laien*. Libelle : Wissenschaft. Libelle-Verl., Bottighofen, 9th edition.

Harald Clahsen. 1982. *Spracherwerb in der Kindheit: Eine Untersuchung zur Entwicklung der Syntax bei Kleinkindern*, volume 4 of *Tübinger Beiträge zur Linguistik. Series A, Language development*. G. Narr, Tübingen.

Harald Clahsen. 1988. Parameterized grammatical theory and language acquisition. *Linguistic theory in second language acquisition*, pages 47–75.

Donald N. Langenberg et al. 2000. National Institute of Child Health and Human Development. Report of the National Reading Panel. Teaching children to read: An evidence-based assessment of the scientific research literature on reading and its implications for reading instruction.

William H. DuBay. 2008. The principles of readability. 2004. *Costa Mesa: Impact Information*, 76.

Katharina Galuschka, Elena Ise, Kathrin Krick, and Gerd Schulte-Körne. 2014. Effectiveness of treatment approaches for children and adolescents with reading disabilities: A meta-analysis of randomized controlled trials. *PloS one*, 9:2.

Ingo Glöckner, Sven Hartrumpf, Hermann Helbig, Johannes Leveling, and Rainer Osswald. 2006. An architecture for rating and controlling text readability. In *KONVENS*, pages 32–35, Konstanz, Germany.

Christine Göpner-Reinecke. 2013. Immer mehr Kinder brauchen vor dem Schulstart Hilfe beim Sprechenlernen. Accessed: September 14, 2015, `http://www.wido.de/meldung_archiv+M50c012ae424.html`.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability Classification for German using Lexical, Syntactic, and Morphological Features. In *COLING*, pages 1063–1080, Mumbai, India.

Rémi Lavalley and Kay M. Berkling. 2014. Data exploration of sentence structures and embellishments in german texts: Comparing children's writing vs literature. In *KONVENS*, pages 241–247, Hildesheim, Germany.

Deborah J. Leong. 1998. Scaffolding emergent writing in the zone of proximal development. *Literacy*, 3.2:1.

James R Martin and David Rose. 2005. Designing literacy pedagogy: scaffolding democracy in the classroom. *Continuing Discourse on Language*, pages 251–280.

Barry McLaughlin. 1990. Restructuring. *Applied linguistics*, 11.2:113–128.

Mario A. Nascimento and Adriano C.R. da Cunha. 1998. An experiment stemming non-traditional text. In *Proc. SPIRE'98*, pages 74–80, Santa Cruz de La Sierra, Bolivia.

Jessica Nelson, Charles Perfetti, David Liben, and Meredith Liben. 2012. Measures of text difficulty: Testing their predictive value for grade levels and student performance. *Council of Chief State School Officers, Washington, DC*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440.

John Ross Quinlan. 2003. *C4.5: programs for machine learning*. The Morgan Kaufmann series in machine learning. Morgan Kaufmann, Amsterdam [u.a.], 5th edition.

Uwe Reichel. 2012. PermA and Balloon: Tools for string alignment and text processing. In *Proc. Interspeech*, pages 1874–1877, Portland and Oregon.

Jürgen Reichen. 2008. *Hannah hat Kino im Kopf: Die Reichen-Methode Lesen durch Schreiben und ihre Hintergründe für LehrerInnen, Studierende und Eltern*. Heinevetter, Hamburg, 5th edition.

Christa Röber-Siekmeyer. 2004. Die Berücksichtigung des kindlichen Sprachwissens für den Schrifterwerb. In *Geschriebene Sprache. Strukturen, Erwerb, didaktische Modellbildungen. Schriftenreihe der PH Heidelberg Beltz Verlag Wissenschaft*, pages 129–144.

Christa Röber-Siekmeyer. 2009. *Die Leistungen der Kinder beim Lesen- und Schreibenlernen: Grundlagen der silbenanalytischen Methode ; ein Arbeitsbuch mit Übungsaufgaben*. Schneider-Verl. Hohengehren, Baltmannsweiler.

A. Schiller, S. Teufel, and C. Thielen. 1995. Guidelines fuer das Tagging deutscher Textcorpora mit STTS, url = `http://www.sfs.uni-tuebingen.de/resources/stts-1999.pdf`, accessed: sept. 15, 2015, address = Stuttgart.

Laurianne Sitbon and Patrice Bellot. 2008. A readability measure for an information retrieval process adapted to dyslexics. In *Second international workshop on Adaptive Information Retrieval (AIR 2008 in conjunction with IIiX 2008)*, pages 52–57.

Steven A. Stahl and Patricia D. Miller. 1989. Whole language and language experience approaches for beginning reading: A quantitative research synthesis. *Review of Educational Research*, 59.1:87–116.

Dorothy J. Steffler. 2001. Implicit cognition and spelling development. *Developmental Review*, 21.2:168–204.

Andrea Waltersbacher. 2012. Jeder vierte Junge zur Einschulung in Sprachtherapie. Accessed: September 14, 2015, `http://www.wido.de/meldung_archiv+m50f798a1a5a.html`.

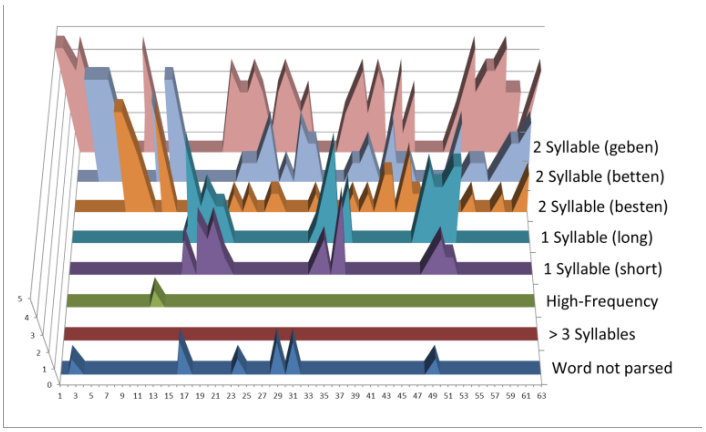


Figure 5: An excerpt from Primer H from 1904, showing progression from 2-syllable types "geben" to "betten" to "besten" to 1-syllable types and then proceeding to mix all variants in training.(x-axis = 5 word slice, y-axis = number of words of that type).

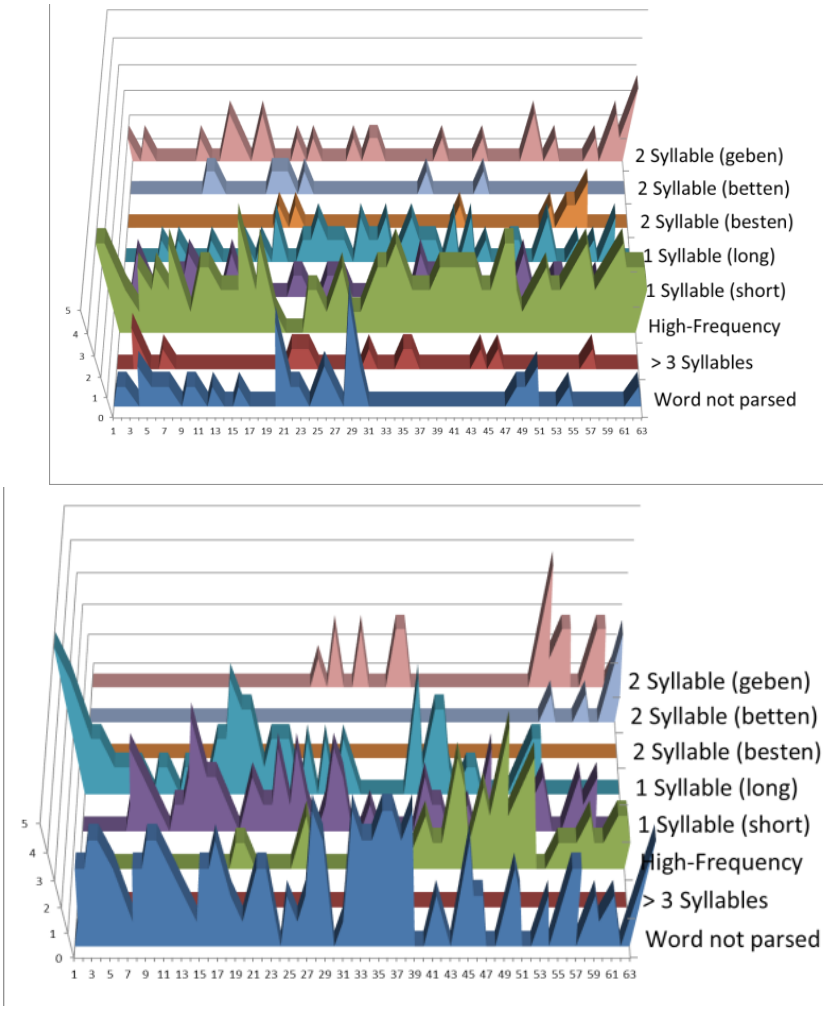


Figure 6: Primers E (at the top) and D show two different ways of using words. While E spends more time on High-Frequency words, Primer D spends time on syllables that are not real words. (x-axis = 5 word slice, y-axis = number of words of that type).

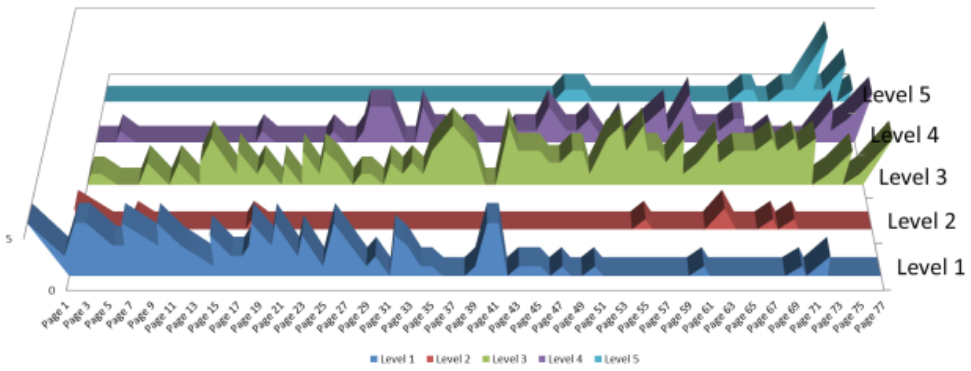


Figure 7: Progression of sentence level complexities for Primer F. Some progression to Level 5 is visible with little practice at level 5.

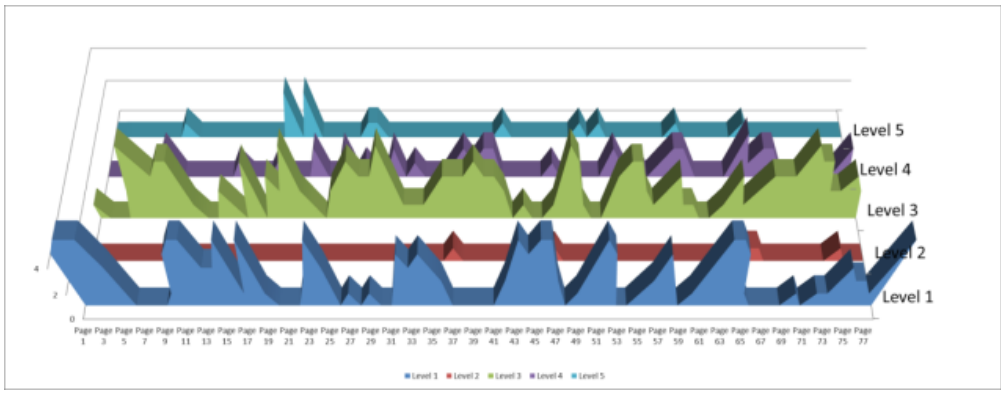


Figure 8: Progression of sentence level complexities for Primer E. Level 3 and 1 are exercised but there is no visible progression.

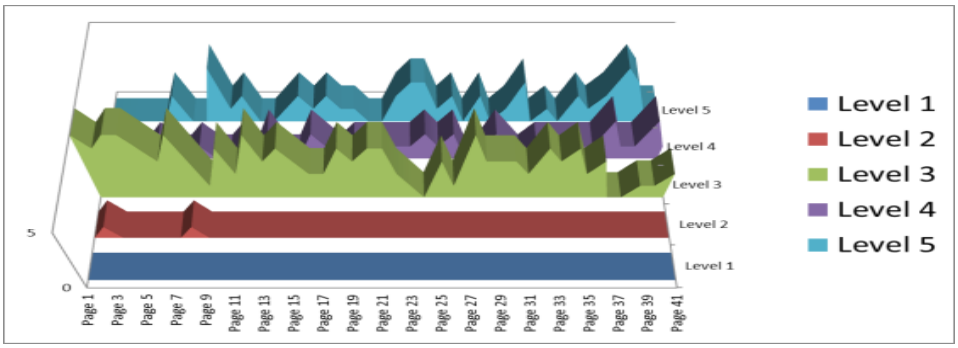


Figure 9: Progression of sentence level complexities for Primer H (1904) from Level 3 to Level 5 with increasingly significant practice at Level 5.

Wie oft schreibt man das zusammen? The Puzzle of Why some Separable Verbs in German are More Separable than Others

Nana Khvtisavrishvili **Stefan Bott** **Sabine Schulte im Walde**
Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart
{khvtisna,bottsn,schulte}@ims.uni-stuttgart.de

Abstract

In this work we address the question why different German particle verbs tend to occur with different frequency proportions in syntactically separated vs. non-separated forms. The problem has been studied from a theoretical point of view and the syntactic conditions that determine particle verb realization in separated and non-separated paradigms are quite clear. But, to the best of our knowledge, the question of why there is a variation among particle verbs with respect to how often they appear in different paradigms has never been addressed empirically so far. In this paper we present a corpus-based study which tackles this question. We formulate various morphological, semantic and pragmatic hypotheses which might explain the variation and we test them with clustering and linear regression techniques.

1 Introduction

German particle verbs (PVs) may occur in different syntactic paradigms, depending on the type of clause and the finite/infinite status of the base verb (BV). One of their best known characteristics is that of syntactic separability. PVs may be written together as one word or appear syntactically separated, as illustrated by (1) and (2). Finite PVs occur obligatorily separated in verb final clauses and syntactically non-separated in verb first and verb second clauses. This is the reason why they are also often called separable verbs.

(1) Die Praxis *sieht* meist noch ganz
The practice *looks* mostly still entirely
anders *aus*.
different *PRT*.
"The practice usually looks entirely different"

(2) Es sind keine Softkorallen, obwohl sie
They are no soft corrals, although they
so *aus|sehen*.
so *PRT|look*.
"They are no soft corrals, although they look like them."

The case of syntactically separated PVs is a quite cumbersome issue for NLP applications, especially in parsing and machine translation. The linear distance between verb and particle can be quite large, which makes it difficult to detect the syntactic dependency between them. Additionally, many verb particles, especially the most frequent, are homophonous to prepositions and other function words.

Even if PVs occur syntactically non-separated, this case is not homogeneous because PVs can also be separated morphologically by a functional morpheme, such as *-ge-* or *-zu-*. Non-separated uses of particle verbs can correspond to one of the following cases, as illustrated in Figure 1.

- Finite verbs in subordinate clauses (FIN): e.g. *. . . dass er sie an|lächelt.* (*. . . that he smiles at her.*)
- Infinitive, e.g. in combination with a modal verb (INF): e.g. *Ich kann da nur an|schließen.* (*I simply have to subscribe to that.*)
- Participle perfect (PP): e.g. *Er hat sie ein|ge|laden.* (*He has invited her.*)
- Infinitive with "zu" (IZU): e.g. *Die Sitzung war auf|zu|zeichnen.* (*The session had to be recorded.*)

Non-separated instances of PVs can occur in subordinate clauses (FIN) or appear in certain grammatical constructions which involve auxiliary verbs (PP, IZU & INF). The syntactically sep-

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 77–86, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

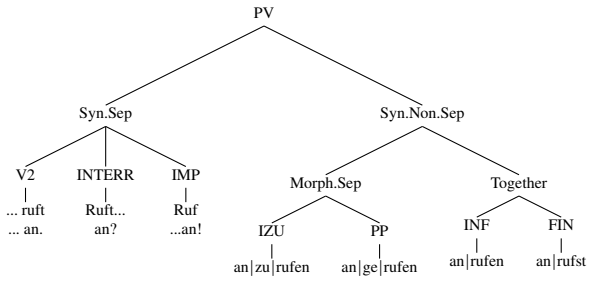


Figure 1: Different syntactic paradigms of the use of particle verbs

arated paradigm (SEP)[1] is much easier to define as a coherent class, since it consists of an inflected main verb and a clause final verb particle, as exemplified in (1). As Figure 1 shows, PVs may occur in indicative (V2), interrogative and imperative root clauses. However, in this work we do not distinguish between interrogative and indicative and we do not consider the imperative because of its low corpus frequency. From this discussion it should be clear that in German the realization of the PV as either separated or non-separated is fully determined by the clause type and the finite/infinitive distinction.

The syntactic and morphological aspects of the separated/non-separated dichotomy have been described adequately in traditional grammars and research literature (Lüdeling, 2001; Jacobs, 2005; Fuhrhop, 2007), but there is one aspect which has never been investigated, namely the proportions or relative frequencies with which different PVs occur in the different syntactic paradigms. To illustrate this, consider the verb *an|sehen* (*to watch/to resemble*) in (3) in contrast to *aus|sehen* (*to appear/to look like*) in (1)/(2).

(3) . . . ein Millionenpublikum das sich
. . . a million audience that REFL
Schrott an|sieht.
trash PRT|looks.
". . . an audience of millions that watches rubbish."

Neither *an|sehen* nor *aus|sehen* appear to be marked for a certain genre or register, they are both ambiguous and they have a similar corpus frequency (114 and 126 per million tokens, respectively). Nevertheless, they behave quite differently with respect to the proportions in which they occur in the syntactically separated paradigm: 20.5% vs 64.7%. This is surprising and, based on the relevant literature, we could not find an indication of why we observe such differences among PVs. Figure 2 shows the distribution of the proportion of separated occurrences over PVs as observed in the SdeWaC-Corpus (Faaß and Eckart, 2013). The x-axis represents relative frequency bands of syntactically separated occurrences of different PVs, the y-axis represents the count of PVs which falls into each relative frequency band. The PV *an|sehen* from example (3) would fall into the relative frequency band 20%-25%, which is the most densely populated one. The black curve represents the approximate density function, a smoothed representation of the histogram. It can be clearly seen that there is quite an amount of variation for which there is no straightforward explanation. Most notably, there is a long tail to the right, which means that a small number of PVs have a high tendency to occur syntactically separated.

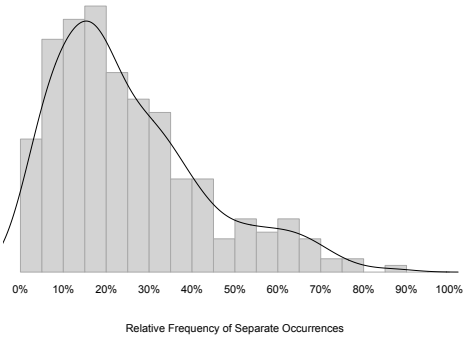


Figure 2: Histogram (bandwidth = 0.5) and density for the distribution of PVs according to their proportion of syntactically separated occurrence

In this paper we attempt to find reasons behind the variation we just described. We formulate various hypotheses on different syntactic, semantic and pragmatic factors which might influence the proportional distribution of PVs over the different syntactic paradigms. The current work mainly has a theoretic interest. It discovers a topic which has, so far, not received attention. It also represents a first attempt to solve the puzzle. Since PVs are very frequent in German, the question may have important implications for practical NLP applications. Parser performance is often poor in cases of separated PVs, in part because of the long lin-

[1] We use the term *paradigm* in a wide sense since the different PV realizations listed in Figure 1 are mutually exclusive. We do not intent to make a statement, however, on the exact theoretical status of this relation.

ear distance between verb and particle. Shedding some light on the issue of PV separability might improve parsing and all subsequent NLP tasks that depend on a reliable detection of the syntactic dependencies between the BV and the verb particles of PVs. Challenges that arise in the process of handling PVs for different NLP tasks show, in turn, the importance of investigating PVs to understand the problems associated with them.

2 Related Work

A number of studies have already investigated the topic of German PVs from both a theoretical as well as a corpus-based perspective. German PVs were extensively studied from the theoretical perspective in works of Lüdeling (2001) and Stiebels (1996); some other works have focused on a single particle such as Springorum (2009), dealing with the semantic of PVs with *an*; Lechler and Roßdeutscher (2009) studied PVs with the particle *auf*; Kliche (2009) looked at PVs with the particle *ab*.

The theoretical studies of PV separability have so far mostly dealt with German PVs with respect to their idiosyncratic behavior. Lüdeling (2001) investigated whether PVs are morphological objects or phrasal constructions and how they can be distinguished from secondary predicate constructions or adverbial constructions. She revealed a series of theoretical problems and analyzed PVs as lexicalized phrasal constructions, considering separability the strongest argument for this analysis. Müller (2001; 2003), in turn, argued for a syntactic analysis of PVs.

Jacobs (2005) studied PVs as one of several cases that pose problems for the determination of word boundaries. This affects the question of separability and orthographic separation. Also Fuhrhop (2007) was concerned with the morphological and orthographic aspect of the separability of German PVs. In contrast to Lüdeling's analysis of PVs as lexicalized phrasal constructions, Fuhrhop analyzed them as graphemic words.

Corpus-based, empirical investigations of PVs have received less attention. Schulte im Walde (2004) used statistical grammars to identify German PVs and provided quantitative description and a preliminary distributional analysis of German PVs. Schulte im Walde (2005) addressed the issue of feature selection to identify semantically nearest neighbors.

Some other works aimed at determining the degree of semantic compositionality of PVs. Bott and Schulte im Walde (2014) predicted the degree of PV compositionality relying solely on word window information. In their approach only lexical distributional distance between a PV and its corresponding BV was considered to be a predictor for compositionality. They were the first to automatically correct PV lemmas which occur in the syntactically separated paradigm, where they are consistently listed as the lemma of the base verb. They reported on problems with automatically parsed data in this respect.

As for the statistical study of variation of particle placement, Gries (2001; 2002; 2011) analyzed the variation of particle placement in English. Since in English the placement of verb particles is subject to relatively free variation (*John picked up the book* vs *John picked the book up*) and in German the realization of PVs as separated or unseparated is tied to the clause type, Gries' work cannot be directly replicated for German data.

To our knowledge no work comparable to what we propose here has been performed so far. In this study, we want to explore the behavior of German PVs with respect to the relative frequency distribution over different syntactic paradigms. In other words, we want to assess the empirical distribution of proportions corresponding to these paradigms and, by doing so, learn something about the nature of PVs.

3 Experiments and Data Analysis

Since we found that it is hard to understand why different PVs tend to occur in different syntactic paradigms in different proportions, our goal was to find factors which might explain the variation we could observe. For our experiments we used clustering techniques and simple correlation analysis based on least squared error regression. Clustering produces a partitioning of the data into classes which are derived in an unsupervised manner. This has two advantages; the first is that the classification is overt and the derived clusters can be inspected directly. The second advantage is that, by virtue of being an unsupervised technique, clustering classifies data points without the need of a previously given classification scheme. The clusters derived in one clustering procedure can be matched against various gold standards.

3.1 Hypotheses

We started out from the basic hypothesis H_b that the variation of the proportion with which different PVs can be observed in different paradigms is not a random factor but must be governed by some underlying reason. We therefor formulated a series of hypotheses which are elaborations of H_b. When formulating our hypotheses we considered two factors: 1) It must be possible to evaluate each hypothesis in an empirical way and 2) it should be interpretable in grammatical terms.

If we turn again to *aus|sehen* and *an|sehen*, the pair of verbs in examples (1) - (3), we already saw that these verbs share a number of common features, even if they occur in the syntactically separated paradigm in different proportions: they correspond to the same base verb, they have a similar corpus frequency and they are both ambiguous. The most evident difference they have is that they appear with different verb particles. It might also be argued that *aus|sehen* is ambiguous to a higher degree than *an|sehen*. Of course different verbs may also differ in the register and genre in which they tend to be used. Since the use of the syntactically separated paradigm is mainly tied to main clauses and different genres/registers may use more or less subordinate clauses, there may also be an indirect relation between genre/register and the tendency of verbs to occur in the non-separated paradigm. Genre and register are, however, often difficult to assess in corpora since genre-specific corpora tend to be much smaller than mixed-genre corpora. Nevertheless we can assume that average sentence length is a rough indicator for such difference.

Based on these considerations we formulated the following hypotheses:

- H_b: The variation of PVs with respect to the proportions that correspond to different syntactic paradigms is not a random factor. It is governed by some other underlying phenomenon.
- H1: Particles: Different particles influence the use of a corresponding PV in different syntactic paradigms.
- H2: Corpus Frequencies: The total corpus frequency has an impact on the proportional distribution over different paradigms.
- H3: Ambiguity: The degree of ambiguity of individual PVs can explain the behavior of PVs with respect to its proportional distribution over different paradigms.
- H4: Sentence Length: We take sentence length as a rough indicator for differences in text genre and register and hypothesize that there are correlations between average sentence length per PV and the proportion with which the PV occurs in different paradigms.

3.2 Data

For the extraction of features we used the SdeWaC corpus (Faaß and Eckart, 2013), a cleaned version of the deWaC corpus, which was compiled by the WaCky initiative (Baroni et al., 2009). SdeWaC consists of sentences from German web pages which contains syntactically well-formed and parseable sentences (880 million tokens). The corpus was tokenized with Schmid's tokenizer (Schmid, 2000), POS-tagged and lemmatized with Schmid's tree-tagger (Schmid, 1994) and parsed with Bohnet's MATE dependency parser (Bohnet, 2010). The parses provide information about dependency relations between components of a sentence. Dependency information is important in cases in which a PV occurs in a separated form, because it shows dependency between a particle and a corresponding base verb. The format of the corpus further provides lemma and part-of-speech annotation. For annotation the STTS tagset (Thielen et al., 1999) was used.

For the selection of the PVs in our data set, three additional corpora were used: HGC (Fitschen, 2004), DECOW12 (Schäfer and Bildhauer, 2012) and the German Wikipedia.[2]

3.3 Selection of Particle Verbs and building of the Data Set

For our experiments we created a data set of PVs which was balanced over the corpus frequencies of PVs and the particles to which they correspond. For this, we selected PVs randomly from three frequency bands – high, mid and low frequency – to investigate the behavior of particle verbs from different frequency bands. Occurrence frequencies are calculated as the harmonic mean of four different frequencies gained from the following corpora: SdeWaC, HGC, DECOW12 and Wikipedia.

Frequency bands are determined for each particle separately, i.e. thresholds for determining fre-

[2] Wikipedia dump (dewiki-20110410). `http://en.wikipedia.org/wiki/Wikipedia:Database_download`

PV	Sep.	PP	IZU	INF	FIN	Non-sep.
aussehen	0.5801	0.0207	0.0123	0.1886	0.1982	0.4198
anblicken	0.7994	0.0252	0	0.0466	0.1288	0.2005
ansehen	0.2025	0.3389	0.1907	0.1659	0.1019	0.7975
zuhören	0.3946	0.0569	0.0019	0.3136	0.2329	0.6054

Table 1: Feature Vectors

quency areas for different particles are different. The thresholds for the frequency bands were calculated by dividing the PVs with the same particle into equally large sets according to their overall corpus frequency (tertiles). In our work we investigateed PVs with the following 11 prepositional particles: *an*, *auf*, *ein*, *aus*, *zu*, *um*, *ab*, *unter*, *durch*, *über* and *nach*. We randomly selected 30 PVs for each particle from three frequency areas. The resulting list contained 938 PVs. Each particle was represented through 90 PVs (30 of low frequency, 30 of mid frequency and 30 of high frequency). The particle *unter* had only 38 corresponding PVs.

One of the problems that arose in the creation of the data set was the fact that PVs may be easily confounded with prefix verbs. Prefix verbs are not separable at all and have a quite different syntactic behaviour. For example, the verb *umarmen* (*to hug*), has the prefix *um-* which has a homophonic verb particle. There are also verbs which are ambiguous between a prefix verb and a particle verb interpretation: *übersetzen* may be a PV (meaning *to cross a body of water*) or a prefix verb (meaning *to translate*). Four of the verb particles we used – *um*, *unter*, *über* and *durch* – are ambiguous between prefix and particle use. Since the data set with 938 PVs was generated automatically, there was a number of verbs which were ambiguous between particle verb and prefix verb interpretation. In order to make sure that no prefix verbs were included in our data set, we manually edited the list of 938 PVs by excluding such cases. PVs whose verbal base corresponds to a prefix verb were excluded as well (e.g. *ausverkaufen/to sell off*).

We know from previous experiments that PVs with very low and very high frequencies tend to be problematic for automatic assessment: low frequency items are likely to present data-sparseness problems and high-frequency items tend to be highly lexicalized and very idiosyncratic in their behavior. For this reason we excluded the top 20 frequent PVs and the 20 PV with the lowest frequency. This revision of the original list (938 PVs) resulted in a new list of 400 PVs. The data set for our experiments contains 400 PVs (targets). Each PV is represented through a six-dimensional feature vector. Features correspond to the different syntactic paradigms a PV can occur in plus the syntactically non-separated use of a PV, which is a sum of the paradigms: PP, IZU, INF and FIN. The values are normalized (relative) frequencies over the total frequency of a PV. For feature extraction only counts from the SdeWaC corpus were used. Table 1 shows a sample of vectors.

3.4 Clustering Experiments

In order to assess our first three hypotheses (H1-H3) we carried our clustering experiments (see section 3.5 for H4). The goal of clustering algorithms is to partition a set of objects in groups (clusters), so that the objects within one group are similar to each other and dissimilar to those in other groups. Object are compared based on particular features. To perform the task of analyzing German PVs empirically, we use a hard clustering method, namely the simple K-means algorithm.[3] On the account that PVs are represented in terms of feature vectors, similarity (or dissimilarity) between two objects is defined as the euclidean distance between the corresponding vectors. The greater the distance, the more dissimilar the objects are; they are then assigned to different clusters.

One of the challenges of the K-means algorithm is to find the optimal K, which must be specified in advance so that the structure of the data can be revealed. The experiments were carried out with different K values: K = 3, 5, 7, 11, 15, 20 for H1 (particles) and H2 (frequency). In addition to these values K = 4 clustering experiments were performed for the H3 (ambiguity), because for this hypothesis we used a reference set with 4 classes of different ambiguity levels (cf. 3.4.1 below).

3.4.1 Reference Sets

For evaluation we used a series of reference sets against which the clusterings were compared and the evaluation metrics were computed. Each reference set was built to represent the information corresponding to our hypotheses listed in section 3.1. For the partitioning of data into ambiguity classes, for example, the degree of ambiguity of a verb was determined by the information gained from different dictionaries. Due to the inconsistency in the

[3]We use the WEKA implementation (Witten and Frank, 2005)

degree of ambiguity a verb has in different dictionaries – GermaNet (Hamp and Feldweg, 1997), Wiktionary,[4] Duden[5] and DictCC[6] – mean ambiguity was calculated and a verb was assigned to a certain class according to its mean ambiguity value. In sum, we used the following reference sets:

- RS1 corresponds to H1, the particle hypothesis. RS1 contains 11 classes which correspond to the 11 particles described in section 3.3. For example *an|sehen* belongs to the class *an*. In this case class affiliation can be defined unambiguously for each verb.
- RS2 models the corpus frequency of PVs (H2): The PVs of RS2 are divided in three classes: H(high), M(mid) and L(low). Because we discarded the 20 most frequent and the 20 PVs with the lowest frequency from the original list of 938 PVs we also had to randomly reduce the mid frequency class in order to obtain a balanced representation of each class. This led to a selection of 88 high-frequency, 80 mid-frequency and 74 low-frequency PVs.
- RS3 captures the ambiguity of PVs. Ambiguity of each PVs is determined by computing the rounded mean ambiguity out of four ambiguities gained from the four sources mentioned above: GermaNet, Wiktionary, Duden and DicctCC. The RS has four classes for unambiguous PVs (A1), and PVs which have two, three or more than three readings (A2, A3 and AG3). *Nach|zahlen*, for example, is unambiguous (A1) and means *to pay later*; *an|sehen* from example (3) has more than three meanings (AG3) and may mean *to look at*, *to watch*, *to have the look of something*, *to consider someone as*.

Note that there is no reference set for Hypothesis 4 (average sentence length), because we do not test this hypothesis with clustering techniques and use corpus counts of sentence lengths (on a continuous scale) instead.

3.4.2 Evaluation

We evaluated the clusterings in terms of *Purity* (Manning et al., 2008), *Rand Index* and *Adjusted Rand Index* (Rand, 1971; Hubert and Arabie, 1985). *Purity* is a measure with values between 0 and 1 which captures the purity of individual clusters in terms of the ratio between the number of elements of the majority class in each cluster and the total of elements in the cluster. A perfect clustering will have a purity of 1. What purity does not capture is the amount of clusters over which each target class is distributed. That means that also non-perfect clusters may achieve a purity of 1 if there are more clusters than target classes. As long as the number of clusters is constant, however, purity is a good and intuitive approximation to clustering evaluation.

The *Rand Index* (RI) looks at pairs of elements and assesses whether they have been correctly placed in the same cluster (which is correct if they pertain to the same target class) or in different clusters (correct if they belong to different target classes). RI is sensitive to the number of non- empty clusters and can capture both the quality of individual clusters and the amount to which elements of target categories have been grouped together. RI looks at pair-wise decisions, which makes it also applicable to comparison with reference data which lists pairwise class membership decisions, but does not necessarily define closed sets of reference classes.

The *Adjusted Rand Index* (ARI) is a version of RI which is corrected for chance. While RI has values between 0 and 1, ARI can have negative values; 1 again represents a perfect clustering. An ARI of 0 indicates a clustering which is close to the random level. While ARI is corrected for chance, the two metrics require a baseline for comparison. For this purpose we use random clustering, where each PV is assigned to a random cluster. In our case we averaged the values over 100 random clusterings.

3.5 Correlations

In order to tackle hypothesis 4 we used corpus-extracted counts of sentence lengths. In this case we deviate from the clustering approach because sentence length is a feature which is easily extracted from the corpus and there appears to be no natural way to bin sentence length into reference classes. In order to test H4, we matched the average sentence length with the percentage of verb realizations in the separated paradigm per PV. Each PV is thus matched to a point in a two-dimensional

[4] http://wiktionary.org

[5] We consulted the online edition: http://www.duden.de

[6] http://www.dict.cc

	K	RS1:Particle			RS2:Frequency			RS3:Ambiguity		
		Purity	ARI	RI	Purity	ARI	RI	Purity	ARI	RI
K-Means	3	0.16	0.0168	0.62	0.42	0.0174	0.56	0.42	0.00422	0.56
	4							0.40	-0.00127	0.60
	5	0.17	0.0150	0.74	0.41	0.0101	0.59	0.40	-0.00727	0.61
	7	0.18	0.0110	0.77	0.44	0.0066	0.62	0.40	-0.00354	0.65
	11	0.23	0.0173	0.83	0.47	0.0098	0.64	0.41	0.00002	0.67
	15	0.23	0.0101	0.84	0.52	0.1579	0.65	0.43	0.00512	0.68
	20	0.25	0.0108	0.85	0.53	0.0132	0.65	0.44	0.00221	0.68
Random Clustering	3	0.15	-0.0000	0.63	0.39	0.0001	0.56	0.38	0.0005	0.57
	4							0.38	0.0002	0.60
	5	0.16	0.0006	0.74	0.40	-0.0002	0.60	0.39	0.0002	0.62
	7	0.17	0.0000	0.78	0.42	0.0004	0.62	0.39	-0.0002	0.65
	11	0.19	-0.0000	0.83	0.44	0.0007	0.64	0.41	-0.0002	0.67
	15	0.21	0.00000	0.85	0.46	-0.0019	0.65	0.42	0.0001	0.68
	20	0.22	-0.00000	0.86	0.48	0.0001	0.65	0.43	0.0007	0.69

Table 2: Results of the clustering experiments for hypotheses 1, 2 and 3

space. Then a simple least squared error regression is applied, using the *lm* function of the R language (R Development Core Team, 2008).

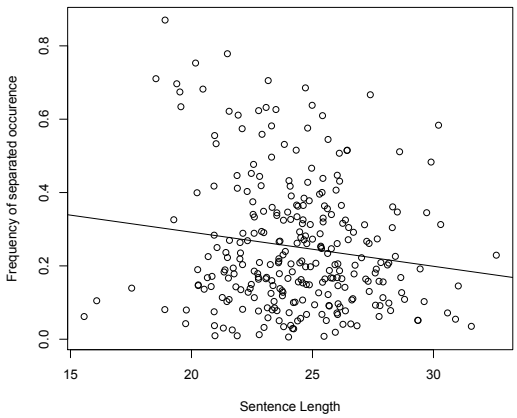


Figure 3: The relation between proportion of the syntactically separated paradigm and average sentence length

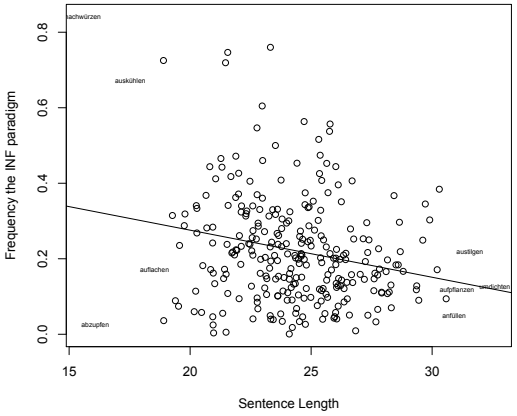


Figure 4: The relation between proportion of the infinite paradigm and average sentence length

4 Results and Discussion

Table 2 shows the results of the clustering experiments for hypotheses H1 to H3. The top part lists the results obtained with K-means while the lower part lists the results of the baseline random clustering. It can be seen that the results are nearly consistently above the baseline, but the difference is not significant. The factors we capture by the reference sets which correspond to these hypotheses seem to have a certain effect on the proportion of PVs with which they occur syntactically separated, but by themselves they do not explain the observed variation with which PVs occur in different syntactic paradigms.

Figure 3 plots the distribution of sentence lengths (H4) against the distribution of relative frequencies (proportions) for the separated paradigm per PV. The black line represents the regression line for the syntactically separated paradigm depending on the average sentence length per PV. The scatterplot suggests that there is a direct relation between the two factors – there is a tendency that PVs that occur very often separately

also tend to occur in shorter sentences – but it is not very strong. This first impression is corroborated by a simple regression analysis which only examines the correlation between average sentence lengths and the relative frequency of the syntactically separated paradigm: the correlation between sentence length and the frequency of the separated paradigm reaches significance, but not with a very high confidence (p=0.018).

This analysis models the branching of the top-node of the tree in Figure 1, the distinction between syntactically separated vs non-separated. In order to check possible correlations between sentence length and *all* syntactic paradigms we carried out a multivariate regression analysis with sentence length as the dependent variable and each of the syntactic paradigms as independent variables (PP, INF, IZU, FIN, SEP), but in this analysis none of the independent variables showed a significant correlation with sentence length.

Just for the sake of error analysis and visual data exploration we also examined the relation of individual syntactic paradigms to sentence length. The most notable correlation we found is the one between the INF paradigm and sentence length. The corresponding scatterplot can be seen in Figure 4. This scatterplot resembles Figure 3, but also shows some differences. The most interesting observation which can be made here concerns the outliers on the x-axis, which is the reason why they are plotted out as PV lemmas (the dots correspond to the rest of all the PVs). The outliers in the left upper corner, the PVs that occur in short sentences and tend to occur very frequently in the INF paradigm all appear in the cooking domain, such as *nach|würzen* (*to add additional spice*), *aus|kühlen* (*to cool down*) or *auf|kochen* (*to reboil*). This hints at an influence of the text domain.

We have made some observations which are worth a closer investigation. We have noticed that a number of PVs sharing the same BV tend to be assigned to the same cluster. What was remarkable was the behavior of PVs with the BV *bauen* (*to build*), i.e. *auf|bauen* (*to build up*), *ab|bauen* (*to dismantle/reduce/mine*), *nach|bauen* (*to reversely engineer*), *aus|bauen* (*to enlarge/equip*), *ein|bauen* (*to install/integrate/build in*), which were very often found in the same cluster across different clusterings. This behavior was observed also in synonym clustering: some synonym pairs which share the same BV were repeatedly found in the same cluster.

Some verbs tend to appear in more formal register and hence have other preferences for syntactic paradigms. To give an example: *zu|senden* and *zu|schicken* (both meaning *to sent to*) can be used in different registers. *Zu|senden* is predominantly used in formal style, whereas *zu|schicken* tends to occur in informal style. This, again, highlights the influence of pragmatic factors, such as register and genre.

Finally we found that much noise was introduced into our data by errors stemming from the linguistic preprocessing. We found errors in the POS tags, most notably verb particles which were tagged as preposition and vice versa. This also means that the syntactic dependency between base verb and particle is not identified correctly. Often the lemmas of PVs were predicted incorrectly, incorporating functional morphemes into the lemma (e.g. *auf|zumachen* instead of *auf|machen*). This shows again that a better treatment of PVs in linguistic processors would be very desirable. A better understanding of empirical aspects of PVs could contribute to an improvement.

5 Conclusions

In this paper we described the empirical distribution of the proportions of German particle verbs with respect to their occurrence in different syntactic paradigms. We were able to show that there is observable variation in the frequencies in which PVs occur in different syntactic paradigms. We could find no explanation for this variation in the relevant literature. We parted from the basic hypothesis that there must be underlying factors which influence the behaviour of PVs in this respect. Building on this assumption, we formulated and tested a series of syntactic, semantic and pragmatic hypotheses about the source of the variation.

We could not provide a definitive answer to our initial question of what factors determine the proportional distributions of PVs over the different syntactic paradigms, but our findings suggest that pragmatic factors, such as genre and register, play an important role. We consider the problem well worth further study, considering that a better understanding of the behaviour of PVs has a high potential to improve the treatment of PVs in NLP tasks such as parsing and machine translation. In future work we plan to take pragmatic factors more strongly into account.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: a Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.

Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is Not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Stroudsburg, PA, USA.

Stefan Bott and Sabine Schulte im Walde. 2014. Optimizing a Distributional Semantic Model for the Prediction of German Particle Verb Compositionality. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 509–516, Reykjavik, Island.

Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC – A Corpus of Parsable Sentences from the Web. In *Language Processing and Knowledge in the Web*, pages 61–68. Springer.

Arne Fitschen. 2004. Ein computerlinguistisches Lexikon als komplexes System. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*.

Nanna Fuhrhop. 2007. *Zwischen Wort und Syntagma: Zur grammatischen Fundierung der Getrennt-und Zusammenschreibung*. Walter de Gruyter.

Stefan Gries. 2001. A Multifactorial Analysis of Syntactic Variation: Particle Movement Revisited. *Journal of Quantitative Linguistics*, 8(1):33–50.

Stefan Gries. 2002. The Influence of Processing on Syntactic Variation: Particle Placement in English. *Verb-particle Explorations*, 1:269–288.

Stefan Gries. 2011. Acquiring Particle Placement in English: A Corpus-Based erspective. *Morphosyntactic Alternations in English: Functional and Cognitive Perspectives. London/Oakville, CT: Equinox*, pages 235–263.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet – A Lexical-Semantic Net for German. In *Proceedings of the ACL Workshop on Automatic Information Extraction and Building Lexical Semantic Resources for NLP Applications*, pages 9–15, Madrid, Spain.

Lawrence Hubert and Phipps Arabie. 1985. Comparing Partitions. *Journal of Classification*, 2(1):193–218.

Joachim Jacobs. 2005. *Spatien: Zum System der Getrennt-und Zusammenschreibung im heutigen Deutsch*, volume 8. Walter de Gruyter.

Fritz Kliche. 2009. Zur Semantik der Partikelverben auf '*ab*'. Eine Studie im Rahmen der Diskursepräentationstheorie. *Masters thesis, Universität Tübingen*.

Andrea Lechler and Antje Roßdeutscher. 2009. German Particle Verbs with *auf*. Reconstructing their Composition in a DRT-based Framework. *Linguistische Berichte*, (220):439–478.

Anke Lüdeling. 2001. *On Particle Verbs and Similar Constructions in German*. CSLI, Stanford.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge.

Stefan Müller. 2001. Syntax or Morphology: German Particle Verbs Revisited. In Nicole Dehé, Ray Jackendoff, Andrew McIntyre, and Silke Urban, editors, *Verb-Particle Explorations*, Interface Explorations. Mouton de Gruyter, Berlin, New York.

Stefan Müller. 2003. Solving the Bracketing Paradox: an Analysis of the Morphology of German Particle Verbs. *Journal of Linguistics*, 39(2):275–325.

R Development Core Team, 2008. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org.

William M. Rand. 1971. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850.

Roland Schäfer and Felix Bildhauer. 2012. Building Large Corpora from the Web Using a New Efficient Tool Chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 486–493.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, volume 12, pages 44–49. Manchester, UK.

Helmut Schmid. 2000. Unsupervised Learning of Period Disambiguation for Tokenisation. Technical report, Universität Stuttgart.

Sabine Schulte im Walde. 2004. Identification, Quantitative Description, and Preliminary Distributional Analysis of German Particle Verbs. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, pages 85–88, Stroudsburg, PA, USA.

Sabine Schulte im Walde. 2005. Exploring Features to Identify Semantic Nearest Neighbours: A Case Study on German Particle Verbs. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 608–614.

Sylvia Springorum. 2009. Zur Semantik der Partikelverben mit *an*. Eine Studie zur Konstruktion ihrer Bedeutung im Rahmen der Diskursrepräsentationstheorie. *Studienarbeit. Universität Stuttgart*.

Barbara Stiebels. 1996. *Lexikalische Argumente und Adjunkte. Zum semantischen Beitrag von verbalen Präfixen und Partikeln*. Number 39. Akademie Verlag.

Christine Thielen, Anne Schiller, Simone Teufel, and Christine Stöckert. 1999. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universitiät Stuttgart and Universität Tübingen.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

WISE: A Web-Interface for Spelling Error Recognition for German: A Description and Evaluation of the Underlying Algorithm

Kay Berkling
Cooperative State University
Karlsruhe, Germany
berkling@dhbw-karlsruhe.de

Rémi Lavalley
Cooperative State University
Karlsruhe, Germany
remiL@singularity.fr

Abstract

This paper evaluates an automatic spelling error tagger that is available via web interface. After explaining the existing error tags in detail, the accuracy of the tool is validated against a publicly available database containing around 1700 written texts ranging from first grade to eighth grade. The precision of the tool ranges from 83% to 100%. Some basic statistics about spelling errors in the existing data set are given to demonstrate potential research areas. The site can be used to further explore this data. In addition, new data can be donated and explored. This process will be described in detail.

1 Introduction

This paper evaluates an automatic spelling error tagger. The annotation and study of spelling errors represents one dimension towards gaining a deeper understanding about children's writing acquisition. A database with 1701 spontaneously written texts from grades 1-8, including *Grundschule*, *Hauptschule* and *Realschule* is used for evaluation. This corpus is described separately in (Berkling et al., 2014; Lavalley et al., 2015). The algorithm's evaluation is reported in this paper. The described website offers other researchers the possibility of browsing through this data and the respective error annotations. For a given combination of features, the user is able to obtain a list of words from spellers matching that feature. Thus, one can compare girls' vs. boys' spellings or study the development of the ability to capitalize correctly from second grade until eighth grade. As examples of how to use the website, we will present some of the spelling errors in this database along with information about the precision of the automatically tagged data.

The rest of the paper is structured as follows. After a brief review of the data collection in Section 2, Section 3 and Section 4 provide details and results on the quality of the automatic spelling error annotation of the data. Section 5 presents some of the statistics obtained when using the website. Section 6 describes the use of the GUI. Section 7 concludes this paper.

2 Evaluation Data

This section briefly describes the collected data and the data transcription and annotation methods. The data described in this paper was collected during the years 2011–2013 by the University of Education, Karlsruhe (Berkling et al., 2014). Text written by children of various ages was collected at schools in and around Karlsruhe, at elementary schools (*Grundschule*) and two types of secondary schools, (*Hauptschule*[1] and *Realschule*[2]). The data was then prepared for automated processing (Lavalley et al., 2015) of orthographic error classification (Berkling et al., 2011).

2.1 Text Elicitation

In order to collect data, children were asked to write as verbose a text as possible.

Grades 1 to 4: Either the picture book "Der kultivierte Wolf" (The Cultivated Wolf (Bloom and Biet, 2008) about a wolf that learns how to read) or "Stimmen im Park" (Voices in the Park (Browne, 1998) about children playing in the park) was read to the students. Afterwards the students were asked to continue the story or write their own story on that topic. This resulted in spontaneously written texts.

Grades 5 to 8: The instruction to the writing task was simply given as either :"Imagine the

[1] Hauptschule: Grades 5-9, offering lower secondary education for anyone.

[2] Realschule: Grades 5-10, offering medium secondary education designated for apprenticeship.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 87–96, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

Category/Level	Explanation
Word	**Sentence Dependencies**
GrS	missed capitalization
GrS_S	beginning of sentence
GrS_other	within sentence
KS	missed decapitalization
KS_WA	beginning of word
KS_WI	within word
Morpheme	**Morpheme Endings**
KA	error due to devoicing:
_AV	<b>,<d>,<g> for /p/,/t/,/k/
_G	<g> for /ç/ after /ɪ/
_S	<s> for /z/
Syllable	**short vs. long vowels**
V_KV (short)	missing silent consonant
V_ie (norm)	incorrect usage of <ie> for /iː/
V_i (exception)	incorrect usage of <i> for /iː/
V_ih (few, frequent)	incorrect usage of <ih> for /iː/
V_LV_h	wrong <ah>,<eh>,<oh>,<uh>
V_LV_aa	wrong <aa>,<ee>,<oo>

Table 1: Spelling Error Categories.

world in 20 years. What has changed? How do you envision your life in 20 years? How, where and with whom do you live? Write a text as detailed as possible, so we can understand you and your ideas."; or "A day with ..." followed by the student's chosen favorite star.

2.2 Text Transcription

All texts were transcribed and anonymized as described in Lavalley et al. (2015). They are available as $target$ text (child's intended text) and $achieved$ text (child's actual writing, including all spelling errors). This combination serves as the foundation for tagging the spelling errors automatically.

2.3 Meta Data

Meta data will serve as a way of indexing the data for statistical analysis on the website. This includes (school type, grade, age, gender, and languages spoken at home) as shown in Figure 8.

3 Definition of Algorithms

After a brief description of the error categories on a theoretical level, the algorithm will be defined.

3.1 Error Categories

All texts are automatically annotated with a number of defined spelling error categories listed in Table 1. The error categories are used as defined in (Fay, 2010; Scholze-Stubenrecht, 2004).

Spelling errors for the German language can be defined based on the level at which rules about language are applied when choosing a grapheme. Table 1 distinguishes word, morpheme and syllable-level spelling issues for the categories that are evaluated in this paper.

Capitalization: Capitalization in German depends on the grammatical function of the word within a sentence. Both incorrect capitalization as well as incorrect non-capitalization are tracked as spelling errors at this level.

Devoicing: At the end of syllables and morphemes, devoicing, **A**uslaut**v**erhärtung (AV), occurs in German pronunciation for most dialects. Here are some examples for this category:

- "Gans" is not pronounced with a soft /z/, which is the normal pronunciation of the grapheme <s>. Because it is pronounced as /s/ it can lead to misspellings like "Ganz" or "Gants".
- "Gras" follows the typical pattern of <ß> used after long vowel to create the /s/ sound, which could lead writers to misspell this pattern as "Grass" or "Graß".
- "Hand" is pronounced as hant /hant/ and therefore often misspelled with a final <t>.
- "lustig" <g> after /ɪ/ is often pronounced as /ç/ and therefore mistakenly spelled as <ch>.

Vowel Length: The short/lax vowels in German are <a>, <e>, <i>, <o>, and <u> while the long/tense vowels in German are <a>, <e>, <ie>, <o>, and <u>. The information of length is not carried in the vowel grapheme except for the case of <ie> vs <i>. Yet, vowel length is semantically discriminative. In analogy to the English "silent <e>" ("cut" vs. "cute"), German orthography makes use of a "silent consonant". By doubling the consonant letter after the vowel, the length of the preceding vowel phoneme is shortened. (For example, "Hüte" vs. "Hütte" changes /yː/ to /y/. This happens in the German bi-syllabic structure called Trochee (stressed, unstressed: ¯ ˘), where the second syllable contains an <e> pronounced as /ə/. The convention is maintained with changes at the morpheme boundary (for example: können vs. könnt).

Exceptions to the system of denoting long vowels vary. /iː/ can be marked as <ie> (default), <i> (exception), <ih> (mostly pronouns, like "ihn" or "ihr"). All other vowels, <a>, <e>, <o>,

and <u> can be be followed by <h> in certain complex syllable endings to mark length. Furthermore, <a>, <e> and <o> can be doubled (<aa>, <ee> and <oo>) to mark length. This form is rare.

For each of the words in the corpus the specific error types are annotated with $Basis$ (base rate) indicating whether the error could have theoretically occurred. In addition, the $Error$ rate denoting an actual occurrence of the error. This differentiation supports error normalization across texts and students.

3.2 Annotation Algorithm

The speech synthesis system MARY (Schröder and Trouvain, 2003) is used to obtain the pronunciation of both target and achieved texts. From there, a simultaneous grapheme and phoneme segmentation and alignment is performed as described in (Berkling et al., 2011). Together with information about syllable boundaries, syllable stress and morpheme boundaries obtained from BALLOON (Reichel, 2012), spelling errors are automatically identified using a rule-based system.

Capitalization: Capitalization and de-capitalization is performed by comparing the achieved and target grapheme. If the target is capitalized then the $Basis$ is tagged. If the achieved grapheme is not, then the $Error$ is tagged. If this happens at the beginning of a sentence, then the subcategory GrS_S is used. For all other words, the subcategory GrS_other is used. If the target is de-capitalized then the $Basis$ for KS_WA category is tagged. If the achieved grapheme is capitalized then the $Error$ is tagged. For a letter that is wrongly capitalized within word the $Error$ for KS_WI is tagged ("Haus" misspelled as "haus"). $Bases = 1$ for KS_WI if there is at least one de-capitalized letter in the word.

Devoicing: Devoicing appears at the end of syllables and morpheme boundaries for certain consonants. The $Basis$ of each KA_AV is tagged for graphemes consonants <b>, <d>, <g>, <ng>, <v>, <w> unless these appear before a vowel or glottal stop. The $Error$ is tagged if these are misspelled as <p>, <t>, <k>, <n>, <f> ("Hant" instead of "Hand"). The $Basis$ of each KA_S is tagged for grapheme <s> pronounced /s/ unless part of an inflectional morpheme in an unstressed syllable preceded by /ə/ or as a "Fugen-s" (Geburt.**s**.tag). The $Error$ is tagged if misspelled as <tz>, <ts>, <z>, <ß>, or <ss> ("Gans" misspelled as "Ganz"). In the current version a $Basis$ is not marked after voiceless consonants because the devoicing remains even if the words are followed by a vowel.[3] The $Basis$ of each KA_G is tagged for graphemes <g> if pronounced /ç/ or /x/ and preceded by /ɪ/. The $Error$ is tagged if the misspelling is <ch> ("lustig" misspelled as "lustich").

Vowel Length: The $Basis$ of each V_KV is tagged when there are double consonant graphemes: <bb>, <dd>, <ff>, <gg>, <ll>, <mm>, <nn>, <pp>, <rr>, <ss>, <tt>, <ck>, or <tz>. The preceding phoneme must be a short vowel and the double consonant must be at the end of a morpheme boundary. The $Error$ is tagged if the grapheme is not correct ("rennen" misspelled as "renen").

Each of V_ie V_i V_ih V_LV_aa is easy to identify. If there is a grapheme <ie>,<i>,<ih> corresponding to the phoneme /iː/ or <aa>,<ee>,<oo> in the target then the $Basis$ is tagged. If the achieved grapheme does not match the target then the $Error$ for the corresponding category is tagged. V_LV_h works similarly, except that there may not be a morpheme boundary within the graphemes <ah>,<eh>,<oh>,<uh>,<äh>,<öh>,<üh>. These graphemes are then tagged with $Basis$. If achieved and target graphemes do not match, an $Error$ is tagged.

4 Evaluation of Error Annotations

The correctness of the tool was measured by manually checking 2000 randomly chosen achieved-target pairs from the corpus, about 10% of the entire word count. The results are given in Table 2.

4.1 Human-Machine Agreement

Regarding the (de-)capitalization errors, the machine can always detect lower and upper case mismatches correctly assuming that correct sentence boundaries are given as input in the case of GrS_S. Therefore, no human agreement is reported. The other categories perform as listed in Table 2.

For each of the categories, the table reports positive (the $Basis$ or $Error$ count for the spelling category > 0) and negative (the $Basis$ or $Error$ count for the spelling category $= 0$) tags. (Note

[3] Devoiced Consonants are: p t k pf ts tʃ f s ʃ ɕ x

Category	Basis		Error	
KA_AV	true(false)		true(false)	
positive	209 (0)	P=1	17(1)	P=.94
negative	1798 (3)	R=.98	1981(1)	R=.94
KA_G	true(false)		true(false)	
positive	28 (0)	P=1	3(0)	P=1
negative	1972 (0)	R=1	1997(0)	R=1
KA_S	true(false)		true(false)	
positive	65 (4)	P=.94	7(0)	P=1
negative	1917 (14)	R=.82	1993(0)	R=1
KV	true(false)		true(false)	
positive	388 (7)	P=.98	115 (5)	P=.96
negative	1596 (17)	R=.96	1878 (2)	R=.98
ie	true(false)		true(false)	
positive	186 (1)	P= .99	49 (3)	P=.94
negative	1809 (6)	R= .96	1945 (4)	R=.92
i	true(false)		true(false)	
positive	55 (11)	P=.83	9(0)	P=1
negative	1939 (2)	R=.96	1991(0)	R=1
aa,ee,...	true(false)		true(false)	
positive	9(0)	P=1	3(0)	P=1
negative	1991(0)	R=1	1997(0)	R=1
ah,eh,oh,uh	true(false)		true(false)	
positive	74(0)	P=1	11(1)	P=.91
negative	1928(0)	R=1	1988(0)	R=1
ih	true(false)		true(false)	
positive	4(0)	P=1	1(0)	P=1
negative	1996(0)	R=1	1999(0)	R=1

Table 2: Machine performance on $Basis$ and $Error$ as evaluated by human expert on 10% of data set (2000 randomly chosen words). Precision: $P = tp/(tp + fp)$, Recall: $R = tp/(tp + fn)$

that a particular error category can appear more than once in a word, like "A**ff**enmu**tt**er", which explains why some numbers do not add up to 2000.) The human rater then sorts both positive and negative tags into true (correctly identified) and false (incorrectly identified). The results are listed along with precision (class is correctly predicted) and recall (ability to select instances of class from data).

Some of the errors that the tool misses are explained in more detail below.

Morpheme: Devoicing - KA

KA_AV reported a false error in this pair of (target achieved) (see Section 2): (entschie**d** entschid). In the pair (gesa**g**t gesat) both basis and error at grapheme <g> were missed. "An**g**st", "krie**g**st" did not get tagged correctly as basis. In the latter case <g> at end of a morpheme boundary should have been tagged with a consonant devoicing. In the case of "Angst", <g> is not located at a morpheme boundary, so it can be debated whether this case falls into this particular category.

KA_G made no mistakes in identifying basis or error tags correctly.

KA_S missed a number of words: "etwas", "Applaus", "ausleiht", "Auspuff", "beste", "es", "Gäste", "heraus", "Krebs". Others were falsely identified: "anstrengend", "bisschen".

Syllable: Silent Consonant - KV

The following pairs were tagged as KV but are not strictly speaking a KV pattern because the double consonant did not appear within a trochee ("rennen") nor at the end of a morpheme boundary (as in "rennt"). It is therefore debatable whether this item is falsly tagged: (allein alein). Other words are imported: (Gorilla Goriler), (Horror Horer), (Installateur Instalatör), (intelligenten inteligenten), (Tickets Tickets) and can be argued not to fall under regular spelling patterns. It should be possible, however, to identify these as non KV because there is neither a morpheme boundary nor a reduced syllable containing an <e> following the double consonant. Another exception is the wrongly tagged (Hartz Hartz) which should not be identified as KV as the preceding r-colored vowel is long (<ar> aɒ) and therefore does not follow the KV pattern.

The following list of words was missed by the algorithm: "Auspuff", "ertappten", "gestoppt", "gezockt", "hockte", "Lego§sammler", "Mucks", "öffnet", "reinlassen", "rockten", "selbstbewusst", "Truppe", "wussten", "Zigaretten", "Zocken", "frisst", "höllische".

Syllable: Regular /iː/ <ie>

Missed words include: "Dienstag§mittag, "die", "Dienerinnen", and "Fernbedienung". "Fossilien" was mistakenly tagged. (sieht siet) and (zieht ziet) were mistakenly tagged with errors as the missing <h> does not belong to the <ie> error category.

Syllable: Irregular /iː/ <i>

The following were badly annotated as all these <i> are pronounced /ɪ/: "Automechaniker", "Chemielaborantin", "direkten", "finanziellen", "Gitarre", "Grafiker", "Mechatroniker", "Navi", "Plastikindustrie". One mistake identified a transcription error on the achieved side: "Abteil**i**ungsleiter", correctly spelled without the <i> in the center as "Abteilungsleiter".

Syllable: Irregular /i:/ <ih>

There were no mistakes in this category.

Syllable: Irregular long vowel (<aa>, ...)

There were no mistakes in this category.

Syllable: Irregular long vowel (ah, ...)

One word was mistakenly identified as an error for this category: (Ru**t**schbahn Ru**d**schbahn), which is a mis-alignment problem due to the spelling error of <dsch>.

The next step is to diagnose the cause for the above listed problems. Causes for these kinds of errors can usually be removed and usually belong to one or the other of the following category of problems:

- In some of these cases the morpheme boundary or the pronunciation was not returned correctly by the underlying tool. (These are fixed by providing the correct pronunciation through an amended pronunciation file and reporting the problem to the research group for the underlying morpheme tagging system, which usually fixes the problem.)
- The rules of the algorithm may need to be refined.
- Foreign words did not get tagged correctly.

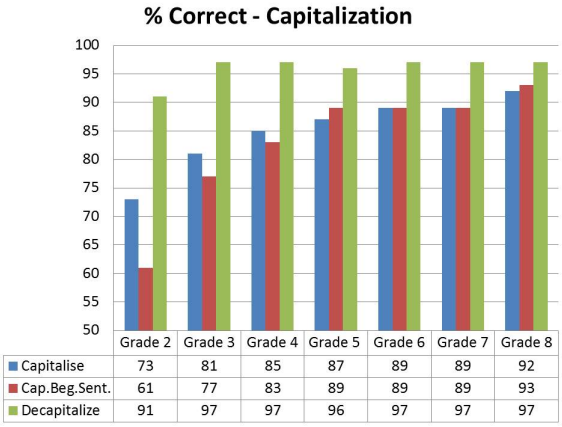
5 Data Exploration

A broad overview of results for the spelling error analysis on the entire data set are given in this section. Depicted for each error category is the normalized value of the fraction of correctly spelled words of that category as given by Equation 1.

$$\frac{\#Basis(CAT) - \#Errors(CAT)}{\#Basis(CAT)} \quad (1)$$

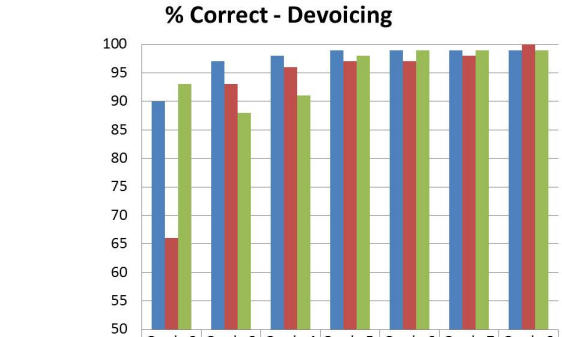
Figure 1 shows that correct (de-)capitalization improves with each grade. However, correct capitalization is still misspelled at a rate of 10% even in eighth grade. Capitalization is difficult within sentence, as German has more complex rules than most other languages. But even at the beginning of a sentence, trivial capitalization is not mastered.

Figure 2 depicts the development of spelling errors for the category of devoiced consonants at the ends of morpheme or syllable boundaries explained in Section 3. The devoiced <s> is the most difficult but all three categories improve rapidly after third grade. Both <ig> (average of .3 occurrences per text in Grade 8) and <s> (average of 3 occurrences per text in Grade 8) are much more rare than AV (average of 10 occurrences per text in Grade 8). This may in part explain the outlier in Grade 2 for /s/.



	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8
Capitalise	73	81	85	87	89	89	92
Cap.Beg.Sent.	61	77	83	89	89	89	93
Decapitalize	91	97	97	96	97	97	97

Figure 1: % Correct for capitalization in general and at sentence start. % Correct decapitalization.

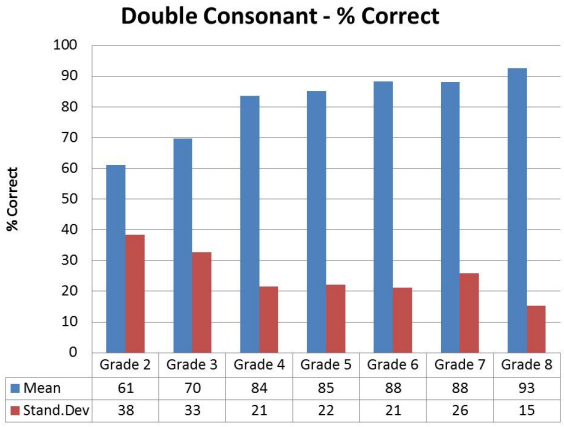


	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8
Voiced Consonant	90	97	98	99	99	99	99
<ig>	66	93	96	97	97	98	100
<s> as /s/	93	88	91	98	99	99	99

Figure 2: % Correct at morpheme and syllable endings for "Auslautverhärtung" - devoicing of voiced consonants in general (AV) for <s> as /s/ and <ig>.

Figure 3 shows the statistical distribution of errors for the example of V_KV. There is a very large variance in occurrence frequency of both base and errors. Mean and variance for % correct for this category show that there is need for further analysis that is beyond the scope of this paper.

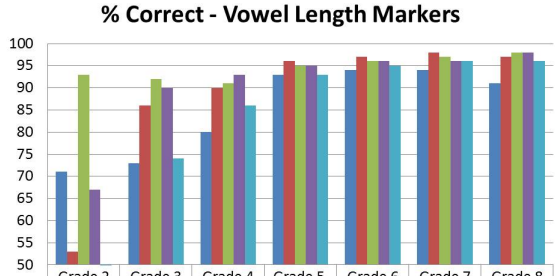
Figure 4 shows the development of spelling errors regarding the marking of long vowels. These include the various ways of denoting long /i:/ as <ih>, <i>, <ie> (default). The marking of <i> is mastered most quickly (3 occurrences per text on average in Grade 8). Further it is surprising



	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8
Mean	61	70	84	85	88	88	93
Stand.Dev	38	33	21	22	21	26	15

Figure 3: Statistics for double (silent) consonant (V_KV) notation to shorten preceding vowel. The graph shows the large variation in the statistics, indicating that a more detailed analysis will be necessary.

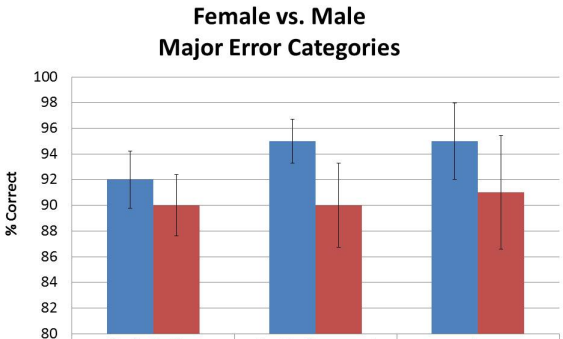
to see how long it takes to master <ih> notation, given the small finite set of high-frequency words that contain it.[4] The default <ie> reaches 90% correctness by grade 3. This performance is reached by Grade 4 for LV_h and in Grade 5 for LV_aa. Only the default category <ie> has a high frequency with 7 occurrences on average in a text by Grade 8.



	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8
aa	71	73	80	93	94	94	91
h	53	86	90	96	97	98	97
i	93	92	91	95	96	97	98
ie	67	90	93	95	96	96	98
ih	19	74	86	93	95	96	96

Figure 4: % Correct for various ways of indicating long vowels, by doubling the vowel letter, adding a silent <h> and special vowel <i>, <ie>, <ih>.

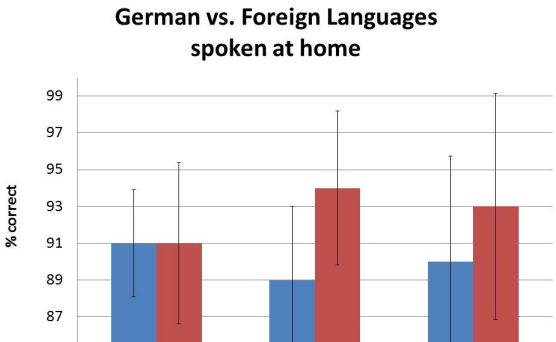
It can be seen that certain error categories are more prevalent than others even into the upper grades. Taking a closer look at categories that do not reach full correctness and that are frequent, one can compare the results given certain meta-data. For example, gender, school-type or multi-linguality.



	Capitalization	Double Consonant	<ie>
Female	92	95	95
Male	90	90	91

Figure 5: % Correct comparing female vs. male writing skills on major error categories.

Figure 5 shows that there is only one category with significant differences between female and male students in 8^{th} grade, namely the usage of double (silent) consonants for vowel duration (V_KV). By Grade 8, 27% of male students and 24% of female students do not reach above 90% correctness regarding double consonant marking. The other two categories only show tendencies. In contrast, Figure 6 shows no significant differences based on languages spoken in 8^{th} grade.



	Capitalization	Double Consonant	<ie>
German	91	89	90
Other	91	94	93

Figure 6: % Correct comparing German speaking students compared to those that speak other languages at home ($Other$).

Figure 7 shows that there is a significant difference in correct capitalization for Realschule vs. Haupt- and Werkschule in Grade 8. The other categories show no significant differences,

[4]High-frequency words according to general statistics may not apply to children's writings. For example words with <ih> occur .6 times on average per text, but are listed in the top 100 words representing 45% of German text.

only tendencies. 37% of students in Werk- and Hauptschule and 26% of students in Realschule do not reach 90% correct command of capitalization.

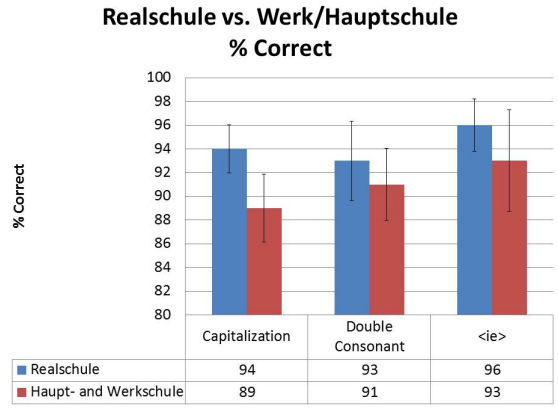


Figure 7: % Correct comparing Realschule vs Haupt- and Werkschule.

The above are some examples of the type of possible analyses that can be performed with respect to the meta data and provides a valuable resource for researchers and perhaps teachers. Depending on which spelling errors are of interest, the user interface provides a template of categories to select from as shown in Figure 9 in the Appendix.

6 Accessibility of Algorithms

The website is available in English and German via http://ktc.dh-karlsruhe.de/wise.php. The users are able to either explore the Karlsruhe Database or upload their own file in order to have the text annotated with spelling errors.

6.1 Data Exploration

As presented in Figure 8, the Karlsruhe Database exploration can be performed by filtering the desired texts according to meta-data, returning the errors committed by children of specific grades, kind of schools, ages, etc. Then the user can choose the types of spelling errors he wants to analyse (Figure 9). The result will be a tabulated file (csv) providing all the (non-unique) word pairs matching the filtering criterion, with basis and error values for each of the selected categories.

6.2 Data Input

If the user wants to perform an analysis on his own data, he can choose the "Upload your own file" option at the top of the web-page. The uploaded file must be a text file with the following format:

The first lines, preceded by #, are used to declare the meta-data. These metadata are:
#Grade:number
#Gender:F/M
#Age:number
#L1:languageSpokenAtHome[, anotherLanguageSpokenAtHome]*
#Schooltype:KindOfSchool (i.e., Grunschule, RealSchule, Hauptschule, Werkrealschule, ...)
#Date:dd/mm/yyyy (date of the data collection)
#Misc:free text (use this field for additional information, e.g. dyslexic child can be indicated here as LRS)
The rest of the file is used to provide the text to analyse, one sentence per line; the format is:
achieved (child) sentence 1
target (corrected) sentence 1
Am Tag danach ging er wieder_in die Schule.
Am Tag danach ging er wieder_in die Schule.
Die Kinder aus seinen{G} Klasse beo§bachten.
Die Kinder aus seiner{G} Klasse beo§bachten
Sie haben noch [§weiter] lesen geübt.
Sie haben noch [§weiter] lesen geübt.
Peter{N} war traurich.
Peter{N} war traurig.
Es gibt Hightek{F}=comuter{F}.
Es gibt Hightech{F}=Computer{F}.

This text can be annotated (grammar errors, foreign words, ...) according to the annotation scheme presented in Table 3 [5]. Words can be annotated with qualifications such as grammatical error ("gebte{G}" instead of "gab" which is clearly not a spelling error). At the sentence level substitutions can be annotated such as [wo der] (using "wo" instead of "der"). In the latter case the student's word will still be analyzed. Text is accompanied by meta data that will support growing a larger indexed database of children texts.

6.3 Data Output

The output of the above input is a tabulated .csv file with Tab as field separator, listing both Basis and Error for each of the corresponding selected error categories.

The following statistics are computed on the whole text and returned with the tagged file:

[5] Numbers are transcribed as they are; words like "Leeeooooooooooooonnn" also should not be corrected.

Letter- and Word-Level Annotations:	
*	unreadable letter
a_b	a and b should have been written separately
a§b	a and b should have been joined
a=b	missing hyphen
a∼b	wrongly placed hyphen
a−−b	denotes split of word at end of line (not hyphen)
a{n}	n repetitions of word a
a{F}	Foreign word defined by non-German graphemes, foreign grapheme-phoneme correspondence
a{I}	incomplete word
a{G}	grammatical errors not to be analyzed for spelling
a{A}	abbreviations such as Etc.{A}
a{N}	Names, not analysed with the spell tagger
Sentence Level Annotations	
[§ fW]	an unknown deletion
[§ b]	a known deletion b
[a §]	an insertion a
[a b]	substitution of a for b a is corrected on target side Achieved: [seinne ihre] Target: [seine ihre]
[a b_c]	best guess of word boundary
[a_b c]	kanicht = ka[n nn_n]icht
[a *]	some combinations of letters make up word a the real word can not be identified.
a can include conventions from word-level annotations For example: [rtchen**gdsdfg *] [rtchen**gdsdfg *] or [a{G} b]	

Table 3: Conventions for annotation of transcriptions as relevant to automatic spelling annotation.

Standard Method:

Each selected category provides the sum of $Basis$ and $Error$ found in the text and their ratio ($Quotient$) of actual errors: total number of $Errors$ divided by total number of $Basis$.

However, these raw statistics may not be appropriated in all cases. For example, consider that a child always misspells a specific word but does not generalize this to other words with similar patterns. In this case, it might be interesting to count pattern occurrences and not word occurrences. To express other ratios, two additional normalization methods have been added to the output.

Achieved-Target Pair Normalization:

$Basis$ counts each occurrence of the same pair (target;achieved) of words. This way, the same exact error, on the same word, repeated several times counts only once towards the final sum and quotient.

Target-Word Normalization:

Each target word counts towards the $Basis$ exactly once, regardless of how many different spelling errors are committed. Errors on this particular target word are then kept as a ratio (For example, in Table 4 30% (.3) of the time, the word "Gott" is misspelled as "Got"). All the occurrences of the same target word sum to 1. In other words, each occurrence of this word counts as $1/nbOccTargetWord$. With this measure, an isolated error on a word which is correctly written all the other times (could be a typo for instance) has a lower impact compared to the other counting methods. Conversely, if a word was spelled correctly once (by chance) out of ten times this should equally have a low impact (keeping the ration at .9).

Table 4 compares these different measures computed for the V_KV error category. The example text contains 10 occurrences of the word "Affe" (*monkey*), misspelled as "Afe", which is a V_KV error. This text also contains 2 correctly spelled occurrences of "stellen" (*place*) and 1 occurrence of "kann" (*I can*) both falling under KV rules. Normalizing differently will modulate the effect of the errors in "Affe". Depending on the use case, all the errors should be counted under the standard (**std**) normalization scheme. This is the case for dictation. Assuming the child principally knows the V_KV pattern, which seems to be the case when looking at the table. Unfortunately, the one word the child is not able to generalize to is the most frequent one. Pair normalization takes this into account by looking at pairs of mistakes, deemphasizing re-occurrence of particular mistakes. Supposing however, that it is important to report on the ratio of types of errors committed on a pair pattern that occurs several times. In this case, the third normalization takes into account the fraction of errors committed. This is shown in the table with the example of the word "Gott". In this case, the child is unsure about the orthography of "Gott" (*god*) and writes it correctly 4 times and incorrectly twice. In the pairwise normalization the $Error$ would be reported as 50/50, counting each pair only once, losing the frequency of each pair. In the target-word normalization the ratio would be presented accurately.

Comparing the three cases, 10 mispellings of "Affe" have a large impact on the standard ratio, with an error rate of 63% for V_KV (10 of the 12 V_KV errors are due to this word). The pair normalized ratio tells us that 40% of the word pairs raise a V_KV error: (Affe;Afe) counts as $Base = Error = 1$ such as the 2 occurrences of "Got" instead of "Gott". "Gott" correctly written

T	A	occ	std		pnorm		tnorm	
			B	E	B	E	B	E
Affe	Afe	10	10	10	1	1	1	1
kann	kann	1	1	0	1	0	1	0
Gott	Gott	4	4	0	1	0	0.7	0
Gott	Got	2	2	2	1	1	0.3	0.3
stellen	stellen	2	2	0	1	0	1	0
	Sum	19	19	12	5	2	4	1.3
	Quot			0.6		0.4		0.3

Table 4: Example of different computations for SUMME and QUOTIENT in case of KV error analysis: standard (std), pair-normalized (pnorm) and targetword-normalized (tnorm). B: Base E:Error T:target word, A: achieved word, occ: number of occurrences in the text.

also counts as $Base = 1$ but $Error = 0$. Target normalized ratio tells us that 33% of target words are wrong: (Affe;Afe) counts as 1, so $Basis = Error = 1$. "Gott" is misspelled 2 times out of 6, so the error ratio for this word is 0.33. Compared to the previous measure, this one takes into consideration the fact that "Gott" has been correctly written 66% of the times, whereas with pair normalized ratio, we considered that there were two different spellings and one was correct, leading to 50% of correct spellings. An example of such an output and the three ways of data normalization is depicted in Figure 10.

7 Conclusions

In this paper we evaluated a tool for automatic annotation of spelling errors on a publicly available database. We introduced a website that allows researchers to explore the tagged corpus or upload new data in order to have it annotated and joined to the data collection effort. This work is an important contribution to the research about spelling acquisition.

Acknowledgments

The work leading to these results was in part funded by a research grant from the German Research Foundation (BE 5158/1-1).

References

Kay Berkling, Johanna Fay, and Sebastian Stüker. 2011. Speech Technology-based Framework for Quantitative Analysis of German Spelling Errors in Freely Composed Children's Texts. In *SLaTE*, pages 65–68.

Kay Berkling, Johanna Fay, Masood Ghayoomi, Katrin Hein, Ludwig Linhuber, and Sebastian Stüker. 2014. A Database of Freely Written Texts of German School Students for the Purpose of Automatic Spelling Error Classification. In *Language Resources and Evaluation Conference LREC 2014*, pages 1212–1217, Reykjavik and Iceland.

Becky Bloom and Pascal Biet. 2008. *Der kultivierte Wolf*. Lappan Verlag, Oldenburg. Andrea Grotelüsche, Translator.

Anthony Browne. 1998. *Stimmen im Park*. Lappan, Oldenburg. Peter Baumann, Translator.

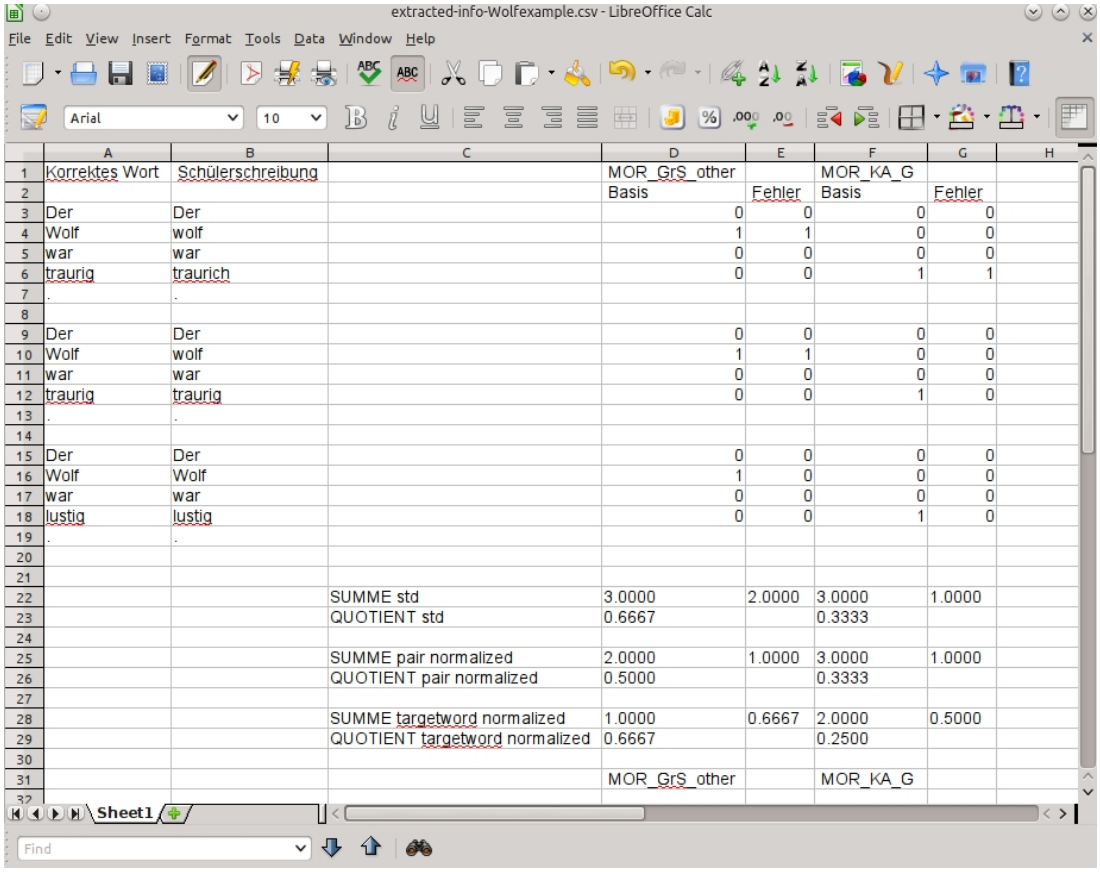
Johanna Fay. 2010. *Die Entwicklung der Rechtschreibkompetenz beim Textschreiben. Eine empirische Untersuchung in Klassen 1 bis 4*. Peter Lang, Frankfurt.

Rémi Lavalley, Kay Berkling, and Sebastian Stüker. 2015. Preparing children's writing database for automated processing. In *Workshop on L1 Teaching, Learning and Technology (L1TLT)*, Leipzig, Germany, September.

Uwe Reichel. 2012. PermA and Balloon: Tools for string alignment and text processing. In *Proc. Interspeech*, pages 1874–1877, Portland and Oregon.

Werner Scholze-Stubenrecht. 2004. *Duden - Die deutsche Rechtschreibung: Auf der Grundlage der neuen amtlichen Rechtschreibregeln*, volume 1 of *Der Duden in zwölf Bänden*. Dudenverl, Mannheim [u.a.], 23 edition.

Marc Schröder and Jürgen Trouvain. 2003. The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching. In *International Journal of Speech Technology*, pages 365–377.



	A	B	C	D	E	F	G	H
1	Korrektes Wort	Schülerschreibung		MOR_GrS_other		MOR_KA_G		
2				Basis	Fehler	Basis	Fehler	
3	Der	Der		0	0	0	0	
4	Wolf	wolf		1	1	0	0	
5	war	war		0	0	0	0	
6	traurig	traurich		0	0	1	1	
7	.	.						
8								
9	Der	Der		0	0	0	0	
10	Wolf	wolf		1	1	0	0	
11	war	war		0	0	0	0	
12	traurig	traurig		0	0	1	0	
13	.	.						
14								
15	Der	Der		0	0	0	0	
16	Wolf	Wolf		1	0	0	0	
17	war	war		0	0	0	0	
18	lustig	lustig		0	0	1	0	
19	.	.						
20								
21								
22			SUMME std	3.0000	2.0000	3.0000	1.0000	
23			QUOTIENT std	0.6667		0.3333		
24								
25			SUMME pair normalized	2.0000	1.0000	3.0000	1.0000	
26			QUOTIENT pair normalized	0.5000		0.3333		
27								
28			SUMME targetword normalized	1.0000	0.6667	2.0000	0.5000	
29			QUOTIENT targetword normalized	0.6667		0.2500		
30								
31				MOR_GrS_other		MOR_KA_G		
32								

Figure 10: Example of tool output.

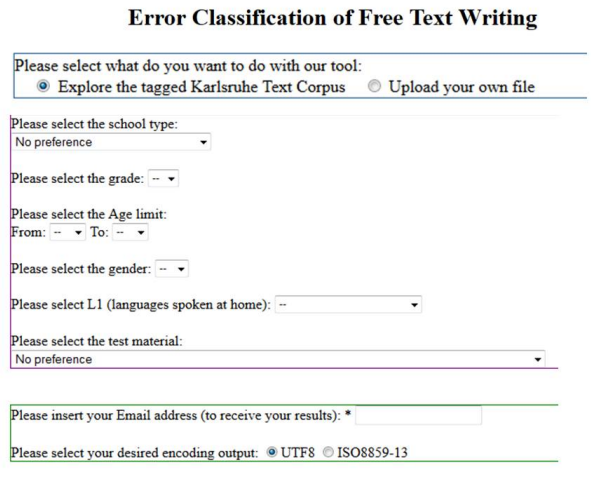


Figure 8: Meta data selection on Web interface.

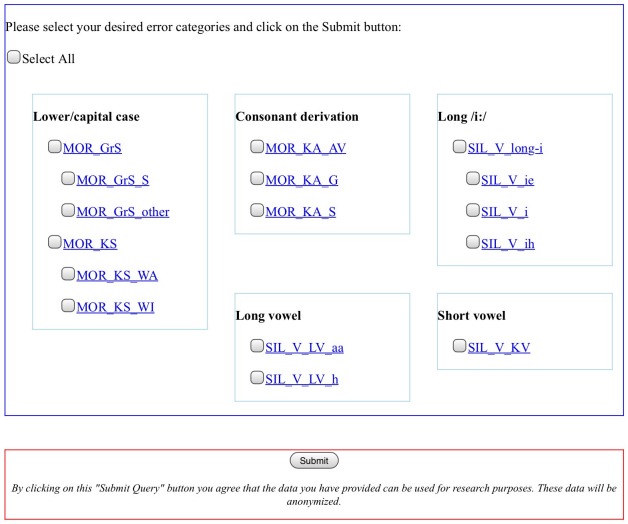


Figure 9: Selection of error categories to mark.

A case-study of automatic participant labeling

Alexander Kampmann
Saarbrücken Graduate School
of Computer Science
Saarland University
Saarbrücken, Germany
kampmann@st.cs.uni-saarland.de

Stefan Thater and **Manfred Pinkal**
Dept. of Computational Linguistics
Saarland University
Saarbrücken, Germany
stth@coli.uni-saarland.de
pinkal@coli.uni-saarland.de

Abstract

Knowlegde about stereotypical activities like *visiting a restaurant* or *checking in at the airport* is an important component to model text-understanding. We report on a case study of automatically relating texts to scripts representing such stereotypical knowledge. We focus on the subtask of mapping noun phrases in a text to participants in the script. We analyse the effect of various similarity measures and show that substantial positive results can be achieved on this complex task, indicating that the general problem is principally solvable.

1 Introduction

Imagine how you tell a friend about a restaurant visit. You will talk about the people you met there, describe the place and maybe compliment the food. But it is very unlikely that you will tell how the waiter showed you to a table, how you scanned the menu or how the food was brought. Those events are typical for a restaurant. You can assume that the listener knows that they happened, even if you do not mention them.

This kind of knowledge can be captured as a script, a sequence of events which typically constitute a restaurant visit or another common activity (Schank and Abelson, 1977). Scripts also contain information about the participants which take part in an activity.

Script knowledge is implicit in most text sources. This is problematic for text understanding systems. If a text understanding system had access to script knowledge, it could infer the implicit events in the same way a human does. Also the implicit nature of script knowledge means that it is hard to get.

There are approaches to collect script knowledge manually (Mueller, 1998), however, this is too much manual work to be practical. Chambers and Jurafsky (2008) learn narrative schemas from large text corpora. Narrative schemas are similar to scripts, but they are not linked to specific scenarios.

We based our work on the script representations by Regneri et al. (2010) who use crowd-sourcing techniques to collect script knowledge. This approach is scalable and does not suffer from noise, as Chambers and Jurafsky's approach does.

For the script of "Visiting a restaurant", the waiter, the guest and the food are typical participants. Regneri et al. (2011) also processed the script representations to identify participants.

We used texts from the "Dinners from Hell" corpus. This corpus is a collection of texts that were submitted to a website[1] which collects stories about bad experiences with restaurants. The texts feature harsh waitstaff or over-expensive restaurants, which means the Restaurant script is the main topic. Thereby a lot of the script events are mentioned explicitly. The corpus also contains some texts which incorporate deviations from the script. As an example, one text explains how a large chandelier falls on the customer's table. This event is not included in the Restaurant script.

In this paper, we present a system that labels noun phrases with the script participants they refer to. As a case study, we evaluated our system on the "Dinners from hell" corpus. We applied participant knowledge from the Restaurant script that was devised by Regneri et al. (2011). 1 shows a sentence with labels automatically assigned by our system. The participants "customer", "food", "drink", "table" and "location" have been identified. We can also see a mistake in the automated labeling. The tickets are labeled as "bill". This happens, because the Restaurant script is about a restaurant were the customers give a spoken order, rather than a written one.

Our contributions are the following:

[1] www.dinnersfromhell.com

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 97–105, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

We (customer) each ordered entrees (food) and sodas (drink), placed the tickets (bill) on our table (table), and went to the buffet line (location) for some salads (food).

Figure 1: A sentence from the development set with generated annotations.

1. We describe an automated participant labeler, which tries to find the participant that is most similar for each noun phrase. We compare several variants of a model which can be used to calculate similarities between participant mentions in texts and script occurrences of a participant.

2. We evaluate our similarity functions on the "Dinners from hell" text corpus, together with the Restaurant script that was provided by Regneri et al. (2011).

3. Our analysis also provides an assessment of the completeness and usefulness of available script information.

The paper is structured as follows: 2 gives an overview of Regneri et al. (2010) and Regneri et al. (2011), which provided the script representations we used. In 3, we describe the annotation process. 4 describes our participant labeling. In 5, we discuss our results. In 6, we conclude the paper.

2 Script data

We used the script representations from Regneri et al. (2010).

In this work, several descriptions of typical instances of a variety of scripts were collected on Amazon's Mechanical Turk[2]. The Mechanical Turk provides access to "Human Computing Resources". This means that human annotators are asked to solve small tasks and are payed small amounts of money for each task.

Regneri et al. (2010) asked annotators on MTurk to provide descriptions of the typical event sequence for a scenario in a telegram-style. This leads to several "Event Sequence Descriptions" (ESDs). In each ESD, there is one sentence per event, however, different annotators often provided different descriptions for the same event. Also the

[2]http://mturk.com

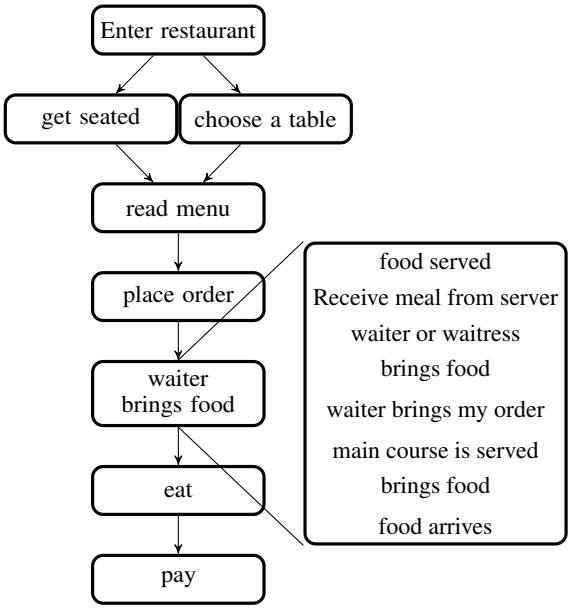


Figure 2: Simplified representation of the restaurant script, with a full view of the "waiter brings food" event.

ESDs from different annotators differ in granularity, and thereby in length. Regneri et al. (2011) aligned event descriptions by a version of multiple sequence alignment which is sensitive to semantic similarity. They obtained a graph representation of the script, where each node contains a set of similar event descriptions, and the edges denote their order in the ESDs. 2 shows a simplified version of the Restaurant script. Each node in the script graph contains all the sentences the annotators provided for this event. In the figure, we labeled each node with a single description. We show the full list for the "waiter brings food" event as an example.

Each noun phrase in an ESD refers to a participant of the script, but different noun phrases may refer to the same participant. In the example, the "waiter" participant is realized as "server" in the second sentence, but as "waiter or waitress" in the third. In follow-up work Regneri et al. (2011) used an Integer Linear Program (ILP) to group the noun phrases to "participant description sets" (PDS). The participant description set for the waiter is $\{\text{waiter}, \text{waitress}, \text{server}\}$

For our experiment, we did not use the ILP approach, but specified a gold standard for the Restaurant script manually. This gold standard is based on noun phrases from Regneri's work, but the grouping has been done manually. We decided to do so, because the ILP solution was to noisy to get good results.

3 Text data: Dinners from Hell

The "Dinners from Hell" corpus has been extracted from the "Dinners from Hell" website. The main topic of the texts in the corpus are restaurant visits, so the script knowledge is explicit in most of them. This makes the texts a good subject for our work.

The corpus contains some texts which describe (displeasing) deviations from the restaurant script. This is a challenge for script-knowledge based systems, as they may not be able to cope with atypical scenarios. On the other hand, there are several texts which have all the typical script events. As an example, one of the stories is about an overly expensive restaurant, another one features a harsh waiter.

We took existing annotations (Rudinger et al., 2015) as a starting point. Three annotators marked independently which verbs they consider relevant for a given script. In the sentence

> I was near Harvard Square and decided to have lunch at a small Chinese restaurant.

the verb "was" would not be marked, because being in a special place is not part of the restaurant script. On the other hand, "have lunch" is typical for a restaurant, so it is marked as script-relevant. The word "decide" was not marked as script-relevant by two annotators, while the third one marked it. In such cases, the solution most annotators used was considered as correct. So "decide" is not script-relevant here.

We parsed the texts with CoreNLP (Manning et al., 2014) and considered direct dependents of event verbs as candidates for participant annotation.

We extended the annotation with participant labels for those candidates.

In the ideal case, the labels in the manual annotation would be the participants that are contained in the script representation. However, it turned out that the script data is incomplete. As an example, none of the ESDs used to derive the script representation contained the option of taking left-overs in a to-go box, which happens regularly in the texts. To account for this problem, we extended the tag set for the annotations. 1 lists the tags that were used in the manual annotation in the left column. The right column contains which of the participant description sets from the Restaurant script we considered equivalent to a gold label.

Gold Annotation	Accepted labels
amount	
coupon	
reservation	
to-go box	
utensils	
customer	customer
cashier	waiter
management	waiter
waiter	waiter
restaurant_institution	waiter
cook	cook
condiments	food
drink	drink
food	food
food_or_drink	food, drink
order	drink, food
credit card slip	credit card slip
bill	bill
tip	tip
kitchen	kitchen, restaurant_location
location	restaurant_location
restaurant_location	restaurant_location
menu	menu
payment method	payment method
table	table

Table 1: Mapping of gold tags to automated labels. Empty fields mean this annotation can not be matched by the participant labeler.

In some cases, the parse trees contained errors. The annotator assigned the '- - -' label to incorrectly recognized words. Words which were direct dependents of an event verb, but missed by the parser, were tagged as well.

During the annotation, it turned out to be difficult to assign unique labels to all participants. As an example, a person who brings food to the table usually is a waiter. However, it may happen that the owner or manager brings food. In this case the owner works as a waiter in his own restaurant. The annotator assigned the waiter label if the person served as a waiter, however, if it was clear from the story that this is the owner, the owner label was used, even if the person took over a waiter's responsibilities.

For this exploratory study, we rely on a single annotator. The manual annotations are used as a gold standard, the evaluation compares the results

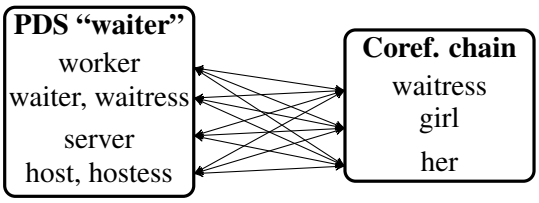


Figure 3: Similarity values between a participant description set (PDS) and a coreference chain.

of our participant labeler to them.

4 Models for Participant Labeling

The automated participant labeler receives the verb annotations in the text and the participant description sets from the script as input. The verb annotations contain the information which verbs are script-relevant. The labeler uses this information and the CoreNLP parser to identify direct dependents of script-relevant verbs as participant candidates. Those will be labeled later on, which is the same thing the annotator did conceptually.

The labeler uses CoreNLP to get coreference chains. We assume that two words in the same coreference chain denote the same participant, so the participant labeler assigns a role label for each coreference chain. In order to do so, it calculates a similarity score between the head nouns of the noun phrases in a coreference chain and the words in a PDS. 3 shows a PDS and a coreference chain.

We obtained several similarity values between each coreference chain and PDS. In 3, three words in the coreference chain and four words in the PDS lead to 12 similarity values. The edges illustrate the twelve similarities we calculate. For the labeling, we need a single similarity value for each participant. In the "max" approach, we calculated the maximum of all similarity values between a coreference chain and a PDS. In the "mean" approach, we used the arithmetic mean of all the values as the similarity. In both cases, all words in a coreference chain are labeled with the participant which reaches the highest similarity value.

4.1 WordNet-based similarity functions

WordNet (Miller, 1995) is a database which contains english words and relations between them. Most notable, WordNet relates hypernyms to their hyponyms. We expect that the script representation contains general terms, while the text rather use the more specific words. This motivates that we use the **hyponym** similarity metric, which comes as a built-in in NLTK (Bird, 2006). We also included **Lin's WordNet-based similarity** (Lin, 1998) and the **Wu-Palmer similarity** (Wu and Palmer, 1994).

All WordNet-based similarity functions calculate a similarity score between two WordNet synsets. SynSets group words with the same meaning together. Words can be in more than one synset, if they have several meanings. As an example, "card" may, among others, refer to the "menu" as well as to a "credit card", so the word is in several synsets.

For our similarity functions, we took all synsets that contain one of the words in the participant and compared them to all the synsets that contain a participant candidate in the respective coreference chain. Again this yields several values per coreference chain. We used an weighted arithmetic mean or simply the maximum to get a single value. The weights in the arithmetic mean are the occurrence counts of words from the participant in the respective synsets.

4.2 Verb-occurrence similarity functions

Some participants occur with some verbs more frequently than others. As an example, "waiters" "bring" food, so the subject of "bring" is most likely a waiter. Thereby we designed a similarity function which uses the verb a participant candidate depends on.

For this similarity function, we worked with the entire coreference chain and the entire PDS directly, rather than comparing individual words. Each participant candidate is a direct dependent of an event-relevant verb. We counted how often which verb is used with a participant candidate in the coreference chain. This gives us a vector of verb occurrence counts.

We also counted the verbs which were used in the PDS together with the noun phrases in the ESDs. This yields a second occurrence count vector. The cosine similarity between the vectors serves as similarity value for the verb similarity function.

4.3 Distributional similarity functions

We complement the knowledge-based WordNet similarity scores with similarity scores from a distributional semantic model. The key idea behind distributional models is that semantically similar words tend to occur in similar linguistic contexts in large text corpora. Distributional models represent these contexts as vectors and compute se-

mantic similarty by comparing these vectors in a high-dimensional vector space.

We use the model of Thater et al. (2011), who train a vector space model from a dependency parsed version of the English Gigaword corpus. We use the model in two ways: In the "uncontextualized" mode we simply compare the vector of the target word with the vector of the word representing the participant in the script; in the "contextualized" mode, we first contextualize the vector of the target word using the syntactic context in which it occurs before comparing it with the vector of the participant. The basic intuition here is that contextualizing a vector should improve the similarity scores for ambiguous target words ("look at *card*" vs. "pay with *card*"). It turns out, however, that the use of uncontextualized vectors leads to a better performance compared to using contextualized vectors (see Section 5).

In cases where the target word occurs in a coreference chain, we used both the sum as well as the mean of the pairwise similarity scores of the vectors of the words in the coreference chain and the vector for the participant.

5 Evaluation

Our approach is not a machine-learning approach. The similarity functions are not trained on our data. However, especially for the combinations of similarity functions, there was a lot of manual optimization needed. In order to avoid a subject bias here, we divided the "Dinners from Hell" into a test and a development set. None of the authors had a look at the test set data until the final evaluation started.

The development set contained 71 texts with 28707 words. The test set contains 72 texts with 28600 words. During the annotation, the annotator reported some problems. Our inspection of the problematic texts lead to updates in the annotation guidelines. All texts we saw in this process were taken to the development set. After that, we selected additional texts for the development set randomly.

We ran our participant labeler on the test set and calculated the precision and recall for a label p as given in 1 and 2.

$$\text{precision}(p) = \frac{|TP_p|}{|TP_p + FP_p|} \quad (1)$$

$$\text{recall}(p) = \frac{|TP_p|}{|TP_p + FN_p|} \quad (2)$$

Intuitively, precision is the fraction of correct labels among the assigned labels. Recall is the fraction of participants which are identified and labeled correctly by the participant labeler.

To give an impression of the performance of the participant labeler with respect to all labels, we use micro and macro averages of precision and recall as given in:

$$\text{macroprecision} = \frac{1}{|P|} \sum_{p \in P} \text{precision}(p) \quad (3)$$

$$\text{macrorecall} = \frac{1}{|P|} \sum_{p \in P} \text{recall}(p) \quad (4)$$

$$\text{microprecision} = \frac{\sum_{p \in P} |TP_p|}{\sum_{p \in P} |TP_p| + |FP_p|} \quad (5)$$

$$\text{microrecall} = \frac{\sum_{p \in P} |TP_p|}{\sum_{p \in P} |TP_p| + |FN_p|} \quad (6)$$

In 3 and 4 all participants have the same influence, no matter how often they occur. 5 and 6 are more relevant for our task, because they consider how often a participant occurs in total, so mistakes in participants which occur seldom have less effect than mistakes in more frequent participants.

The participant labeler and the annotator worked under different conditions. First of all, the annotator fixed parser errors on the fly. Our participant labeler is incapable of doing so. Also the annotator used a tag set that contains some participants that do not occur in the script representation. One example is the "to-go box". None of the initial ESDs contained the option to take left-overs home in a to-go box. 1 lists the tags that the annotator used in the left column, the right column lists the tags our participant labeler uses which we consider equivalent.

One of the main differences is in the "food" tags. The annotator assigned the "food" label for "food" and the "drink" label for drinks. If it was not obvious whether it was drink or food, as in the sentence "the waiter brought our order" the "food_or_drink" tag was used. The participant labeler never uses the "food_or_drink" tag, it always commits to one of the specific options.

This leads to a problem with respect to false negatives. If the annotator assigned the "food_or_drink" tag and the participant labeler decides for something wrong, is this counted as a false negative

Heuristic		Micro-			Macro-		
		Precision	Recall	F-Score	Precision	Recall	F-Score
Base lines							
perfect labeling		0.86	0.86	0.86	0.83	0.78	0.80
manual WordNet		0.02	0.02	0.02	0.00	0.07	0.00
string equality		0.22	0.21	0.21	0.46	0.42	0.44
random		0.06	0.05	0.05	0.05	0.05	0.05
most frequent participant		0.32	0.27	0.29	0.02	0.07	0.03
WordNet-based similarity functions							
Lin	mean	0.43	0.38	0.41	0.33	0.47	0.39
	max	0.36	0.37	0.37	0.28	0.42	0.34
Wu-Palmer	mean	0.40	0.35	0.37	0.29	0.47	0.36
	max	0.37	0.37	0.37	0.35	0.54	0.42
Hypernym	mean	0.51	0.49	0.50	0.38	0.53	0.45
	max	0.59	0.59	0.59	0.42	0.55	0.47
squared lin	mean	0.52	0.49	0.50	0.35	0.52	0.42
	max	0.35	0.34	0.34	0.28	0.41	0.33
squared Wu-Palmer	mean	0.46	0.42	0.44	0.32	0.51	0.39
	max	0.37	0.37	0.37	0.35	0.54	0.42
Verb similarity functions							
verb		0.10	0.08	0.09	0.07	0.16	0.10
Distributional similarity functions							
VSM, sp, max		0.36	0.37	0.36	0.37	0.57	0.45
VSM, cosine, max		0.35	0.36	0.35	0.34	0.53	0.42
VSM, cosine, mean		0.35	0.36	0.36	0.32	0.53	0.40
cosine, max		0.28	0.27	0.28	0.27	0.41	0.32
sp, max		0.29	0.29	0.29	0.28	0.44	0.34
cosine, mean		0.28	0.28	0.28	0.27	0.41	0.33
Combinations of similarity functions							
Wu-Palmer + verb, max		0.33	0.32	0.33	0.25	0.41	0.31
sq. Wu-Palmer + verb, max		0.35	0.35	0.35	0.26	0.45	0.33
Wu-Palmer + Verb, mean		0.38	0.32	0.35	0.21	0.37	0.27
Lin + Verb, mean		0.22	0.18	0.20	0.17	0.32	0.22
mean of Wu-Palmer + max of sq. Lin		0.57	0.56	0.56	0.31	0.56	0.40
mean Wu-Palmer + cosine, max		0.55	0.53	0.54	0.36	0.57	0.44
hill-climbing		0.60	0.60	0.60	0.35	0.59	0.44
sum of all		0.38	0.39	0.38	0.35	0.57	0.44

Table 2: Results of our participant labeling. VSM means the vector space model with no context information. sp denotes the scalar product

with respect for the "food" tag or the "drink" tag? We double counted here, which leads to a higher number of false negatives.

The participant labeler can not automatically correct parsing errors, thus it can not assign the same labels as in the gold standard in all cases. Also the double counting effect obscures the evaluation metrics. To account for those effects, we added a simulated, perfect labeling, which assigns the gold annotation in cases the participant labeler can handle and a wrong tag otherwise. This perfect labeling serves as an upper bound for the performance of our automated labeling.

We added three base lines. String equality counts how many words in a coreference chain are equal to words in the participant and normalizes by the number of words in the chain. The random assignment assigns a participant randomly, most frequent participant assigns "customer", which is the most frequent participant in the "Dinners from hell" corpus, to all participant candidates.

5.1 WordNet-based similarity functions

As a base line for WordNet-based approaches we manually selected a WordNet-synset per participant and labeled each participant candidate with the closest participant according to Lin's similarity measure. The results are given as 'manual' in 2. This approach is substantially outperformed by the automated synset selection. We selected just one synset per participant, but for some participants, there is no single, descriptive synset, which we think is the problem about the manual selection.

For the WordNet-based similarity functions, we get several values per function, because there are several synsets per participant. We used either a weighted average or the maximum to get a single value, so there are two rows per WordNet-based similarity in 2.

For an arithmetic mean, small differences between values can be lost due to the mean calculations. On the development set, we observed that this effect happened for several cases in Lin and Wu-Palmer similarity. To counteract this effect, we also tried squared versions of both similarity measures. Squaring a similarity score makes small values smaller, but has little effect on values close to 1. Thereby squaring the similarity gives larger differences between large values and close-by, but smaller values.

With our WordNet-based approaches, we achieve a Micro-Precision of up to 59%, the Macro Precision reaches 42%. Squaring has a positive effect on the mean similarity and almost no effect on the maximum similarity, which is expected, as squaring does not change which of the numbers is larger.

5.2 Verb occurrence similarity function

The verb occurrence similarity function reaches a Micro-Precision of 10% and a Macro-Precision of 7%. This is substantially worse than the wordNet-based approaches. However, words like "he" or "she" do not occur in WordNet, so the WordNet-based approaches can not label them, unless there is some word that offers more information in the same coref-chain. With the verb similarity function, our participant labeler managed to label some mentions of pronouns correctly.

5.3 Distributional similarity functions

For the distributional similarity functions, we evaluated how context information influenced the decisions and whether it made a difference if we used the maximum or the average to get a single value for a coreference chain. Also we compared the cosine and the scalar product as vector similarity.

The best performing similarity function in this category uses no context information, the scalar product and the maximum. I reaches a Micro-Precision of 36%. All in all, the scalar product seems to be better than the cosine similarity, which confirms earlier results with other distributional approaches.

All in all, the distributional approaches do not outperform the WordNet-based approaches.

5.4 Combining similarity functions

We observed that all of the similarity functions perform well in different situations, so it is reasonable to assume that combinations may perform even better. As a test, we averaged Wu-Palmer and Lin similarity. The result is the second best heuristic in our evaluation, which encouraged us to try several combinations.

As a base line, we averaged all similarity functions, except for the scalar products. The reason to exclude the scalar products is that all other similarities are between 0 and 1, which the scalar product is not. This means the scalar product would dominate all other heuristics. The results for the sum are included in 2 as "sum of all."

Additionally, we experimented with several combinations of individual similarity scores. First, we designed some combinations by hand. Our intuition was that WordNet-based similarities and the verb similarity should play together well, because they provide information for different parts of speech. Second, we combined functions with a high dissimilarity between the confusion matrices, which leads to a selection of functions which make mistakes for different participants. Third, we applied a simulated hill-climbing on the development set to find a good combination. This heuristic is basically a weighted sum of other heuristics.

The results of the combinations are somewhat discouraging. Apparently, the WordNet-based heuristics are overrated by high scores from the verb heuristics. So a well-performing heuristic can not reliably overvote a bad performing one.

The most dissimilar confusion matrices can be found for the distributional model with context information and the mean of Wu-Palmer similarity. The combination outperforms both partners, and almost reaches the best performing individual score. So the idea of combining heuristics with different errors seems to work.

Hill climbing leads to the best performing combined similiarity score in terms of Micro F-Score. However, the differenece to the best performing individual score (Hypernym, max) is small.

6 Conclusion

In this paper, we reported on a case study on automatically mapping noun phrases in texts to participants in scripts, and showed that substantial positive results can be achieved on this complex task. Our automatic participant labelling system uses similarity scores between pairs of words to identify the most apropriate pariticipant for a given noun phrase. We investigate the effect of various similarity measures. The best individual measure achieves a (Micro-) F-score of 0.59, compared to the baseline of 0.29 (most frequent participant) and the upper bound of 0.86. The system performance can be further improved to 0.6 by combing different similarity measures using hill climbing.

The WordNet-based similarity functions perform reasonably well on this task. This came as a surprise to us, as we initially believed it would not contain a lot of relevant words.

We were also surprised that distributional similarity functions can not outperform the WordNet-based functions. We believe that the problem is that our words are all from the same domain (restaurant), and thereby have high (distributional) similarity anyways.

Our results hold for the "Dinners from hell" corpus. It is not clear whether our similarity functions can be applied in domains other than restaurant visits. More texts and different scripts would be necessary in order to evaluate this.

Also we do not believe that we found the best possible combination of similarity functions yet. While hill climbing yielded a good similarity function, it outperforms our initial, uninformed guess only slightly. It might be possible to come up with a better way to combine similarity scores.

The WordNet-based approaches work best on nouns. Distributional approaches, on the other hand, should be able to handle all word classes. Thereby it is a straight-forward idea to implement a similarity function which relies on distributional similarity for everything except nouns, which can be handled by WordNet. We did not implement this approach so far.

However, we got the impression that sparse script data is a more severe problem. Some participants which occur in the texts do not occur in the script. Further research about how to collect script data is required.

Acknowledgments

We thank Alessandra Zarcone and our student assistant Tatiana Anikina for providing the annotation used in our study.

References

Steven Bird. 2006. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, COLING-ACL '06, pages 69–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 789–797.

Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November.

Erik T Mueller. 1998. *Natural language processing with ThoughtTreasure*.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. 2010. Learning script knowledge with web experiments. In *Proceedings of ACL 2010*, Uppsala, Sweden, July. Association for Computational Linguistics.

Michaela Regneri, Alexander Koller, Josef Ruppenhofer, and Manfred Pinkal. 2011. Learning Script Participants from Unlabeled Data. In *Proceedings of RANLP 2011*, Hissar, Bulgaria.

Rachel Rudinger, Vera Demberg, Ashutosh Modi, Benjamin Van Durme, and Manfred Pinkal. 2015. Learning to predict script events from domain-specific text. *Lexical and Computational Semantics (* SEM 2015)*, page 205.

Roger Schank and Robert Abelson. 1977. Scripts plans goals and understanding.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word Meaning in Context: A Simple and Effective Vector Model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1134–1143, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Zhibiao Wu and Martha Palmer. 1994. Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

A Hybrid Approach to Extract Temporal Signals from Narratives

Thomas Bögel **Jannik Strötgen** **Michael Gertz**
Institute of Computer Science, Heidelberg University, Germany
{thomas.boegel,stroetgen,gertz}@informatik.uni-heidelberg.de

Abstract

When processing literary narratives, standard temporal annotation specifications – typically developed for processing news-style documents – do not match the expectations of literary scholars. Thus, a different definition of *temporal signals* is required. In this paper, we define this concept from the narratological perspective and present our hybrid approach developed in the context of the heureCLÉA[1] project to extract temporal signals. Our evaluation demonstrates high quality extraction results, making the approach directly applicable to the literary domain.

1 Temporal Signals

The temporal markup language TimeML (Pustejovsky et al., 2005) was developed for temporally annotating text documents such as business news. Annotations include temporal expressions, temporal signals, events, and temporal relations. Besides four types of temporal expressions (date, time, duration, and set expressions) covered by TimeML's TIMEX3 tag, the SIGNAL tag is used to annotate expressions that might be helpful for temporal reasoning, e.g., temporal prepositions and conjunctions (Pustejovsky et al., 2005).

In general, narratologists make a much more fine-grained distinction between different kinds of temporal expressions (Lahn and Meister, 2013, §2.3) and include many more that are neglected by TimeML. Temporal expressions (including temporal signals) from TimeML represent a very small sub-set of the narratological definition, such as narratological time-points ("On July 22, 1848"). Many more expressions, however, are covered by the narratological definition, e.g., event-related time points (e.g., "while the bells were ringing…"). We thus define a *temporal signal* from the narratological perspective as *a phrase capturing temporal semantics excluding tense*.

[1] http://heureclea.de/

2 A Hybrid Approach

In contrast to TimeML's TIMEX3 and SIGNAL, the narratological definition of temporal signals implies that one is faced with an open vocabulary. The combination of an open vocabulary and the context of narrative texts with substantial variations in styles and textual content across different texts leads to the issue of data sparsity when trying to predict temporal signals. With a rule-based approach, instances in documents different from the set of documents used for deriving the rule set will most likely not be found. Thus, instead of applying a fully rule-based approach for their extraction – similar to the temporal tagger HeidelTime (Strötgen and Gertz, 2013) for TIMEX3 –, we use a hybrid extraction approach.

After extending HeidelTime to radically increase the recall for extracting narratological temporal signals, we use machine learning to remove incorrectly extracted expressions and to achieve a balanced relation between precision and recall. This technique combines the advantages of both approaches: (i) generalized heuristics yield high recall without the need of copious amounts of training data; (ii) the machine learning component is perfectly suited for improving precision by looking at universal contexts that are preferably general and can be applied across different texts.

Extension of HeidelTime. Instead of applying HeidelTime to extract TIMEX3 annotations only, we extend its rule base by adding general rules to capture a broad range of temporal signals. Note that due to the open vocabulary used to formulate temporal signals, these rules are very general and aim at a high recall ignoring precision issues. The output of HeidelTime's extended version is thus a set of candidate signals $C = \{c_1, \ldots c_n\}$.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 106–107, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

	# doc.	token	TIMEX3	signals
train	21	79,431	315	3,144
test	4	23,218	11	215

Table 1: Statistics of the training and test set.

Machine Learning. To boost precision, we train a classifier that judges the output of the heuristic system. For each candidate c, a binary classifier determines whether c represents a signal, resulting in a set of final predictions for temporal signals $S = \{s_1, \ldots, s_m\}$, where $S \subseteq C$. The classifier is trained on manual annotations of the training set.

3 Data Sets and Evaluation

We use the corpus of the heureCLÉA project, consisting of 25 narrative texts in German from various authors of the 20th century that comprise less than ten pages (Bögel et al., 2014). To train and evaluate our approach, we split the data into distinct training and test sets. The data set statistics in Table 1 show that narratological temporal signals are much more prevalent in the data than TIMEX3 expressions.

Manual annotation. To extend the rule set and train the classifier, we performed an error-driven evaluation. After a first run of HeidelTime, an expert in narratology manually annotated erroneous, missing, and correct candidates in all training documents. The test set was annotated separately and without prior knowledge about system predictions.

Feature set. Overall, we used 17 features to train the classifier. The feature set comprises information about the length and part-of-speech tags of the candidate, structural properties like the occurrence of the candidate in complex sentence structures, as well as string-based features characterising the subject and verb of the sentence containing the candidate and the presence of temporal adverbs within the sentence. Finally, we investigate changes of verb tense in the surrounding context.

Evaluation. To demonstrate that the narratological perspective on temporal signals differs from TimeML's approach, we first evaluate the performance of HeidelTime on the test set. Then, we show the effects of tackling the problem with a heuristic system by extending and generalizing HeidelTime's rule set. Finally, we report the results of our hybrid approach.

The evaluation results in Table 2 confirm the assumption that a temporal tagger in isolation is not sufficient to extract narratological temporal signals, yielding devastating recall. Extending the rule set significantly increases recall but leads to many false positives – as expected. Finally, our hybrid approach that combines heuristics and machine learning achieves the best and most balanced result with a high precision of 74.7% and 83.5% for strict and loose evaluation metrics, respectively. While recall is slightly decreased due to the nature of our setting, it is still solid, especially considering the fundamental textual differences between training and test set. The large drop in recall of the hybrid system for the loose setup can be explained by the fact that we treat overlapping temporal signals as candidates that should be filtered out to boost precision.

		prec.	rec.	F_1
HeidelTime	*strict*	23.1	1.4	2.6
(TIMEX3 only)	*loose*	84.6	5.1	9.7
Heuristics	*strict*	33.5	78.1	46.9
(ext. HeidelTime)	*loose*	38.5	89.8	53.8
Hybrid	*strict*	74.7	71.7	73.2
(Heuristics + ML)	*loose*	83.5	77.6	80.4

Table 2: Evaluation results on the test set.

4 Future Work

As mentioned above, many recall errors are due to the handling of overlaps as errors when training the classifier. Thus, the next step is to add overlapping candidates as a separate classification outcome to increase the recall of the system. In addition, we are working on a more fine-grained classification of different types of temporal signals by implementing a two-step classification.

References

Thomas Bögel, Jannik Strötgen, and Michael Gertz. 2014. Computational narratology: Extracting tense clusters from narrative texts. In *LREC*, pages 950–955.

Silke Lahn and Jan Christoph Meister. 2013. *Einführung in die Erzähltextanalyse*. J.B. Metzler.

James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Sauri. 2005. Temporal and Event Information in Natural Language Text. *Language Resources and Evaluation*, 39(2–3):123–164.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298.

A Resource for Natural Language Processing of Swiss German Dialects

Nora Hollenstein
Institute of Computational Linguistics
University of Zurich
hollenstein@ifi.uzh.ch

Noëmi Aepli
Institute of Computational Linguistics
University of Zurich
naepli@ifi.uzh.ch

Abstract

Since there are only a few resources for Swiss German dialects, we compiled a corpus of 115,000 tokens, manually annotated with PoS-tags. The goal is to provide a basic data set for developing NLP applications for Swiss German. We extended the original corpus and improved its annotation consistency. Furthermore, we trained dialect-specific PoS-tagging models and implemented a baseline system for dialect identification.

1 Introduction

Swiss German is a dialect continuum which includes dialects derived from Standard German. However, NLP tools for Standard German do not perform well on Swiss German as it differs from Standard German in terms of phonetics, vocabulary, morphology and syntax. Moreover, there is no official orthography standard. Nevertheless, Swiss German has been used more frequently in recent years not only in informal contexts but also in the media and by literary authors. Therefore, the need for dialect-specific resources as a foundation for Swiss German NLP becomes evident. In previous work, Scherrer and Rambow (2010a) reviewed the existing resources and applications that have been developed for Swiss German language processing.

The present work builds on an existing corpus for Swiss German dialects (Hollenstein and Aepli, 2014). *NOAH's Corpus* includes written texts from different genres: Wikipedia articles, news articles, the SWATCH annual report (2012), chapters from novels and web blogs. We extended the corpus through further manual annotation in order to provide a larger lexical resource, which is freely available for research purposes[1]. Moreover, we explored two applications of this corpus: dialect-specific PoS-tagging and dialect identification.

2 Annotation

We extended the corpus by adding additional texts of the genres mentioned above to reach 115,000 tokens. The manual annotation of PoS-tags was conducted by the authors (Swiss German native speakers) and followed the same guidelines from Hollenstein and Aepli (2014). The STTS (Schiller et al., 1999) was chosen as the basic tag set with the addition of a few new tags to cover phenomena not present in Standard German. Swiss German requires a tag *PTKINF* (infinitive particle) for sentences such as *"Am erschtä Tag simmer go (PTKINF) poschtä."*[2], which has no direct translation to Standard German. Furthermore, a "+"-sign is added to the tag of merged words, which appear due to the lack of an orthography standard.

2.1 Annotation Consistency

In order to provide a reliable resource for language technology, we place great importance on the manual annotation process. To ensure that the annotation is consistent throughout the whole corpus and to further specify the annotation guidelines, we applied a simplified version of the *variation n-gram method* (Dickinson and Meurers, 2003). This technique detects sequences of *n* tokens which occur multiple times in the corpus with varying annotation. The detected n-gram sequences were manually analyzed.

3 Dialect-specific PoS-Tagging

We trained the BTagger (Gesmundo and Samardžić, 2012) on the annotated data. Over the whole corpus a tagging accuracy of over 90% was reached (Hollenstein and Aepli, 2014). This on-going work places emphasis on the variety of dialects present in Swiss German. Taking advantage of the fact that dialect information is

[1] Download link: http://kitt.cl.uzh.ch/kitt/noah/corpus

[2] Translation: *"The first day we went shopping."*

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 108–109, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

available as metadata in our corpus, the PoS-tagger was trained for each dialect separately. We focused on the five dialects for which the largest amount of training data is available and evaluated these through a 10-fold cross-validation. Each model was trained on 4,000 tokens. The results can be observed in Table 1. More training data will be needed to improve the performance of these models.

Dialect	Accuracy
Aarau	85.73%
Basel	85.28%
Bern	87.85%
Ostschweiz	85.77%
Zürich	87.47%

Table 1: PoS-Tagging accuracy for each of the five dialects.

4 Dialect Identification

As a second application to this corpus, we are in the process of building a dialect identification system. Scherrer and Rambow (2010b) showed that dialect identification via a character n-gram approach could indeed perform well even for very similar dialects, given that sufficient training data from different sources is available. With this corpus, which provides a variety of text genres and dialect information as metadata for most of the included articles, we believe that we have a solid data set to develop a dialect identification model.

In order to implement a baseline system for five major Swiss dialects (Aarau, Basel, Bern, Ostschweiz and Zürich) we compiled a development set of 1,470 sentences and a test set of 250 sentences (50 per dialect). The trained dialect ID model uses a character-based trigram approach. We trained a trigram language model for each dialect and scored each test sentence against every model. The predicted dialect was chosen based on the lowest perplexity.

Table 2 shows the results of this baseline system. Overall, this model reached an F-score of 0.66. To improve this model in the future, the limited amount of training data and the similarity between dialects will have to be taken into account.

5 Conclusion and Future Work

We have extended the corpus for Swiss German dialects to 115,000 manually PoS-tagged tokens.

Dialect	P	R	F
Aarau	0.30	0.36	0.33
Basel	0.54	1.0	0.70
Bern	0.52	0.76	0.62
Ostschweiz	0.68	1.0	0.81
Zürich	0.74	1.0	0.85
Average	0.56	0.82	**0.66**

Table 2: Performances of the trigram model on the test sentences. The columns refer to precision, recall and F-score respectively.

Furthermore, we improved the quality of the data by conducting consistency tests. Employing this improved corpus, we experimented with two possible NLP applications. First, we trained dialect-specific PoS-tagging models which reached between 85% and 88% accuracy. Second, we implemented a baseline system for dialect identification for future research. This system based on character-based language models achieved an overall F-score of 0.66. The dialect ID model for the Swiss German dialect continuum is subject to future work.

References

M. Dickinson and D. Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 107–114. Association for Computational Linguistics.

A. Gesmundo and T. Samardžić. 2012. Lemmatisation as a tagging task. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 368–372. ACL.

N. Hollenstein and N. Aepli. 2014. Compilation of a Swiss German dialect corpus and its application to pos tagging. *COLING 2014*, page 85.

Y. Scherrer and O. Rambow. 2010a. Natural language processing for the Swiss German dialect area. In *Proceedings of the Conference on Natural Language Processing (KONVENS)*, pages 93–102, Saarbrücken, Germany.

Y. Scherrer and O. Rambow. 2010b. Word-based dialect identification with georeferenced rules. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1151–1161. Association for Computational Linguistics.

A. Schiller, S. Teufel, C. Stöckert, and C. Thielen. 1999. Guidelines für das Tagging deutscher Textkorpora mit STTS, August.

Annotating Modality Interdependencies

Eva Reimer[1], Bianka Trevisan[1], Denise Eraßme[1], Thomas Schmidt[2], Eva-Maria Jakobs[1]
[1]RWTH Aachen University
{e.reimer, b.trevisan, e.m.jakobs}@tk.rwth-aachen.de
[2]Institut für Deutsche Sprache (IDS) Mannheim
thomas.schmidt@ids-mannheim.de

Abstract

This paper discusses computational linguistic methods for the semi-automatic analysis of modality interdependencies (the combination of complex resources such as speaking, writing, and visualizing; MID) in professional cross-situational interaction settings. The overall purpose of the approach is to develop models, methods, and a framework for the description and analysis of MID forms and functions. The paper describes work in progress—the development of an annotation framework that allows annotating different data and file formats at various levels, to relate annotation levels and entries independently of the given file format, and to visualize patterns.

1 Objective

Professional verbal interaction settings are often characterized by *modality interdependencies* (*MID*): the combination of complex resources such as speaking, writing, and visualizing. In the research project ModiKo (funded by the DFG, JA1172/3-1), we investigate MID in professional interaction settings. The aim is to develop methods and a framework for the systematic description and analysis of MID forms and functions. Both the quantity of data and the complexity of phenomena to be examined require novel methods. Ultimately, we envisage an annotation framework that allows to annotate different data (text, video, sketches) and file formats (.txt, .mpeg, .jpeg) at various levels, to relate annotation levels and entries independently of the given file format, and to visualize patterns of relations.

2 Case study

The project ModiKo uses data generated in a former research project (IMIP, 2008-2011, BMBF) investigating professional interactions as part of industrial process modelling. In the case study, experts interview employees in a company aiming to understand how production processes are organized. Figure 1 shows a typical situation and example of our data. A process modeler (on the right) interviews an employee (on the left) to get details about the clearance of goods.

Figure 1. Case Example

During the interview, the actors interrupt each other, speak simultaneously, make notes on a clipboard to fix information, or use the clipboard to visualize parts of the production process. By doing so, they combine resources (speaking, writing, visualizing).

3 Corpus

The original database consists of video files (548 minutes, .mpeg), transcripts of verbal interactions (266 pages, .docx) and sketches (89 sheets, .jpeg). The transcription of the verbal parts follows GAT 2 (Selting et al 2009).

The project's research interests—to investigate MID forms and functions—require a more sophisticated solution because of the broad range of phenomena to be considered. For a precise and flexible handling of data, the ModiKo corpus distinguishes three types of documents: (1) *primary documents* (video files of the observed professional interactions and files of the sketches produced by the actors involved), (2) *secondary documents* (multimodal transcripts of the primary documents), and (3) *tertiary documents* (multi-level annotations of the secondary documents). The tertiary document allows the researcher to annotate different phenomena such as speech-

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 110–111, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

accompanying gestures (e.g. to point at sth), objects used in the observed situations (e.g. a clipboard), or contextual and linguistic information. The transformation of primary into secondary documents is an interpretative work.

4 Approach

In the following, a first approach for the multi-level annotation (tertiary documents) is outlined. The approach builds up on existing tools for multimodal annotation (e.g. ELAN, ANVIL) (Sloetjes et al 2011; Kipp 2014). However, these tools only allow restricted linking of annotations, particularly, the linking of text and video data annotation with sketches (Kipp 2014).

For the implementation of MID annotation, ModiKo aims at uncovering challenges and requirements for the development of a holistic annotation tool. The approach is driven by both the ModiKo research objective and the data richness of the case example. The analysis of MID forms and functions and the identification of related patterns require an efficient annotation method allowing to annotate different data and file formats at various levels, to relate annotation levels and entries independently of the given file format, and to visualize patterns.

The approach is innovative because of the complex annotation tool combining different tasks or objectives: the annotation of speech-related phenomena, a linguistic multi-level description of the primary document (Trevisan 2014), and the annotation of MID forms and functions. The annotation of speech-related phenomena and the linguistic multi-level annotation are a pre-condition for the detection of patterns indicating MID forms and functions.

The development of the annotation system is a highly challenging task for different reasons: it requires pioneering in the development of annotation levels, categories, and rules for the description and detection of MID forms and functions. Part of the approach is a quantitative analysis: How often does a MID form or function occur in the interaction scheme? What are typical indicators for a certain form or function?

Despite the quantity and heterogeneity of the data, the annotation system must fit criteria such as efficiency, flexibility, and coherence, i.e., minimized effort needed to achieve the intended annotation, possibility to add, correct, modify categories and annotation levels, as well as consistency and reliability of annotations.

In ModiKo first steps have been completed: indicators for MID forms were identified manually. A prominent example is the verb *schreiben* (*to write*) in the phrase *ich schreib hier mal rein.* The term indicates the MID form speaking/writing (Ullrich et al forthcoming). The existing corpus was transformed into a form suitable for manual and (semi-)automatic analysis and annotation. To this end, the selected text files (transcripts) were transferred to EXMARaLDA (Schmidt and Wörner 2014) and bundled with related video files and sketches. Videos and transcripts were aligned, the diagrams' content modeled in XML, and metadata was systematically recorded for all data types. The first half of the corpus in this form is now ready for tokenization and PoS-tagging. Finally, for each file linguistic annotation levels were defined and created. At the moment, the linguistic annotation follows Trevisan (2014). The annotation levels for MID forms are work in progress.

5 Outlook

Future work will focus on the development of suitable visualization methods for the interplay of data from different modalities, an optimization of the semi-automatic annotation, and a more systematic approach for the annotation of videos. Concluding, the transformation in tertiary documents is a first step to overcome the challenges for a formal and systematic description of MID.

References

Michael Kipp. 2014. ANVIL: The Video Annotation Research Tool. *Handbook on Corpus Phonology*. Oxford University Press: 420-436.

Thomas Schmidt and Kai Wörner. 2014. EXMARaLDA. *Handbook on Corpus Phonology*. Oxford University Press: 402-419.

Margret Selting et al. 2009. Gesprächsanalytisches Transkriptionssystem 2 (GAT 2). *Gesprächsforschung-Online-Zeitschrift zur verbalen Interaktion* (10).

Han Sloetjes, Aarthy Somasundaram, and Peter Wittenburg. 2011. ELAN–Aspects of Interoperability and Functionality. *ISCA*: 3249-3252.

Bianka Trevisan. 2014. Bewerten in Blogkommentaren. Mehrebenenannotation sprachlichen Bewertens. Dissertation. RWTH Aachen University.

Anna Ullrich, Eva-Maria Jakobs, and Denise Eraßme. Forthcoming. „ich schreib hier mal rein ähm". Modality-taking–Schreibhinweise in professionellen mündlichen Interaktionssituationen.

Annotation and analysis of the LAST MINUTE corpus

Dietmar Rösner, Rico Andrich, Thomas Bauer, Rafael Friesen, Stephan Günther
Otto-von-Guericke Universität
Institut für Wissens- und Sprachverarbeitung (IWS)
Postfach 4120, D-39016 Magdeburg
roesner@ovgu.de

Abstract

This paper presents and discusses the techniques used for annotation and analysis of the LAST MINUTE corpus. This corpus comprises multimodal recordings (audio, video, bio-psychological data, verbatim transcripts) of Wizard of Oz simulated naturalistic human companion interactions in German.

1 Introduction

How do 'naive' users spontaneously interact with a system that – like companion systems (Wilks, 2010) – allows them to converse in spoken natural language? Can distinct user groups be detected based on observed linguistic behaviour? How do observed linguistic markers correlate with socio-demographic or psychometric data of the users?

These are issues that are highly relevant for the design of companion-like systems that shall flexibly adapt to their users based on personal characteristics and preferences as well as on the current situation. For the investigation of such questions, corpora with recordings of naturalistic interactions between users and (typically Wizard of Oz (WoZ)) simulated systems are indispensable assets (Legát et al., 2008; Webb et al., 2010).

Analysis of corpora of transcribed naturalistic interactions demands for different types of processing: shallow techniques with broad coverage as well as fine-grained analyses of dedicated passages in the textual recordings. Often these approaches are employed in sequence: first, quantitative analyses based on shallow processing (e.g. detection and counting of structures captured by regular expressions) result in statistical distributions for feature values of interest. Then, for a follow up in-depth analysis of the resp. extreme cases or outliers, qualitative approaches need to be employed.

In this paper we report on the role of shallow and 'deep' techniques – in the sense just presented – in evaluating one such corpus for German: the LAST MINUTE corpus (LMC). This corpus is derived from a large scale Wizard of Oz (WoZ) experiment where users had to solve a mundane task with the need for planning, replanning and strategy change (Frommer et al., 2012b; Rösner et al., 2014).

The paper is organised as follows: In section 2 the LAST MINUTE corpus is shortly presented. This is followed by a discussion of the methods used to analyse the transcripts from this corpus (section 3). Section 4 presents and discusses results from the analysis of discourse (4.1), behavior (4.2) and wizard errors and inconsistencies (4.3). In the summary (section 5) we discuss consequences for the design of future companion systems.

2 LAST MINUTE corpus

The experiment that underlies the multimodal recordings in the LAST MINUTE Corpus was designed in such a way, that the dialogs between simulated system and users were on the one hand restricted enough but on the other hand still offered enough opportunities for individual variation (Frommer et al., 2012b; Rösner et al., 2012). The domain chosen – packing a suitcase for a holiday trip of fourteen days – was mundane enough not to require any specialist knowledge as a prerequisite on the side of the subjects. As a key aspect, an inherent need for re-planning (need for unpacking after reaching a weight limit) and for strategy change (from summer to winter items after the delayed weather information about the target location) was built into the WoZ szenario.

The LMC is a valuable resource based on a large number of highly formalised, yet still variable experiments with subjects balanced with respect to gender and age group. In addition to work based on the verbatim transcripts the LMC has as well been employed for research based on other modalities, i.e. in audio analysis (e.g. (Prylipko et al., 2014)), in video analysis and in fusion of analysis

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 112–121, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

results from different modalities (e.g. (Frommer et al., 2012a)).

As a resource the LMC is 'middle ground' between on the one hand data (or a corpus) from a small scale experiment with narrow interactions and a single hypothesis only and on the other hand a corpus based on recordings from virtually unrestricted real life interactions (like e.g. Vera am Mittag (Grimm et al., 2008) with recordings from a German TV talk show). A more detailled presentation of the LAST MINUTE Corpus is given in (Rösner et al., 2014).

3 Methods

3.1 Discourse Annotation

The LAST MINUTE corpus comprises transcripts of all N = 133 experiments performed. On average each experiment took approximately 30 minutes real time, summing up to more than 56 hours of recorded interactions. In order to be able to quantitatively compare and contrast different dialog courses an adequate representation is needed.

The transcripts in the LMC are annotated with labels for the series of subsequent dialog acts of user and system, the so called dialog act representation (DAR, (Rösner et al., 2014)). This level of representation is independent of the domain of discourse, i.e. it is by no means restricted to the very task presented in LAST MINUTE but is applicable to all types of task-oriented user companion dialogs.

An example The DAR example in Table 4 (cf. appendix) is taken from a dialog segment where a subject (S; here: 20110401adh) tries to pack a (winter) coat but the requests for packing (RP) are rejected (RjP) by the wizard (W) several times (SRP WRjP pairs) and therefore the subject has to request the unpacking of several other items (SRU WAU pairs) in order to create sufficient space. Please note the emotional expression of relief (*'gott sei dank'*, engl. *'thank god'*) when the subject finally succeeds.

3.2 Dialog success measures (DSMs)

A LAST MINUTE experiment is made up of two major phases: a personalisation phase followed by the problem solving phase. In personalisation the system prompts the subject for personal data and stipulates narratives, for example about prior experiences with technical items. In the problem solving phase users have the option to express their requests for the various available actions (packing, change of selection category, unpacking, listing of suitcase contents, ...). User requests may be either accepted and confirmed by the system or they may be rejected.

This allows to evaluate the dialog course both locally and globally. Locally accepted requests are evaluated positively and rejections resp. get a negative score.

The relation between subject requests and their acceptance or rejection allows to define measures for the global dialog success in the problem solving phase of LAST MINUTE (Rösner et al., 2014):

- ratio between the accepted subject requests and the total number of subject requests (termed **DSM1**)
- ratio between the accepted subject requests and the total number of turns (i.e. not only subject requests) in problem solving (termed **DSM2**)

Thus for all subjects the following must hold: $0 \leq DSM2 \leq DSM1 \leq 1$.

3.3 Discourse analysis

Both dialog success measures are employed in the following analyses. They allow the following types of investigations:

- How do user groups based on socio-demographics differ with respect to global dialog success (cf. 4.1)?
- How do user groups that are defined based on distinct behavior during the experiment differ with respect to global dialog success (cf. 4.2)?

The methods employed in discourse analysis of the LMC are as follows: The LMC comprises full transcripts of N = 133 experiments. The transcripts are available as an XML-based data structure in the FOLKER format (Schmidt and Schütte, 2010). This highly structured format contains not only the transcription of all user and wizard contributions of every experiment in their relative temporal order, but also additional annotations. These annotations range from recorded nonphonological events (e.g. sighing, coughing, ...) to discourse level events (e.g. dialog act labels).

Starting from the FOLKER encoded transcripts we determine – typically with shallow techniques,

often based on regular expressions – features (or markers) either for complete transcripts or for their subparts (e.g. personalisation vs. problem solving or their resp. subphases). Such features are calculated on every level of the linguistic system, i.e. from the lexical level (e.g. occurrence counts for classes of lexical items) via syntax (e.g. preferred syntactic style in user commands) to semantic classifications (e.g. local meaning of user utterances) and pragmatic concerns (e.g. can the user's current intention be detected?).

The feature sets derived in this way then undergo a thorough analysis in which we combine quantitative and qualitative approaches from corpus linguistics (Gries, 2009). The quantitative methods start with compiling the empirical distributions of the feature values. These are visualised appropriately and e.g. tested for normality vs. skewness. Transcripts of (extreme) outliers are then additionally checked qualitatively – typically by human interpretation – in order to detect and discuss possible causes for deviation.

A recurring finding for virtually every investigated feature is that the distribution of feature values shows a large variance. This even holds for features that quantify aspects of the overall extent of the highly standardised experiments (cf. table 1).

Analysing the cause of the observed variance is a major issue in the work reported here. The different user groups based on socio-demographic features – i.e. age group (young vs. elderly subjects) and the four combinations of age group with gender – are potentially a primary source of the observed variance. Indeed: for many features the differences between the age groups and for gender-conditioned subgroups prove to be significant (cf. section 4.1).

When significant differences in the distribution of feature values have been found between socio-demographic groups then the additional question arises if these differences correlate significantly with differences in dialog success (as measured with DSM1 and DSM2).

3.4 Behavioral Analysis

In behavioral analyses errors that users make and problems they face are valuable assets. This holds especially when early occurrences of problematic user behavior prove to be predictive for later global dialog success or failure. As will be elaborated in section 4.2, early errors in the personalisation phase could be identified that bear this predictive power. The data analysis methods employed in evaluating observed differences in user behavior are the same as presented above. The only difference is that user groups are now defined on *observed differences in behavior in the course of the dialogs* and no longer on a priori differences between subjects like age group or gender.

4 Results

An early result about the socio-demographic subgroups in the cohort of the LAST MINUTE experiment was that the subgroup of young women has significantly higher values of dialog success measures than the other three subgroups (i.e. young men, elderly women, elderly men, cf. (Rösner et al., 2014)). What other significant differences between these subgroups can be detected? For lack of space, we will concentrate on differences between socio-demographic subgroups with respect to verbosity and with respect to usage of politeness particles.

4.1 Discourse analysis: Age and Gender matters

Differences in verbosity: We employ the ratio of Tokens per Turn (TpT) as a verbosity measure for the user contributions in the LMC dialogs. Given the distinct nature of the different phases in the LAST MINUTE experiment, the measure varies between the more narrative oriented phases in personalisation and the phases of problem solving with a preference for usually shorter commands.

Major results for problem solving include (cf. fig. 2, table 2): age group matters, young subjects are significantly less verbose than elderly (Wilcoxon: W = 1722, p = 0.03251), whereas gender gives insignificant differences only. In addition, the pairings of age group and gender result in significant differences as well (Kruskal-Wallis chi-squared = 8.375, df = 3, p = 0.03886).[1]

Similar results hold for TpT values for other parts of the experiment. A point in case is for example the narratives phase in personalisation (cf. table 2).

Politeness particles as indicators for CASA: When humans conversing with a computer system do employ politeness particles this can be seen as indicator for (mindlessly) treating Computers as Social Actors (CASA, (Nass and Moon, 2000))

[1] Unless noted otherwise all statistical tests and calculations have been performed with the R language (R Development

Table 1: Examples of empirical distributions of features based on complete transcripts (N = 133)

marker	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD	total
tokens	266.0	444.0	545.0	602.7	699.0	1601.0	247.34	80160
turns	62.00	81.00	86.00	86.08	91.00	111.00	9.95	11448
TpT	2.804	5.143	6.282	7.060	8.109	19.290	2.95	n.a.

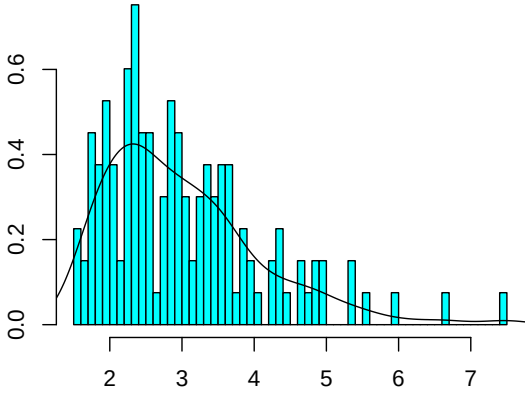


Figure 1: Distribution of Tokens per Turn (TpT) ratios for problem solving ($N = 133$)

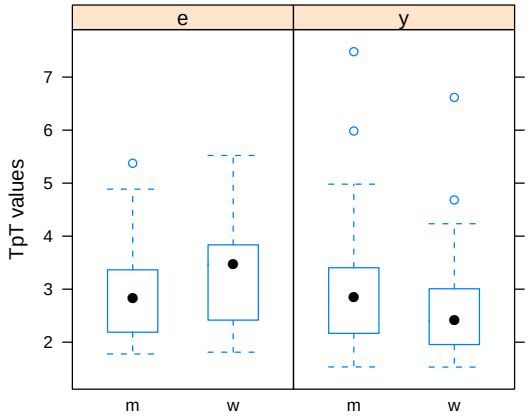


Figure 2: Tokens per turn ratios for problem solving conditioned by age group (e = elderly, y = young) and gender (m = men, w = women) ($N = 133$)

Table 2: Test results (Wilcoxon tests) for differences in mean verbosity between socio-demographic groups (e = elderly, y = young; m = men, w = women)

marker	g1	rel	g2	p-value
TpT probl. solv.	y	<	e	0.02016
TpT probl. solv.	m	<	w	n.s.
TpT pers. narratives	y	<	e	0.00423
TpT pers. narratives	w	<	m	n.s.

Table 3: Age group and gender differences in usage of politeness particles

nr pol. parts	em	ew	ym	yw	$\sum$
0 (below median)	7	5	**19**	**24**	55
1 (at median)	3	2	6	4	15
$>$ 1 (above median)	**19**	**25**	11	8	63

Counting the number of occurrences of politeness particles *'bitte'* (engl. *'please'*) and *'danke'* (engl. *'thank you'*) in user utterances per transcript provides distributions for all N = 133 subjects as visualised in figs. 3 and 4. Please note: 55 subjects never uttered one of these politeness particles. The median lies at one occurrence.

Again age matters: The subgroup with counts of used politeness particles above the median is clearly dominated by elderly subjects, whereas the subgroups below and at the median are dominated by young subjects (cf. table 3).

Kruskal tests show significant results for the two age groups (Kruskal-Wallis chi-squared = 24.6171, df = 1, p-value = 6.993e-07) and the four pairings of gender and age group (chi-squared = 26.0632, df = 3, p-value = 9.251e-06), but for gender we get insignificant differences only.

4.2 Behavioral analyses

In the following paragraphs subgroups of subjects with distinct problems are investigated.

Early problems with 'tell and spell': At the very beginning of the personalisation phase every subject is prompted:

```
Bitte nennen und buchstabieren Sie zunächst Ihren
Vor- und Zunamen!
```

Core Team, 2010) (Baayen, 2008)

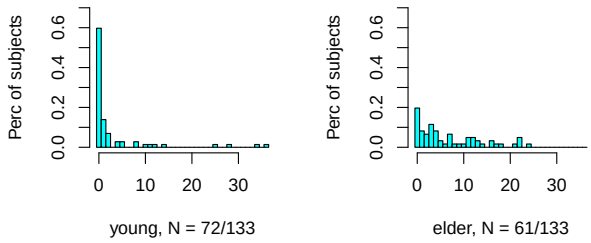


Figure 3: Distributions of number of occurrences of politeness particles in user utterances per transcript, subgroups young vs. elderly (Please note the different relative amount of zero occurrences)

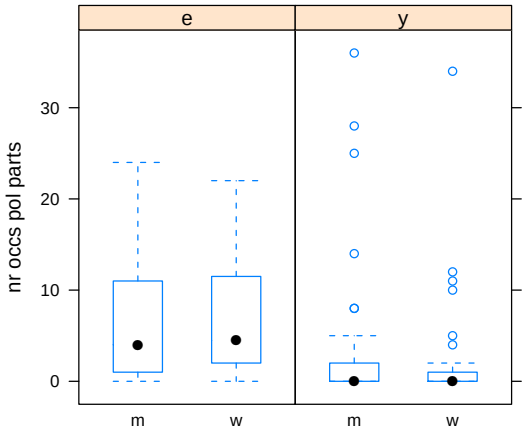


Figure 4: Number of occurrences of politeness particles in user utterances per transcript, conditioned by age group and gender (Please note the zero medians for the young subgroups and the resp. outliers)

```
[Please tell and spell your first name and surname!]
```

Some subjects need several trials, some even completely fail to provide the requested information. An example excerpt: subject 20110131bcl (anonymized) with three unsuccessful trials

```
{00:18} W guten tag und herzlich willkommen (.) ...
          [Hello and welcome. ...]
          bitte nennen und buchstabieren sie
          zunächst ihren vor und zunamen
          [please tell and spell your first name
          and surname]
{00:45} (1.89)
{00:47} P charlotte kurz
{00:48} (5.88)
{00:54} W bitte nennen und buchstabieren sie
          zunächst ihren vor und zunamen
{00:58} (---)
{00:59} P charlotte (.) kurz
{01:00} (8.58)
{01:09} W bitte nennen und buchstabieren sie
          zunächst ihren vor und zunamen
{01:13} (1.08)
{01:14} P charlotte kurz (3.8) ((schnalzt)) (.)
          mein vorname ist charlotte °h (-)
          mein familienname ist kurz
          [charlotte kurz (3.8) ((smacks)) (.)
          my first name is charlotte °h (-)
          my family name is kurz]
```

Please note: the wizards issued no more than three prompts, even when the third trial was still faulty. (Obviously, in such cases a runtime companion system should give a more adequate system response and not simply repeat the partially fulfilled prompt).

From N = 133 subjects the answer to the prompt 'Please tell and spell . . . ' is accepted after the first response for 113 subjects, after the second trial for 12 subjects and after the third trial for 8 subjects. Actually the task completion ratio is even worse: 20 subjects only *spell* but do not tell their name, 2 more leave the first name out. (Please note: wizards did not react to these latter types of incomplete answers). In sum: from $N = 133$ subjects the answer to the prompt 'Please tell and spell . . . ' is wrong or incomplete in at least 34 cases (i.e. 25.6%), full task completion is reached only in 74.4% of the cases.

The age groups differ with respect to task completion: exactly 2 spelling request are needed by 5 elderly and 7 young subjects, whereas the 8 subjects with exactly 3 trials are all elderly.

Why should 'tell and spell . . . ' be a problem? The failure of subjects with respect to this task may be attributed to 'inattentional deafness' (Dalton and Fraenkel, 2012) or to effects of cognitive aging (Wolters et al., 2009) in general.

This leeds to the following **hypothesis**:

> Subjects with problems with the 'tell and spell . . . ' task will have problems with other parts of the experiment as well and will have lower values in the dialog success measures.

To test this hypothesis we do contrast the distribution of dialog success measures for the no problem group (i.e. exactly one trial) and the complementary problem group (i.e. with two or more trials).

The difference in means for DSM2 (no problem: 0.7075; problem: 0.6612) is significant as a Wilcoxon test reveals (W = 773, p-value = 0.02482; the distribution of the no problem group clearly differs from a normal distribution).

Similar results hold for DSM1: the problem group has poorer dialog success values and - again - these differences between the no problem and the problem group are significant (Wilcoxon: W = 770.5, p-value = 0.02382).

In sum: problems with the very first task in personalisation are an early predictor for later problems in the problem solving dialog of LAST MINUTE.

Early predictor: problems in 'data acquisition'. In the personalisation phase initiative lies primarily with the system. Here a typical adjacency pair (Jurafsky and Martin, 2008) is made up of a wizard prompt or question followed by a user narrative or answer.

Already the third wizard prompt demands for quite a number of personal data thus challenging not only the subjects's hearing understanding and comprehension abilities but as well her/his short term memory capacity.

```
Damit sich das Computerprogramm individuell an Sie
anpassen kann, sind einige konkrete Informationen
zu Ihrer Person erforderlich. Können Sie bitte zu
folgenden Punkten Angaben machen: Ihr Name, Ihr
Alter, Ihr Wohnort, Ihr Beruf, Ihr Arbeitsort, Ihre
Familie, Ihre Körpergröße, Ihre Konfektionsgröße,
Ihre Schuhgröße?

[in order to adapt to you the computer programme
needs some specific pieces of information. Could you
please give the following details: your name, your
age, your place of residence, your profession, your
place of work, your family, your body height, your
clothing size, your shoe size?]
```

If subjects do not give all of the requested data they are reprompted for missing data with questions of the type *'bitte ergänzen sie angaben zu . . . '* (engl. : *'please complete information about . . . '*).

Thus, in cases of a normal dialog course in personalisation the sources of variation are reprompts (e.g. 'tell and spell'), the number of questions of the type 'please complete information about . . . ' and the number of prompts for 'more detail'. Sources of variation in unforeseen courses are user questions, e.g. caused by hearing and/or understanding problems like in the following excerpt (subject 20110131bcl):

```
{03:09} W bitte ergänzen sie angaben zu ihrer körpergröße
         [please complete information about your body height]
{03:13} P ihrer was ihrer welcher größe
         [your what your which height]
{03:18} W bitte ergänzen sie angaben zu ihrer körpergröße
         [please complete information about your body height]
{03:22} P das weiß ich nicht (--) welche größe denn das weiß
          verstehe ich jetzt nicht so richtig
         [I do not know (--) which height
         I do not really know understand this]
```

Please note: a higher number of adjacency pairs in personalisation thus in general indicates problems on the subject's side. The empirical distributions of the total number of user turns in data acquisition, conditioned by age group and gender, are visualised in fig. 6.

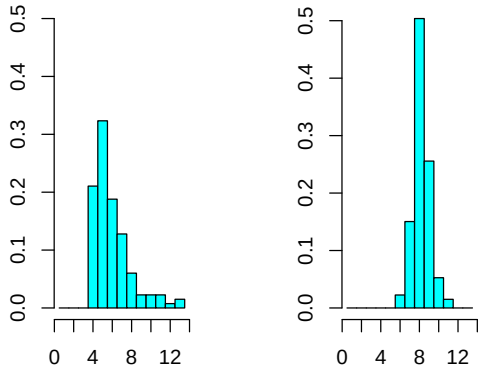


Figure 5: Total number of user turns in subphases of personalisation per transcript ($N = 133$): data acquisition (left), narratives (right)

In the following we perform a median split with respect to the total number of turns (i.e. adjacency pairs) in the subphase 'data acquisition'. The overall result: the subgroup of subjects below (and at) the median (of 5) has significantly better values for both dialog success measures in problem solving. For both dialog measures Wilcoxon tests judge the differences between the groups as significant (DSM1: W = 1746, p-value = 0.04035; DSM2: W = 1604.5, p-value = 0.00718).

Issues of control: Being in control or not is an important issue in a dialog. In the LM experiments the issue of control is underlying the distinction between two types of category change:

- subject induced category change (SICC): the subject explicitly utters a request for category change,
- wizard induced category change (WICC): the wizard enforces a category change.

More than half of the subjects are 'in control' in this sense. They have either zero or only one or at most two wizard induced category changes (from a total of 14 in a complete experiment). The complement of this group ('poor control') has between three and up to 10 WICCs.

Poor control of category changes (i.e. `WICCs > 2`) predicts poor global dialog success. The two subgroups – at and below the

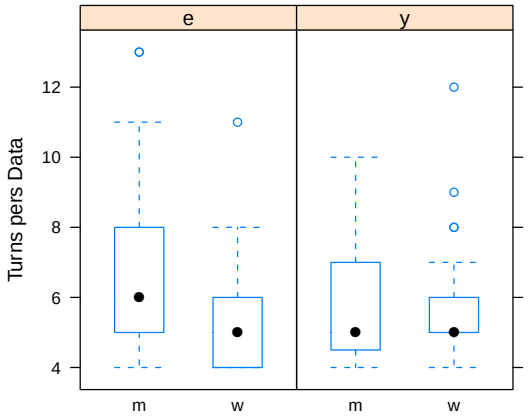


Figure 6: Total number of user turns in data acquisition per transcript, conditioned by age group and gender ($N = 133$)

WICC median of 2 or above the WICC median, resp. – show significant differences in both global dialog success measures (DSM1: Wilcoxon test, p-value = 0.02614; DSM2: Wilcoxon test, p-value = 3.610e-6).

Again: age group makes a major difference between the two subgroups whereas gender differences are only of minor relevance.

Long pauses as indicators of helplessness: There is a subgroup of the subjects with poor control that – after some choices in a category – passively wait without any further action, sometimes for 40 seconds or more, until the system finally enforces a category change (WICC).

Not surprisingly the occurrence of such a type of long pause is again a predictor for global dialog failure. The subgroup of ten subjects that have at least one occurrence of a pause longer than ten seconds before a WICC has significantly poorer dialog success measures when compared to the complementary group of 123 subjects without such pauses (Wilcoxon tests, DSM1: W = 268, p-value = 0.0031, DSM2: W = 138.5, p-value = 4.87e-05).

4.3 Wizard problems: errors and inconsistencies

The LAST MINUTE experiment is a carefully designed and highly standardised experiment, based on a detailled manual (Frommer et al., 2012b), performed by intensively trained personnel (wizards) with elaborated computer support. In spite of intensive training and the detailled manual the wizards did not always operate consistently and accurately. This is not surprising given the large number of subjects and the time span of nearly a year for the completion of all N = 133 experiments. Fine-grained investigation of wizard behavior is necessary for quality assurance: it helps to avoid erroneously attributing a problematic dialog course to the subject when actually the wizard caused the problem.

We found for example inconsistent wizard behavior by analyzing the subject initiated category changes. It turned out that some rejected wordings would have been accepted by different wizards or even by the same wizard in other situations.

We also found wizard errors, characterised as situations where a wizard did not operate according to the guidelines of the manual. One type of such a wizard error is the rejection of a subject request with 'your input cannot be processed' (WRjNp) when indeed the intention of the subject was clearly recognizable and the intended action was performable.

An example (subject 20110329aus):

```
{15:04} W ... bevor weitere artikel ausgewählt wer-
          den können (.) müssen sie für genügend
          platz im koffer sorgen (.) hierfür können
          bereits eingepackte artikel wieder ausge-
          packt werden (.) auf nach frage erhalten
          sie eine aufzählung der bereits ausge-
          wählten artikel [... before more items
          can be chosen (.) you have to create
          enough space in the suitcase (.) for this
          purpose already packed items can be un-
          packed (.) upon request you can get a
          listing of the already chosen items]
{15:27} (2.81)
{15:30} P ja bitte [yes please]
{15:31} (3.85)
{15:35} W ihre aussage kann nicht verarbeitet werden
          [your statement cannot be processed]
```

In sum: a manual like (Frommer et al., 2012b) is *necessary*, but by *no means sufficient* for successful experiments. A manual defines the overall structure of experiments, but for non-trivial interactions nearly necessarily many questions will arise.

In other words: spontaneous improvisation by wizards seems unavoidable, but it has a price: unreflected and unsupervised improvisation may – very likely – result in inter-session inconsistencies in wizard behavior. Indeed, the LAST MINUTE corpus contains many occurrences of inter-session wizard inconsistencies.

An example of such inconsistencies is the acceptance or rejection of synonyms. In some cases wizards accepted synonyms of packed items, in other cases they did not. In the following excerpt from 20101220bmh the use of a synonym is accepted

```
{18:51} P vier paar socken auspacken hh°
          [unpack four pairs of socks]
{18:58} P vier paar strümpfe wurden entfernt (.)
          [four pairs of socks were removed]
          sie können fortfahren [please proceed]
{19:02} P ein badeanzug [a bathing suit]
```

In contrast, the same synonym is rejected for subject 20100901amb:

```
{15:23} P °h entferne (--) drei socken
          [remove three socks]
{15:30} W ihre aussage kann nicht verarbeitet werden
          [your statement cannot be processed]
{15:33} P drei socken [three socks]
{15:38} W der gewünschte artikel ist nicht im koffer
          enthalten [requested item is not contained
          in suitcase]
{15:42} P na ick hab vierzehn socken reinjepackt
          (werden ja wohl) drei drin sein (.)
          ((schmatzt)) °hhh (4.04) welche artikel
          sind im koffer enthalten
          [well I have packed fourteen socks
          then three should very well be there
          (.) ((smacks)) °hhh (4.04) which
          items are contained in the suitcase]
```

Please note: refused unpacking requests of this type are very irritating for subjects. This is underlined by the protesting reaction of the subject.

5 Discussion and Outlook

We have presented examples of analyses of transcripts in the LAST MINUTE corpus of naturalistic human companion interactions and we have illustrated the interplay of shallow, quantitative and broad coverage approaches with qualitative human interpretations. In the following we will summarise major insights from these analyses and discuss their consequences for the design of future companion systems.

5.1 Major insights from analyses

Major insights from the analyses can be summarised as follows:

User groups based on socio-demographics matter. This holds especially for the differences between young and elderly subjects with the former being more successful *on average*. On the other hand, gender matters only when taken into account as a further subcondition after an age group based primary grouping.

One of the sources of communication problems seem to be difficulties in comprehending and memorizing information that was given as spoken language utterances by the system. Such problems occur significantly more often with elderly subjects. Early occurrences of such problems in speech understanding are a strong predictor for global failure of the (independent) later problem solving dialog (cf. 4.2).

A strong indicator for a potential user problem is an overly long pause when the user actually has the turn, i.e. the right to give the next utterance (cf. 4.2).

The analysis of wizard errors and inconsistencies and the analysis of resp. user reactions (cf. 4.3) clearly demonstrates the dominance of semantic and pragmatic expectations of subjects in user companion interaction. Users are obviously puzzled when the system tries to enforce lexical or syntactic constraints that are in conflict with the user's expectations.

5.2 Consequences for the design of companion systems

The findings from the analyses of the dialogs in the LAST MINUTE corpus have consequences for the design of companion systems that are based on speech interaction with their users.

On the one hand differences between socio-demographic groups – especially differences between age groups – have to be taken into account by the dialog management of companion systems. On the other hand the broad variance between individuals (cf. table 1 or (Wolters et al., 2009)), demands for personalised calibration of dialog management strategies. Tests that are easy to perform and evaluate and that have strong predictive power for potential problems in the subsequent global dialog course – cf. section 4.2 – may be employed for this purpose.

In addition the dialog history of the user companion interactions needs to be monitored continuously. Special emphasis shall be given to situations where the user has the turn but does not take it within a certain time span. As discussed in section 4.2 such overly long pauses are strong indicators for problems and helplessness on the user's side and demand for an adequate response by the system.

Acknowledgments

The presented study is performed in the framework of the Transregional Collaborative Research Centre SFB/TRR 62 "A Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG). The responsibility for the content of this paper lies with the authors.

References

R.H. Baayen. 2008. *Analyzing Linguistic Data – A Practical Introduction to Statistics using R*. Cambridge University Press.

P. Dalton and N. Fraenkel. 2012. Gorillas we have missed: Sustained inattentional deafness for dynamic events. *Cognition*, 124(3):367–372.

J. Frommer, B Michaelis, D. Rösner, A. Wendemuth, R. Friesen, M. Haase, M. Kunze, R. Andrich, J. Lange, A. Panning, and I. Siegert. 2012a. Towards emotion and affect detection in the multimodal LAST MINUTE corpus. In *LREC 2012 Conf. Abstracts*, pages 3064–3069.

J. Frommer, D. Rösner, M. Haase, J. Lange, R. Friesen, and M. Otto. 2012b. *Teilprojekt A3 – Früherkennung und Verhinderung negativer Dialogverläufe – Operatormanual für das Wizard of Oz-Experiment; Arbeitspapier des Sonderforschungsbereichs - Transregio 62 'Eine Companion-Technologie für Kognitive Technische Systeme' = Project A3 - Detection and avoidance of failures in dialogues*. Pabst Science Publishers, Lengerich.

S. T. Gries. 2009. *Quantitative Corpus Linguistics with R: A Practical Introduction*. Routledge.

M. Grimm, K. Kroschel, and S. Narayanan. 2008. The Vera am Mittag German audio-visual emotional speech database. In *Multimedia and Expo, 2008 IEEE Int. Conf. on*, pages 865–868.

D. Jurafsky and J. H. Martin. 2008. *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall, 2nd edition.

M. Legát, M. Grůber, and P. Ircing. 2008. Wizard of oz data collection for the czech senior companion dialogue system. In *Fourth Int. Workshop on Human-Computer Conversation*, pages 1 – 4, University of Sheffield.

C. Nass and Y. Moon. 2000. Machines and Mindlessness: Social Responses to Computers. *Journal of Social Issues*, 56(1):81–103.

D. Prylipko, D. Rösner, I. Siegert, S. Günther, R. Friesen, M. Haase, B. Vlasenko, and A. Wendemuth. 2014. Analysis of significant dialog events in realistic human-computer interaction. *Journal on Multimodal User Interfaces*, 8(1):75–86.

R Development Core Team, 2010. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

D. Rösner, R. Friesen, S. Günther, and R. Andrich. 2014. Modeling and Evaluating Dialog Success in the LAST MINUTE Corpus. In *Proc. of LREC'14*, Reykjavik, Iceland. ELRA.

D. Rösner, J. Frommer, R. Friesen, M. Haase, J. Lange, and M. Otto. 2012. LAST MINUTE: a Multimodal Corpus of Speech-based User-Companion Interactions. In *Proc. of the 8th LREC*, pages 2559–2566, Istanbul, Turkey.

T. Schmidt and W. Schütte. 2010. Folker: An annotation tool for efficient transcription of natural, multi-party interaction. In N. Calzolari (Conference Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

M. Selting, P. Auer, D. Barth-Weingarten, J. R Bergmann, P. Bergmann, K. Birkner, E. Couper-Kuhlen, A. Deppermann, P. Gilles, S. Günthner, et al. 2009. Gesprächsanalytisches Transkriptionssystem 2 (GAT 2). *Gesprächsforschung-Online-Zeitschrift zur verbalen Interaktion*, 10.

Nick Webb, David Benyon, Jay Bradley, Preben Hansen, and Oli Mival. 2010. Wizard of Oz Experiments for a Companion Dialogue System: Eliciting Companionable Conversation. In *Proc. of LREC'10*. ELRA.

Y. Wilks. 2010. *Close engagements with artificial companions: key social, psychological, ethical and design issues*, volume 8. John Benjamins Publishing.

M. Wolters, K. Georgila, J.D. Moore, and S.E. MacPherson. 2009. Being Old Doesn't Mean Acting Old: How Older Users Interact with Spoken Dialog Systems. *ACM Trans. Access. Comput.*, 2(1):2:1–2:39.

Tag	german text	english gloss
	...	...
SRP	ein mantel	a coat
WRjP	der artikel mantel kann nicht hinzugefügt werden (.) anderenfalls würde die maximale gewichtsgrenze des koffers überschritten werden	the item coat cannot be added (.) otherwise the maximal weight limit of your suitcase will be exceeded
SNp	((raschelt)) ((schmatzt))	((rustles)) ((smacks))
SNp	(-)	(-)
SRU	ein buch raus	one book out
WAU	ein buch wurde entfernt	a book has been removed
SRP	ein mantel	a coat
WRjP	der artikel mantel kann nicht hinzugefügt werden (.) anderenfalls würde die maximale gewichtsgrenze des koffers überschritten werden	the item coat cannot be added (.) otherwise the maximal weight limit of your suitcase will be exceeded
SRU	badelatschen raus	beach slippers out
WAU	ein paar badelatschen wurden entfernt	a pair of beach slippers has been removed
SRP	ein mantel	a coat
WRjP	der artikel mantel kann nicht hinzugefügt werden (.) anderenfalls würde die maximale gewichtsgrenze des koffers überschritten werden	the item coat cannot be added (.) otherwise the maximal weight limit of your suitcase will be exceeded
SOT	tja	well
SNp	(1.77)	(1.77)
SOQ	was kann man denn noch rausnehmen	well what else can be removed
SNp	(1.48)	(1.48)
SNp	pf pf pf pf pf pf pf	pf pf pf pf pf pf pf
SNp	(4.8)	(4.8)
SRU	zwei bh raus	two bras out
WAU	zwei bhs wurden entfernt	two bras have been removed
SRP	ein mantel	a coat
WAP	ein mantel wurde hinzugefügt	a coat has been added
SOT	gott sei dank	thank god
	...	...

Table 4: Excerpt from transcript with DAR labels: S indicates a subject and W a wizard contribution. Dialog acts include requests (R), rejections (Rj), accepts (A) for actions like packing (P) or unpacking (U). SNp stands for nonphonological utterances, SOT and SOQ for offtalk and questions resp. According to the GAT-2 minimal standard (Selting et al., 2009) short pauses are noted as (.) and (-), longer pauses with their duration (in seconds) in brackets, e.g. (1.77).

Automatic induction of German aspectual verb classes in a distributional framework

Jürgen Hermes
University of Cologne
Department of Linguistics
Linguistic Data Processing
hermesj@uni-koeln.de

Michael Richter
Radboud University
Nijmegen
Department of Linguistics
mprrichter@t-online.de

Claes Neuefeind
University of Cologne
Department of Linguistics
Linguistic Data Processing
c.neuefeind@uni-koeln.de

Abstract

The central question of this study is whether aspectual verb classes (Vendler, 1967) can be induced from corpus data in a fully automatic, distributionally motivated procedure. We propose an operationalization of 'aspectivity' utilizing distributional information about nominal fillers in the argument positions of verbs in combination with aspectual features automatically derived from dependency information. Using a support vector machine classifier and a classification into five aspectual classes (Richter and van Hout, 2015) as the gold standard, we observed excellent results that support our hypothesis.

1 Introduction

This study aims to empirically validate aspectual verb classes in German using corpus data. It is primarily motivated by a lack of studies on the induction of the complete Vendlerian typology. Vendler (1967) distinguished the four aspectual classes: *accomplishments, achievements, states* and *activities*, which are based on the temporal scheme of verbs and verb phrases. The grammatical verb category *aspect* (Klein, 2009) allows for a classification into *species of verb*[s] (Vendler, 1967), which Klein (2009: 22) defines by the five temporal features *qualitative change* ('non-stative' vs. 'stative'), *boundedness* (initial and final boundary, i.e. 'processes' vs. 'events'), *duration* ('punctual' vs. 'non-punctual' contents), *inner quantification*: ('iterative', 'frequentative', 'semelfactive'), *phase* ('inchoative', 'terminative', 'resultative', etc.), whereby the superior criterion is that of perfective vs. non-perfective aspect (Klein, 2009).

Research on aspectual verb classes is of particular relevance because the temporal and causal structure of events can be represented (Vendler, 1967; Fernando, 2004; Gründer, 2008); by aspect, which yields classificatory criteria for linguistic units such as verbs and documents (Siegel, 1997; Siegel and McKeown, 2000).

Based on previous work of Richter and Hermes (2015) we hypothesize, that aspectual verb classes can be automatically induced from the classified nominal fillers in the argument position of verbs. The nominal fillers were combined with aspectual features (co-occurences of specific words / phrases in dependent and governing positions relative to the verb), as we aimed to test whether, and what degree those features would improve the quality of the classification, an additional question being whether from single aspectual features satisfying classification results could be achieved. Our hypothesis refers to the *Distributional Hypothesis* (Rubenstein and Goodenough, 1965; Schütze and Pedersen, 1995; Landauer and Dumais, 1997; Pantel, 2005) which claims that semantically related linguistic elements appear in semantically related contexts. The present study in the framework of a vector space model is also driven by the *Statistical Semantics Hypothesis* (Weaver, 1955; Furnas et al., 1983; Turney and Pantel, 2010) which states that linguistic meaning can be derived from statistical linguistic patterns.

In order to test our hypothesis, we took a test set of 95 verbs from Schumacher (1986). Based on a dependency parse of the SdeWaC

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 122–129, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

corpus we determined the nominal fillers and their classes in argument positions (that is, in subject, direct object, and prepositional object positions) and additionally extracted aspectual features as defined by Vendler (1967) with regard to their structural positions. Both the nouns and the aspectual features were extracted from the respective sentences the 95 verbs occurred in. The indirect object was left out because there were few occurrences of verbs with indirect objects. In addition, Richter and Hermes (2015) brought to light, that indirect objects - contingent upon their sparsity - were weak predictors of verb classes. The aspect-based classification of Richter and van Hout (2015) was used as a gold standard in this study. This classification consists of five classes and extends the typology of Vendler (1967) by adding the class *accomplishments with an affected subject*.

In the present study we represent verbs as feature vectors that consist of nouns in argument positions separated into areas according to their noun classes which were induced by cluster analysis, accompained by aspectual features, the research questions being 1. to what extent the aspectual features will increase the predictive power of the nominal fillers in arguments positions, and 2. whether aspectual features as singletons would be sufficient to predict aspectual verb classes. The test set of verbs was classified in a supervised learning procedure using a support vector machine (SVM) classifier.

2 Related work

There are few studies which address the topic of automatic induction of aspectual verb classes. By focusing on tense forms, Klavans & Chodorow (1992) determined gradual *state*-properties of verbs. Siegel (1997) and Siegel & McKeown (2000) classified verbs using temporal and modal indicators such as temporal adverbs, tense forms and *manner*- and *evaluation*-adverbs into the two aspect classes *states* and *events*. No attempts have been made so far to induce the complete Vendlerian Typology.

Studies on the automatic induction of non-aspectual verb classes from Dorr & Jones (1996), Merlo & Stevenson (2001), Preiss, Briscoe & Korhonen (2007), Joanis, Stevenson & James (2008), Vlachos, Korhonen & Gahramani (2009) and Parisien & Stevenson (2011), amongst others, provide corpus based evidence that argument frames, syntactic subcategorization information and, in addition, aspect (Joanis, Stevenson & James, 2008) are reliable predictors. Merlo & Stevenson (2001), who induced a Levin-compatible classification from the argument structure of verbs, draw the conclusion that the semantics of the argument structure is decisive for the classification of verbs.

Classifications have been empirically induced which are compatible with the classifications of Levin (1993) and Schumacher (1986) but only for German. Examples are classes such as *verbs of existence*, *verbs of linguistic expression and verbs of vital needs*, see Schumacher, (1986) and *verbs of transfer of possession* und *verbs of communication*, see Levin (1993). Schulte im Walde & Brew (2002) induced 14 verb classes from a test corpus of 57 verbs focusing solely on the syntactic information of verbs. In a follow up study, Schulte im Walde (2003) induced 43 verb classes from a test corpus of 168 verbs considering both syntactic and semantic information, and in a replication of this study Schulte im Walde (2004) induced 100 verb classes from a test set of 883 verbs. Schulte im Walde (2003) concludes that in order to get linguistically plausible clusters, both idiosyncrasies and general, more abstract properties of verbs have to be taken into consideration.

3 Methodology

In this study, we classified a selection of 95 common German verbs taken from Schumacher (1986), who defines seven lexical semantic macrofields; *Verben der allgemeinen Existenz* ('verbs of general existence'), *Verben der speziellen Existenz* ('verbs of special existence'), *Verben des sprachlichen Ausdrucks* ('verbs of linguistic expression'), *Verben der Differenz* ('verbs of difference'), *Verben der Relation und des geistigen Handelns* ('verbs of relation and mental processing'), *Verben des Handlungsspielraums* ('verbs of freedom of action') and *Verben der vitalen Bedürfnisse* ('verbs of vital needs'). The macrofields are split into 30 subfields. We chose the verbs randomly from the thirty subfields, the only criterion being the inclusion of every subfield in order to cover the total semantic range of Schumacher's typology (1986).

Figure 1 presents the complete workflow of our analysis, from raw sentence data taken from the SdeWaC corpus (down left) to sets of classified verbs (up right). Below we describe the

main steps of the workflow as numbered in *figure 1*.

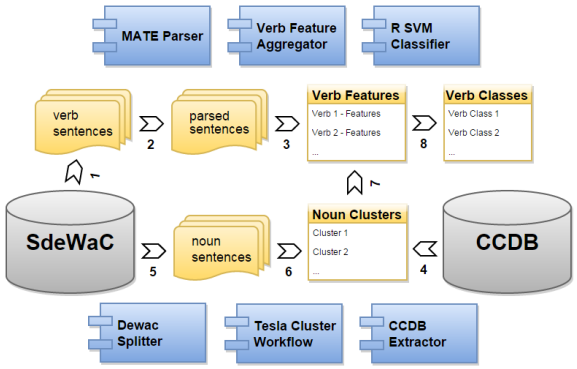


Figure 1 ***Synopsis of our processing workflow. Each processing step is realized as a largely independent, subordinated workflow, e.g. the description of the Tesla Cluster Workflow in figure 2 (see below).***

3.1 Determination of verb features

Initially, we extracted 3000 sentences for each verb in our list **(step 1)** and parsed them **(step 2)** with the *Mate Dependency Parser* from Bohnet (2010)[1] in order to determine subjects and objects (accusative, dative, and prepositional) for each verb **(step 3)** and in addition to determine aspectual features, which were suggested by Vendler (1967) to distinguish aspectual verb classes. The list of aspectual features is given below:

1. verb in imperative form,
2. verb complex with *aufhören / stoppen* ('to stop / to finish') as governing verbs,
3. verb complex with *überzeugen* ('to convince') as governing verb,
4. matrix verb with time adverbials for durations, like *minutenlang* ('for minutes'), *in einer Minute* ('in a minute'),
5. matrix verb with time units, like *minute* ('minute'), *jahrhundert* ('century'),
6. matrix verbs with *seit* ('since'), combined with a time unit,
7. matrix verb with adverbials *sorgfältig / mit Sorgfalt* ('careful / with care'),
8. matrix verb with adverbials *absichtlich / mit Absicht* ('on purpose'),
9. matrix verb with adverbials *fast / beinahe* ('almost').

[1] See \url: https://code.google.com/p/mate-tools/

To reduce the feature space and to increase the allocation density of the vectors, we clustered all nominal fillers that were identified to be verb arguments of relevant frequency. For comparative reasons, the cluster analysis was conducted in two independent subtasks.

3.1.1 Noun-clustering with the CCDB

First, the nouns were weighted by the TF-IDF measure and classified by a cluster analysis carried out on a matrix of similarity values taken from the co-occurrence data bank (CCDB) of the Institut für Deutsche Sprache Mannheim (IDS). On the matrix of the similarity values, a k-means cluster analysis was carried out. According to the *Bayesian Information Criterion* there are three optimal noun classes for subjects and prepositional objects and five for direct objects. This result was confirmed by inspecting the within variance of the resulting clusters.

$$\vec{v} = \begin{pmatrix} wn_1c_1 \\ wn_2c_1 \\ . \\ . \\ . \\ wn_n\ c_1 \\ wn_1\ c_2 \\ wn_2\ c_2 \\ . \\ . \\ . \\ wn_nv_n \end{pmatrix}$$

$(wn_i\ c_j\text{: } Weight\ of\ noun\ n_i\ in\ noun\ class\ \ c_j\ )$

Figure 2. ***Dimensions of verb vectors*: *Weighted verbs in noun class areas.***

The verbs' vectors consist of areas for each argument type that is, three areas for argument types in total and each area is split into areas for each noun class as depicted in *figure 2*.

3.1.2 Noun-clustering in Tesla

Additionally, we set up a workflow for noun clustering in Tesla[2]. Here, we computed co-

[2] Tesla (**T**ext **E**ngineering **S**oftware **LA**boratory, see \url: http://tesla.spinfo.uni-koeln.de) is an open source virtual research environment, integrating both a visual editor for conducting text-engineering experiments and a Java IDE for developing software components.

occurrence vectors based on a subset of the *SdeWaC* corpus, containing about 3000 sentences for each noun (**step 5**). For feature selection we used the simple frequency-based heuristics described in Levy & Bullinaria (2004), taking the *k* most frequent types of our corpus as vector features. The vectors were computed in three different configurations. As a baseline, we first took the 200 most frequently occurring elements (mostly closed class function words such as *und* 'and', *zu* 'to', *weil* 'because / since, etc.), and a context window of size 1, accepting only the direct neighbors as co-occurrences. In the second configuration, co-occurrences were computed against the 2000 most frequently occurring elements within a fixed context window of 5 items to both sides. In addition, we employed a positional weighting scheme using the HAL model (Hyperspace Analogue to Language, see Burgess & Lund 1996). In a third test, we took the 10.000 most frequently occurring words and a window size of 10 words, again using the HAL-weighting scheme. While the restriction to function words within a narrow window mainly reflects grammar-related distributional properties, the consideration of content words in combination with a broader window and position weighting emphasizes the more semantically oriented aspects of their distribution. The resulting vectors were weighted by the TF-IDF measure and passed to the cluster analysis **(step 6)**. The corresponding software component workflow as conducted in Tesla is shown in *figure 3*.

For cluster analysis we used three different clusterer implementations adopted from the *ELKI Data Mining API*,[3] namely KmeansLloyd, KmeansMacQueen, and KMedoidsEM, with cluster sizes of k=3 for subjects and prepositional objects and k=5 for direct objects, which we adopted from the CCDB cluster task (see above). For evaluation, we additionally processed all nouns with k=10, resulting in a total of 18 experiments (3 vector configurations for 3 clusterers with 2 different configurations). Finally, for each experiment the resulting word-cluster-pairs were transformed into verb vectors (see next section) and passed to the following processing steps.

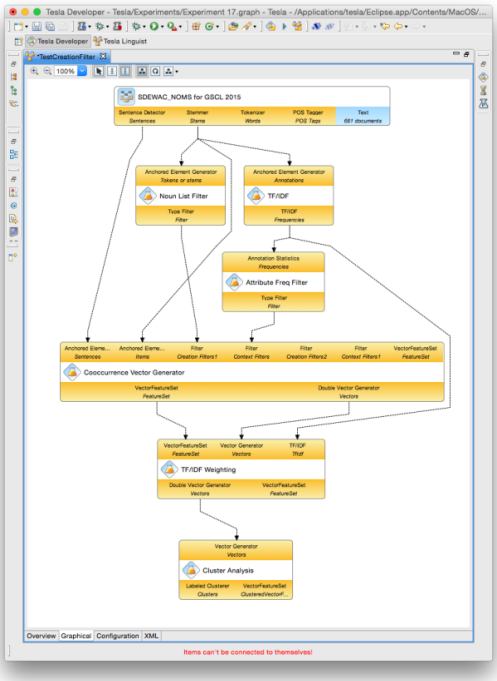


Figure 3. ***Cluster analysis workflow in Tesla. The Clusterers operate on TF-IDF-weighted co-occurrence vectors, computed on a subset of the SdeWaC corpus.***

3.1.3 Generating the Verb Vectors

Within step **(7)** the verb vectors resulting from step **(3)** were re-assembled using the generated noun classes from both steps **(4)** and **(6)**. Hereby the verb vectors could be reduced to 20 (nine aspectual features plus three features for each the subject and prepositional object clusters plus five direct object clusters), respective to 39 dimensions (nine aspectual features complemented by ten clusters for each argument position).

3.2 Classification

For the classification of the 95 verbs **(8)** we used a SVM classifier with a non-linear kernel. The identified arguments and the aspectual features were aggregated for each verb type and represented in feature vectors. For 35 verbs we adapted the classification to the five aspectual verb classes defined in Richter and van Hout (2015), the remaining verbs were manually classified into the five aspectual classes. We trained the SVM using this aspectual classification as training data and tested it with a 10-fold cross-validation. We give some examples of the verb classes of the aspectual gold standard classification below. Note that this classification uses Vendler's definitions (four verb classes), with

[3] The open source framework ELKI (**E**nvironment for Developing **K**DD-Applications Supported by **I**ndex-Structures) was developed at the LMU Munich, see \url: http://elki.dbs.ifi.lmu.de.

the addition of the separate sub-class of accomplishment verbs:

1. **accomplishments**: *aufbauen auf* ('to build on / to be based on'), *herstellen* ('to produce'), schneiden ('to cut'), *zersägen* ('to saw into pieces'), *verlängern* ('to extend'), *mitteilen* ('to tell / to inform'), *übermitteln* ('to communicate / to forward'), *verhindern* ('to prevent'), *abgrenzen* ('mark off / to define'), *verändern* ('to change')
2. **accomplishments with affected subject:** *untersuchen* ('to examine'), *bedenken* ('to consider'), *erörtern* ('to debate'), *nachprüfen* ('to ascertain / to check'), *aufessen* ('to eat up'), *essen* ('to eat'), *beachten* ('to note'), *kaufen* ('to buy')
3. **activities**: *laufen* ('to walk / to run'), *eingehen auf* ('to respond to so. / sth.'), *hämmern* ('to hammer'), *ansteigen* ('to increase'), *fallen* ('to fall'), *denken* ('to think'), *stattfinden* ('to take place'), *wachsen* ('to grow')
4. **achievements:** *einschlafen* ('to fall asleep'), *vergehen* ('to go (by) / to pass / to diasappear'), *übersehen* ('to overlook'), *verlieren* ('to lose'), *anfangen* ('to begin'), *abweichen* ('to deviate'), *sich orientieren an* ('to be geared to'), *richten auf* ('to direct towards / to focus')
5. **states**: *existieren* ('to exist'), *fehlen* ('to lack'), *müssen* ('to must'), *halten für* ('to take so. / sth. for so. / sth.'), *folgen aus* ('to follow from'), *angehören* ('to belong to'), *übereinstimmen* ('to agree'), *betreffen* ('to concern'), *abweichen* ('to deviate') , *verhindern* ('to prevent'), *sein* ('to be'), *vorherrschen* ('to predominate')

4 Results

In order to evaluate the consistency of the comparisons of the classifications against the gold standard we calculated both accuracy and Cohen's kappa. The latter measure considers the number of classes and gives the significance levels.

As a first step, we compared the results from the classifications based on different approaches of clustering arguments for the verb vectors (see *Table 1*). The noun cluster method in the first column specifies

1. the selected data source (CCDB vs. SdeWaC)
2. the number of noun clusters for each argument position
3. the cluster method (K-Medoids vs. K-Means Lloyd vs. K-Means MacQueen)
4. the number of features for each noun (l, 200 vs. 2000 vs. 10000)
5. the context window for co-occurences (win, 1 vs. 5 vs. 10)
6. the quantification of dimensions (count every token vs. count only one token per noun type)

Taking the classification with five aspectual verb classes as the gold standard, ten noun classes per argument position clearly outperform the approaches with fewer features. Additionally, counting every noun token leads to better results than counting only the noun types. Medium length vectors (2000 dimensions), constructed on the basis of a medium context width (window size of five elements) achieve the best outcomes. Specifically, the verb vectors of the 'SdeWaC 10 clusters kmeansLloyd l2k win5 tokens'-noun clustering show the best performance. *Figure 4* depicts the accuracy of combinations of features and is subject of the following result description.

Feature combinations exclusively comprising aspect yield high accuracy values. The combinations aspect-subject - direct-object and aspect-subject - direct-object - prepositional-object outperform the remaining feature combinations with .95 accuracy, **κ** = .93, and .94 accuracy, **κ** = .90, respectively. Kappa values above .81 are characterized as almost perfect agreement and therefore highly significant. The feature combinations aspect- direct object-prepositional object with .88 accuracy, **κ** = .84, and aspect-prepositional object with .86 accuracy, **κ** = .81, achieve also almost perfect agreements. Substantial agreements, that is, above .61, can be observed with the combinations aspect-subject-prepositional object, .84 accuracy, **κ** = .78, aspect-direct object, .92 accuracy, **κ** = .75, and aspect-subject, .82 accuracy**, κ** = .75. Considering the feature combinations without aspectual features, only the combination subject-prepositional

object achieves a satisfactory result with 62 accuracy, **κ** = .43, which is a moderate agreement.

The single features achieve only fair agreements: Aspect achieves .57 accuracy, **κ** = .34, subject achieves .52, **κ** = .27, and direct object and prepositional object achieve .51 accuracy and **κ** = .24 each,

Noun cluster method	**acc**	**K**
CCDB, 3-5-3 clusters, kmeans, tokens	.76	.71
CCDB, 3-5-3 clusters, kmeans, types	.64	.41
CCDB, 10 clusters, kmeans, tokens	**.93**	**.90**
CCDB, 10 clusters, kmeans, types	.60	.39
SdeWaC, 3-5-3 clusters, kmediods l2k, win5, tokens	.85	.80
SdeWaC, 10 cluster kmediods l10k win10 tokens	.85	.80
SdeWaC, 10 cluster kmediods l200 win1 tokens	.86	.83
SdeWaC, 10 cluster kmediods l2000 win5 tokens	**.93**	**.90**
SdeWaC, 10 cluster kmediods l2000 win5 types	.69	.48
SdeWaC, 3/5cluster kmediods l2000 win5 tokens	.85	.80
SdeWaC, 10 clusters, kmeansLloyd l10k win10 tokens	.88	.84
SdeWaC, 10 clusters kmeansLloyd l200 win1 tokens	.88	.84
SdeWaC, 10 clusters kmeansLloyd l2k win5 tokens	**.94**	**.93**
SdeWaC, 10 clusters kmeansMacQ l10k win10 tokens	.89	.86
SdeWaC, 10 cluster kmeansMacQ l200 win1 tokens	.92	.88
SdeWaC, 10 cluster kmeansMacQ l2k win5 tokens	.90	.90

Table 1. ***Evaluation of different approaches to construct the verb vectors of the verbs.***

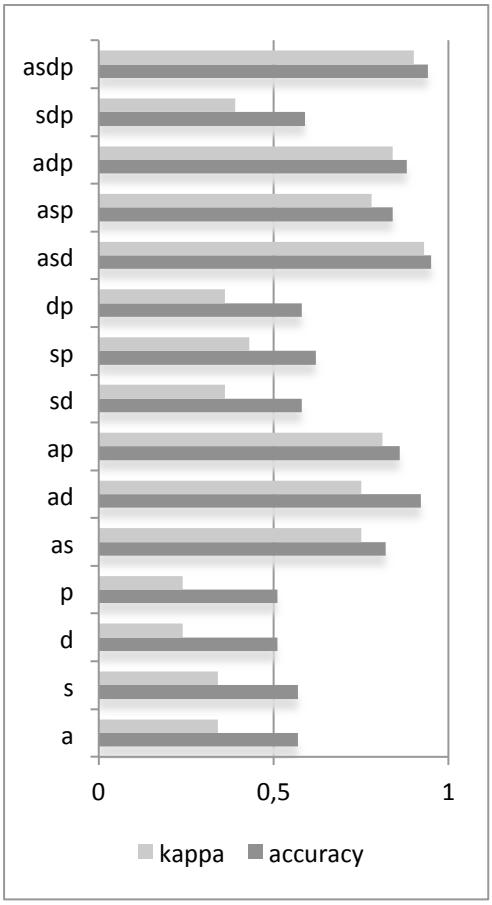


Figure 4 ***Accuracy of the argument and aspectual features from the best performing noun cluster method using five aspectual verb classes as gold standard.***

Legend: s: subject, d: direct object, p: prepositional object, a: aspectual features and combinations of predictors, for instance, da: direct object and aspect, sp: subject and prepositional object.

5 Conclusion

The study provides evidence for the hypothesis that the Vendlerian aspectual verb classes (plus a class of *accomplishments with an affected subject*) can be induced from classified nominal fillers in argument positions in combination with aspectual features in dependent or governing positions. In contrast to the study of Richter and Hermes (2015), which identified the subject as the feature with the highest predictive power, this study reveals that combinations of features comprising aspect features clearly outperform the remaining feature combinations which do not include aspect features, and also outperform the single features, that is, aspect, subject, direct object and prepositional object. This result could be observed in all clustering approaches, which, without exception, attest the high predictive

power of feature combinations comprising aspect. Aspectual features as singletons are not sufficient to predict aspectual verb classes.

In addition, it could be observed that a higher number of noun clusters in the argument features improves the quality of classifications. Compared to the 3-5-3 noun clustering (that is, 3 noun classes in the subject position, 5 noun classes in the object position and 3 noun classes in the prepositional object position), the 10-10-10 noun classes combinations achieve much better results. We draw the conclusion that the higher the number of noun classes, the greater the discriminating effect, presumably because a more finely grained distinction of semantic properties is achieved. This also holds for the comparison of the classification quality of verb vectors containing noun tokens with vectors containing noun types. The number of verb tokens is, by definition, considerably higher than that of the types, and we observed that with regard to the classification quality, verb vectors with noun tokens clearly outperform vectors with noun types. Finally we discovered that medium length vectors based on a medium context frame yield the best results.

All in all, the results of this study show that: 1. aspectual verb classes can be empirically validated, 2. classified nouns in argument positions in combination with aspectual features are reliable predictors of aspectual verb classes, i.e. the meaning of nouns (and noun classes, respectively) correlates with aspectual parts of the verbal meaning.

References

Judith Aissen. 2003. *Differential Object Marking: Iconicity vs. economy*. In: *Natural Language and Linguistic Theory*, 21:435–483.

Bernd Bohnet. 2010. *Very High Accuracy and Fast Dependency Parsing is not a Contradiction.* In: The 23rd International Conference on Computational Linguistics *(COLING 2010),* Beijing, China.

William Croft. 2003. *Typology and Universals*. Cambridge University Press (2nd edition).

Tim Fernando. 2004. *A finite-state Approach to Events in Natural Language Semantics*. Journal of Logic and Computation, 14(1):79-92.

Bonnie J. Dorr and Doug Jones. 1996. *Role of Word Sense Disambiguation in Lexical Acquisition: Predicting Semantics from Syntactic Cues*. In: *Proceedings of the 16th International Conference on Computational Linguistics*, 322–327. Copenhagen.

George W. Furnas, Thomas K. Landauer, Louis M. Gomez and Susan T. Dumais. 1983. *Statistical semantics: Analysis of the potential performance of keyword information systems.* Bell System Technical Journal, 62(6):1753–1806.

Sabine Gründer. 2008. *An Algorithm from Adverbial Aspect Shift.* In: *Proceedings of the 22nd International Conference on Computer Linguistics* (Coling08), 289–296.

Eric Joanis, Suzanne Stevenson and David James. 2008. *A General Feature Space for Automatic Verb Classification.* Natural Language Engineering 14(3):337-367.

Judith L. Klavans and Martin Chodorow. 1992. *Degrees of Stativity: The lexical representation of verb aspect.* In: *Proceedings of the 14th International Conference on Computational Linguistics.*

Wolfgang Klein. 2009. *How time is encoded*. In: Wolfgang Klein and Ping Li (eds.), *The expression of time*, 39–82. Mouton de Gruyter, Berlin.

Thomas K. Landauer and Susan T. Dumais. 1997. *A solution to Plato's problem: The Latent Semanctic Analysis theory of the acquisition, induction, and representation of knowledge*. Psychological Review, 104:211–140.

Joe P. Levy and John A. Bullinaria. 2001. *'Learing lexical properties from word usage patterns: Which context words should be used?' Connectionist models of learning, development and evolution,* 273-282. Springer, London.

Beth Levin. 1993. *English Verb Classes and Alternations*. The University of Chicago Press, Chicago.

Kevin Lund and Curt Burgess. 1996. *Hyperspace analogue to language (HAL): A general model semantic representation.* In: *Brain and Cognition 30*.

Paola Merlo and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. In: *Computational Linguistics*, 27(3):373-408.

Patrick Pantel. 2005. Inducing ontological co-occurrence vectors. In *Proceedings of Association for Computational Linguistics* (ACL-05):125–132.

Christopher Parisien and Suzanne Stevenson. 2011. *Generalizing between form and meaning using learned verb classes*. In: *Proceed-*

ings of the 33rd Annual Conference of the Cognitive Science Society Boston, Massachusetts.

J. Preiss, T. Briscoe, and A. Korhonen. 2007. *A System for Large-Scale Acquisition of Verbal, Nominal and Adjectival Subcategorization Frames from Corpora*. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics,* 912–919, Prague.

Michael Richter and Jürgen Hermes. 2015. *Classification of German verbs using nouns in argument positions and aspectual features.* In: Vito Pirrelli, Claudia Marzi and Marcello Ferro (eds.) *Word Structure and Word Usage*. *Proceedings of the NetWordS Final Conference*, 177–181, Pisa.

Michael Richter and Roeland van Hout. 2015. *A classification of German verbs using empirical language data and concepts of Vendler and Dowty.* To appear in 2015. In: *Sprache und Datenverarbeitung – International Journal for Language Data Processing.*

Herbert Rubenstein and John B. Goodenough. 1965. *Contextual correlates of synonomy.* In: *Communications of the ACM,* Vol 8, (10):627–633.

Sabine Schulte im Walde and Chris Brew. 2002. *Inducing German Semantic Verb Classes from Purely Syntactic Subcategorisation Information.* In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 223–230. Philadelphia.

Sabine Schulte im Walde. 2003. *Experiments on the Choice of Features for Learning VerbClasses*. In: *Proceedings of the 10th Conference of the European Chapter of the Association for computational Linguistics*, 315–322. Budapest.

Sabine Schulte im Walde. 2004. *Automatic Induction of Semantic Classes for German Verbs*. In: Stefan Langer and Daniel Schnorbusch (eds.), *Semantik im Lexikon*. 59–86. Gunter Narr Verlag, Tübingen.

Eric V. Siegel. 1997. *Learning methods for combining linguistic indicators to classify verbs*. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*.

Eric V. Siegel and Kathleen R. McKeown. 2000. *Learning Methods to Combine Linguistic Indicators: Improving Aspectual Classification and Revealing Linguistic Insights*. Computational Linguistics, 595–627.

Helmut Schumacher. 1986. *Verben in Feldern*. V*alenzwörterbuch zur Syntax und Semantik deutscher Verben.* De Gruyter, Berlin & New York.

Hinrich Schütze and Jan Pedersen. 1993. *A vector model for syntagmatic and paradigmatic relatedness.* In: *Making Sense of Words:* 104–113. Ninth Annual Conference of the UW Centre for the New OED and Text Research, Oxford.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Zeno Vendler. 1967. *Linguistics in Philosophy*. Ithaka / New York: Cornell University Press.

Andreas Vlachos, Anna Korhonen and Zoubin Ghahramani. 2009. *Unsupervised and constrained Dirichlet process mixture models for verb clustering'*. In: *Proceedings of the EACL Workshop on Geometrical Models of Natural Language Semantics*, 74–82. Athen.

Warren Weaver. 1955. *Translation*. In: William N. Locke and A. Donald Booth, *Machine Translation of Languages*: 15–23. MIT Press, Cambridge Mass.

Correlation between Lexical and Determination Types

Oliver Hellwig
University of Düsseldorf
ohellwig@phil-fak.uni-duesseldorf.de

Wiebke Petersen
University of Düsseldorf
petersen@phil.uni-duesseldorf.de

Abstract

The Paper presents a corpus based study of Löbner's theory of determination types. According to Löbner, nouns prefer a syntactic determination mode that is congruent to their inherent type of lexical determination. The study applies machine learning methods to a large corpus of German texts for detecting the determination modes of nouns automatically and addresses the problem of inducing the lexical determination type of a noun from its distributional fingerprint of occurrences in different determination modes

1 Introduction and previous research

The paper aims at assessing the theory of congruent determination (Löbner, 2011) by applying methods from Computational Linguistics to a corpus of German sentences. Löbner claims that the lexical semantics of a noun influences its syntactic mode of determination. He distinguishes two binary semantic features that result in a fourfold classification of nouns; each class corresponds to a 'natural' mode of determination. The two binary features are uniqueness ($\pm U$) and relationality ($\pm R$). Inherently unique nouns refer to a single, unique referent (e.g., 'sun' or 'steer' in contrast to 'star' or 'wheel'). The reference of an inherently relational noun depends on an additional possessor argument which needs to be specified (e.g., 'surface [of something]', 'brother [of someone]' in contrast to 'sun' or 'stone'). Combining the two binary features results in four lexical determination types of nouns: *Sortal* nouns describe non-unique, non-relational referents ($-U-R$, 'stone', 'chair'); *individual* nouns describe non-relational, but unique referents ($+U-R$, 'sun', 'Peter'), *relational* nouns describe non-unique referents that require a possessor ($-U+R$, 'brother', 'leaf'), and *functional* nouns describe unique, relational referents ($+U+R$, 'steer', 'trunk'). The four basic lexical noun types are summarized in Table 1.

In usage, nouns often occur in determination modes that are not congruent with their underlying lexical types. Löbner (2011) claims that incongruent determination is a marked option and involves a type shift of the noun in the sense of Partee (1986). For example, if a sortal noun is used in a context in which it is uniquely referring for pragmatic reasons, the noun is shifted from $-U$ to $+U$, as in 'the train arrives' when someone waits at the station. The lexical distinction in $\pm U$ nouns and uniqueness shifts are clearly marked in languages with a split article system that distinguish between strong and weak definite articles. In contrast to standard German, for example, the Ripuarian dialect spoken in the Rhineland makes this distinction. The weak definite article 'dr' is used with inherently $+U$ nouns. If a non-unique noun ($-U$) is used in a definite context, the incongruent use is marked by the strong definite article 'dä':

(1) Ripuarian [cf. Löbner (2011)]:
   a. Dr Zoch$_{ind.}$ kütt.
      DEFARTWEAK parade comes.
   b. Dä Zoch$_{sort.}$ kütt.
      DEFARTSTRONG train comes.

The sentences (1-a) and (1-b) only differ with respect to the definite article being used. While in (1-b) 'Zoch' is read as a sortal noun ('train'), in (1-a) 'Zoch' refers to the carnival parade, which is conceptualized as a unique individual in cities like Cologne or Düsseldorf.

The lexical $\pm R$ distinction and type shifts along this dimension are less frequently marked in languages. However, languages with an overt morphology for (in)alienability mark shifts from $-R$ to $+R$ (Ortmann, 2015):

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 130–137, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

	non-unique reference [-U]	unique reference [+U]
non-relational [-R]	**sortal noun** tree, stone, woman	**individual noun** pope, universe, Mary
relational [+R]	**proper relational noun** sister [of], student [of], page [of]	**functional noun** mother [of], dean [of], cover [of]

Table 1: The four basic lexical noun types according to Löbner (2011)

(2) Diegueño [after Nichols (1992)]

a. *ʔ–əɬaly*
1sg–mother
'my mother'

b. *ʔ–ə^n–ewa*
1sg–poss–house
'my house'

In Diegueño, for example, the word 'ewa' ('house') is conceptualized as a sortal noun ($-R$). In order to be used in a possessive context, it has to be shifted by a possessive marker to $+R$. Other languages like Yucatec indicate by a derelativizing marker shifts from $+R$ to $-R$.[1]

The $\pm R$ distinction also plays a crucial role in formal compositional approaches to the semantics of genitive constructions. In this context, the question arises whether in constructions such as 'Mia's sister' and 'Mia's pen' the possessor 'Mia' is an argument or a modifier of the head noun. Previous research has given three different answers to this question: (1) the possessor is always an argument of the head noun, which is shifted to $+R$ if it is lexically non-relational (Vikner and Jensen, 2002); (2) the possessor is always a modifier and there are no genuine $+R$ nouns in the lexicon; and (3) whether the possessor acts as an argument or as a modifier depends on its underlying lexical $\pm R$ feature (Barker, 1995; Barker, 2011). Partee and Borschev (2003) critically discuss all three options, discard the second and come to the conclusion that the interpretation of genitive constructions is determined by the lexical $\pm R$ feature of a noun.

The discussed examples give evidence for Löbner's fourfold noun classification from a typological perspective. However, Löbner (2011) argues that lexical $\pm U$ and $\pm R$ distinctions and shifts between the determination types are also relevant for languages that do not mark lexical types and shifts overtly. He claims that the 'natural' unshifted mode of determination – that means the determination mode that is congruent with the lexical type of determination – influences the relative frequencies of the different determination modes in which a noun occurs. He reports statistical evidence for his claim from studies for Swedish (Fraurud, 1990) and English (Vieira, 1998; Nissim, 2004; Jensen and Vikner, 2004). However, these studies concentrate only on either the $\pm U$ or the $\pm R$ distinction. A first statistical investigation that combines both features has been carried out in Horn and Kimm (2014) for German using a small hand annotated corpus consisting of two fictional short stories with only 456 noun tokens. The study shows that for both features, $\pm U$ and $\pm R$, the congruent determination is more frequent than the incongruent one.

[1] An investigation on further typological evidence for type shifts can be found in Ortmann (2015).

Det$_{+U}$ determination mode
singular definite ('die Sonne', ['the sun']; 'Mia')
contracted singular definite ('zur Sonne', ['to the sun'])
possessive pronoun ('mein Kopf', ['my head'])
left genitives ('Mias Kopf', ['Mia's head'])
Det$_{-U}$ determination mode
indefinite article ('ein Stein' ['a stone']),
plural ('(die) Steine' ['(the) stones']),
quantifiers ('jeder/kein/einige/beide/zwei Stein(e)', ['every/no/some/both/two stone(s)]')
contrastive demonstrative ('dieser/jener Stein, ['this/that stone']),
interrogative ('welcher Stein', ['which stone'])
Det$_{+R}$ determination mode
possessive pronoun ('mein Kopf', ['my head'])
left genitive ('Mias Kopf', ['Mia's head')
prepositional phrase ('der Kopf von Mia', ['the head of Mia'])
Det$_{-R}$ determination mode
absolute use

Table 2: Investigated modes of determination in German

This paper aims at testing the hypothesis of congruent determination for German with a large amount of data. In parallel to the lexical features $\pm U$ and $\pm R$, we use the features DET$_{\pm U}$ and DET$_{\pm R}$ in order to distinguish the determination modes in actual language use. Table 2 shows a selection of different modes of determination in German with their respective features based on Löbner (2011). In order to avoid tedious manual annotations of actual uses, labelers for DET$_{\pm U}$ and DET$_{\pm R}$ are developed and applied to a corpus of German news texts. The syntactic determination

mode of each noun token is derived from the results of the labelers. For each noun lemma, its determination modes are cross-tabulated with its lexical semantic determination type. The semantic types are provided by a manual annotation of the $\pm U$ and $\pm R$ features of frequent nouns drawn from a German dictionary (Duden, 1997).

The rest of the paper is structured as follows. In Section 2, we describe the two labelers for $\text{DET}_{\pm U}$ and $\text{DET}_{\pm R}$, whereby the $\text{DET}_{\pm R}$-labeler is an enhancement of the work presented in Hellwig and Petersen (2014). Section 3 investigates the correlation between the lexical determination type and determination modes in language use for a group of high-frequency lemmata, thereby providing an estimation for the frequency of incongruent determination in German.

2 Automatic annotation of types of determination modes

The automatically annotated modes of determination will be cross-tabulated with the lexical types of determination in order to investigate whether the latter influences the former. Therefore, automatic $\text{DET}_{\pm U}$ and $\text{DET}_{\pm R}$ classifiers are required that have a high precision. Recall is less important for our purpose, as long as the classifiers do not systematically overlook a particular mode of determination.

2.1 Detecting unique reference ($\text{DET}_{\pm U}$)

Due to the German article system, detecting the uniqueness of reference is a comparatively easy task that can be performed with a rule-based approach. The rules used in our $\text{DET}_{\pm U}$ classifier rely on the POS information that is produced by the MATE parser (Bohnet and Nivre, 2012) and the Stanford NLP parser for German (Rafferty and Manning, 2008). Additionally, morphological information about number created by MATE is taken into account. The labeling rules detect the modes of determination as given in Table 2. Nouns in singular number that are directly preceded by (contracted) definite articles, possessive pronouns or another noun in genitive case are labeled as $+U$. Plural nouns or nouns in singular number that are preceded by a member from a finite list of function words ("jeder" ['each'], "solcher" ['such'],"ein" ['a'], "dieser" ['this'],...) are labeled as $-U$. All other nouns are labeled as undetermined with regard to the uniqueness of their referents.

Type	P	R	F
SVM^{HMM}	92.76	75.7	83.37
CRF	92.23	71.68	80.66
ME	93.97	64.83	76.73
Tree	67.14	84.86	74.97

Table 3: Word-based evaluation by classifier for the class DET_{+R}, 30-fold cross-validation, no additional training data

Type	P	R	F
SVM^{HMM}	99.06	99.77	99.42
CRF	98.9	99.77	99.33
ME	98.63	99.84	99.23
Tree	99.4	98.38	98.89

Table 4: Word-based evaluation by classifier for the class DET_{-R}; same settings as in Table 3

2.2 Detecting relationality ($\text{DET}_{\pm R}$)

The $\text{DET}_{\pm R}$ classifier builds on the relation detection classifier described in Hellwig and Petersen (2014), which performs a three-class labeling task with classes POR (noun in possessor position), PUM (noun in possessum position), and no-poss. For our purpose, we can reduce the task to a binary classification problem for the features $\text{DET}_{\pm R}$ by merging the classes POR and no-poss to DET_{-R} and identifying the class PUM as DET_{+R}. In Hellwig and Petersen (2014) three statistical classification methods are used that are able to handle sequential data from a categorical scale: Hidden Markov Support Vector Machines (SVM^{HMM}) (Altun et al., 2003), Conditional Random Fields (CRF) (Lafferty et al., 2001), and Maximum Entropy (ME) (Ratnaparkhi, 1998). Additionally, Hellwig & Petersen experiment with a rule-based tree classifier that shows low precision, but rather good recall scores. Table 3 and Table 4 show the results of the binary classification for 30 cross-validations. Due to the reduced complexity of the task, the results for DET_{+R} are slightly better than those reported for PUM in Hellwig and Petersen (2014). The results in Table 4 demonstrate that reliable results should be expected for DET_{-R} determination.

We test three strategies to improve over the baseline results for DET_{+R} that are shown in Table 3. In the first approach, we postprocess the output of the four basic classifiers with another nonlinear classifier. For this sake, the symbolic outputs of SVM^{HMM} and of the tree classifier are transformed into the pseudo-confidence values of 1.0

Meta-classifier	P	R	F
Logistic regression	94.70	74.87	83.63
Neural network	94.46	75.76	84.08

Table 5: Results of post-processing the classifications with meta-learners; same settings as in Table 3

Voting	P	R	F
3-majority	95.23	73.48	82.95
4-majority	99.27	51.78	68.06

Table 6: Results of combining single classifiers by majority voting; same settings as in Table 3

for the class DET_{+R} and 0.0 for the class DET_{-R}, and the confidence values generated by CRF and ME are added to this feature vector. The resulting four-dimensional vectors, which consist of pseudo-confidences for SVM^{HMM} and the tree classifier and of true confidence values for CRF and ME, are used for training (a) a logistic regression and (b) a neural network with one hidden layer of dimension 2. Table 5 records the results of applying these two meta-learners. Both sets of results don't differ significantly from the best single classifier (refer to Table 3).

In the second improvement approach, we apply semi-supervised learning to the binary classification problem. First, we train all classifiers with the full set of gold data used in Hellwig and Petersen (2014), which consists of approximately 1.100 manually annotated sentences. Next, we use the trained classifiers to annotate a holdout-part of the new unannotated corpus (1M sentence extract from the Leipzig corpora, 2005) without supervision. Because SVM^{HMM}, CRF, ME, and the tree classifier can provide different decisions for each word in the holdout set, we merge their decisions for each word using majority voting. The majority voter is expected to produce accurate results, and not to be biased towards certain types of relational constructions. We test majority voting with a winning majority of three and of four classifiers (results in Table 6). While the result for 3-majority is close to the baseline values shown in Table 3, the 4-majority produces a high accuracy along with a rather low recall. The high precision of the 4-majority is basically desirable for our purposes, as long as the low recall of approximately 52% does not point to a bias in the voting process. Therefore, we study the effect of majority voting on the distribution of the different relational determination modes (see Table 2) of DET_{+R} records from the gold data. When comparing the frequencies of relational types in these samples with their frequencies in the full gold data, a χ^2 test yields a p value of 0.3533 for 3-majority, but of $p = 0.0000056$ for 4-majority voting with notable frequency differences for the determination modes 'possessive pronoun' and 'prepositional phrase with *von*'. To avoid the bias inherent in the 4-majority voting process, we build a new additional training set T^*. This set consists only of those sentences from the holdout set in which 3-majority votes were obtained for all words. When retraining the classifiers, the training set for each fold n consists of the respective $(n-1)$ parts of the gold data and the full set T^*. Table 7 shows that the retraining slightly improves the F score of SVM^{HMM}, but produces a lower recall for ME when compared with the data in Table 3.

Type	P	R	F
SVM^{HMM}	93.26	76.97	84.34
CRF	92.19	71.8	80.73
ME	95.17	53.56	68.54

Table 7: Performance of the classifiers after semi-supervised retraining with the gold data and with 15.326 silver-annotated sentences (10.330 PUM annotations, 3-majority); 30 cross-validations. Cmp. with Table 3

In the third improvement approach, we use neural embeddings of the context words instead of their sparse 1-of-n encodings, and train a neural network on the given binary classification task. We test this approach because the possessive constructions studied in this paper can be interpreted as a kind of frame semantic relation, though on a rather abstract level. Mesnil et al. (2015) report about significant improvements when a combination of recurrent neural networks and pretrained neural word embeddings is applied to a simple form of frame semantic labeling (i.e., slot filling). We use a 750 MB corpus of German newspaper texts to train the neural embeddings. The newspaper corpus is lemmatized using MATE in order to reduce the size of the vocabulary, and then fed into the word2vec software (Mikolov et al., 2013).[2] The input features for the neural network are constructed as follows. For each word w_j, we build

[2]Parameters: hidden size: 300, window size: 5, training iterations: 10.

a vector of subfeatures by concatenating (1) its neural embedding of size 300 that is generated by word2vec, (2) its case information according to MATE, (3) its POS tag according to MATE, and (4) a Boolean flag indicating if the word is terminated by the letter s (which is a frequent genitive ending in German). Features (2) and (3) are added in 1-of-n encodings after removing POS tags that occur less than 100 times in the training corpus. For each word w_i to be classified, these features are collected for w_{i-1}, w_i, and w_{i+1}, and concatenated to form the input vector of the neural network. The resulting concatenated vector is fed into an Elman network.[3] A cross-validation with 30 folds using the same data as for Table 3 and Table 7 produces $P = 79.29, R = 16.67, F = 27.54$. Obviously, recurrent neural networks combined with pretrained neural word embeddings are not really helpful for the given task. This conclusion is supported by a closer inspection of the parameters learned by the CRF model. The highest scoring parameters are induced by rules that operate almost exclusively on POS tags, case information for different context words, or combinations of POS and case information.[4] A full deep-learning pipeline will certainly generate different and better results. However, the problem at hand may lend itself to a shallow solution that is focused on morpho-syntactic features.

None of the three improvement strategies examined in this section substantially improves the performance of the $\text{DET}_{\pm R}$ labeler, while they increase the complexity of the classification workflow at the same time. Therefore, we decided to use the baseline system with binary output for the labeling task described in the next section.

3 Results

The paper aims at examining if and how the syntactic determination modes of nouns in language use are correlated with their semantic lexical determination types. The preceding section has described the classifiers that are used to detect the different syntactic determination modes automatically. This section introduces the data source for the lexical determination types. In addition, it determines the frequencies of congruent and incongruent determination uses for frequent nouns by contrasting the syntactic determination modes that were detected using the automatic labelers with those of the manually annotated lexical determination types.

3.1 Data for lexical determination types

For information about the lexical determination types we had access to manually annotated lexical data that was originally created for internal use in SFB 991 and that contains approximately 23.000 tokens annotated with concept types. The data consists of noun lemmata that occur at least ten times in a corpus consisting of German news, narrative, and scientific texts. Each noun occurrence is disambiguated with respect to the readings given in the standard German dictionary Duden (1997). Finally, each lexical reading that occurs at least once is classified with respect to its lexical type of determination (see Table 1).

To obtain clearer effects, we have selected only those noun lemmata (1) to which only a single lexical determination type had been assigned, and (2) that occured at least 100 times in our corpus (1M sentence extract from the Leipzig corpora, 2005). This filtering process produced a candidate list L_{cand} of 217 noun lemmata that contained 24 functional, 48 individual, 53 relational, and 92 sortal nouns.

3.2 Congruency of determination

The German 1M sentence extract of the Leipzig corpora from 2005 was processed with the $\text{DET}_{\pm U}$ and $\text{DET}_{\pm R}$ labelers from Section 2, skipping all sentences for which MATE could not detect a single root node. Only those records were retained for which the $\text{DET}_{\pm U}$ labeler assigned either DET_{-U} or DET_{+U}, and for which at least three members of the $\text{DET}_{\pm R}$ classifier ensemble agreed (3-majority). Furthermore, all records whose lemmata were not found in L_{cand} were removed. The resulting list L_{res} contained 433.230 noun tokens for the 217 lemmata from L_{cand}.

Table 8 displays the absolute frequencies from L_{res} for the lexical (rows) determination types and the syntactic (columns) determination modes. The data shows on one hand that within one lexical determination type the congruent determination mode is dominant only for sortal ($-U-R$) and individual ($+U-R$) nouns. Relational ($-U+R$) and functional ($+U+R$) nouns occur more often in DET_{-R} contexts than in DET_{+R}. A first shallow analysis

[3] Architecture: one hidden layer with 100 units, output layer with one unit. Activation functions: Sigmoid for the hidden layer, softmax for the output layer. Sequential gradient descent learning, initial learning rate: 0.005. A validation set was used for early interruption of the learning process.

[4] The five highest scoring rules of the CRF were: pos=N:PUM, m1-pos=PRPOSS:PUM, p1-word=der:PUM, pos=PREP:no-poss, and p1-case=gen:PUM.

	DET_{-U} DET_{-R}	DET_{+U} DET_{-R}	DET_{-U} DET_{+R}	DET_{+U} DET_{+R}
-U-R	**140.706**	87.476	4.716	7.042
+U-R	531	**66.773**	8	58
-U+R	43.504	17.999	**10.243**	5.457
+U+R	15.239	21.264	3.714	**7.674**

Table 8: Absolute frequencies of congruent and incongruent determination types

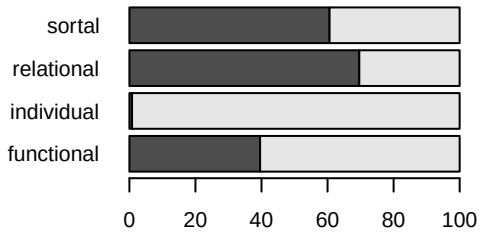


Figure 1: Uses of $DET_{\pm U}$, split up by lexical types. Light grey: Unique uses, dark grey: non-unique uses; x-axis in percent

of the incongruent DET_{-R} uses indicates that the relational argument is frequently implicitly fixed by the context, but not overtly expressed within the noun phrase. On the other hand, the lexical determination type that is congruent with a given determination mode is preferably chosen. There is only one exception to this rule: Among the noun tokens which are used in the DET_{+U-R} determination mode the dominant group is formed by sortal nouns and not by individual nouns with congruent determination type. However, this is due to the fact that sortal nouns are much more frequent than individual nouns.

In addition to the absolute frequencies given in Table 8, the bar plots in Figure 1 and Figure 2 show the percentage distribution of the $DET_{\pm U}$ and $DET_{\pm R}$ features separately for the four lexical determination types. The plots indicate that on average, if one looks at the accumulated frequencies of all nouns belonging to one lexical type, $+U$ nouns show a stronger tendency to occur in DET_{+U} contexts than $-U$ nouns and, $+R$ nouns show a stronger tendency to occur in DET_{+R} contexts than $-R$ nouns. Similar results have already been reported in the smaller manual study (Horn and Kimm, 2014). Thus, there is statistical evidence for the hypothesis that the lexical determination type of a noun influences its use in actual determination modes with a tendency towards congruent determinations.

However, this influence is not strong enough to determine the lexical determination type of a single noun lemma by its frequency distribution over the different types of determination modes in actual language use. Figure 3 shows all 217 lemmata from L_{cand} grouped by their lexical determination types and placed by their individual distributions in $DET_{\pm U}$ / $DET_{\pm R}$ contexts. The plot demonstrates that, apart from the individual nouns, which mainly consist of named entities, all other noun types spread over a rather big region instead of neatly gathering in one corner.

The picture is especially blurred for relational and functional nouns, which occur less frequently in relational determination mode than expected from their underlying lexical types. This result, which partially coincides with the findings reported in Horn and Kimm (2014), points to the influence of other linguistic factors such as anaphoric uses that will be examined in a follow-up study.

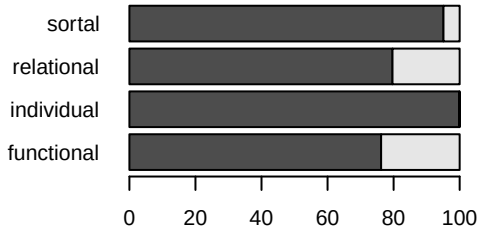


Figure 2: Uses of $DET_{\pm R}$, split up by lexical types. Light grey: Relational uses, dark grey: non-relational uses

Acknowledgement

Research for this paper was funded by a grant of the DFG (SFB 991, project C02).

References

Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Hidden Markov Support Vector Machines. In *Proceedings of the Twentieth International Conference on Machine Learning*.

Chris Barker. 1995. *Possessive Descriptions*. CSLI Publications, Stanford.

Chris Barker. 2011. Possessives and relational nouns. In *Semantics: An International Handbook of Natural Language Meaning*, chapter 45, pages 1108–1129. de Gruyter.

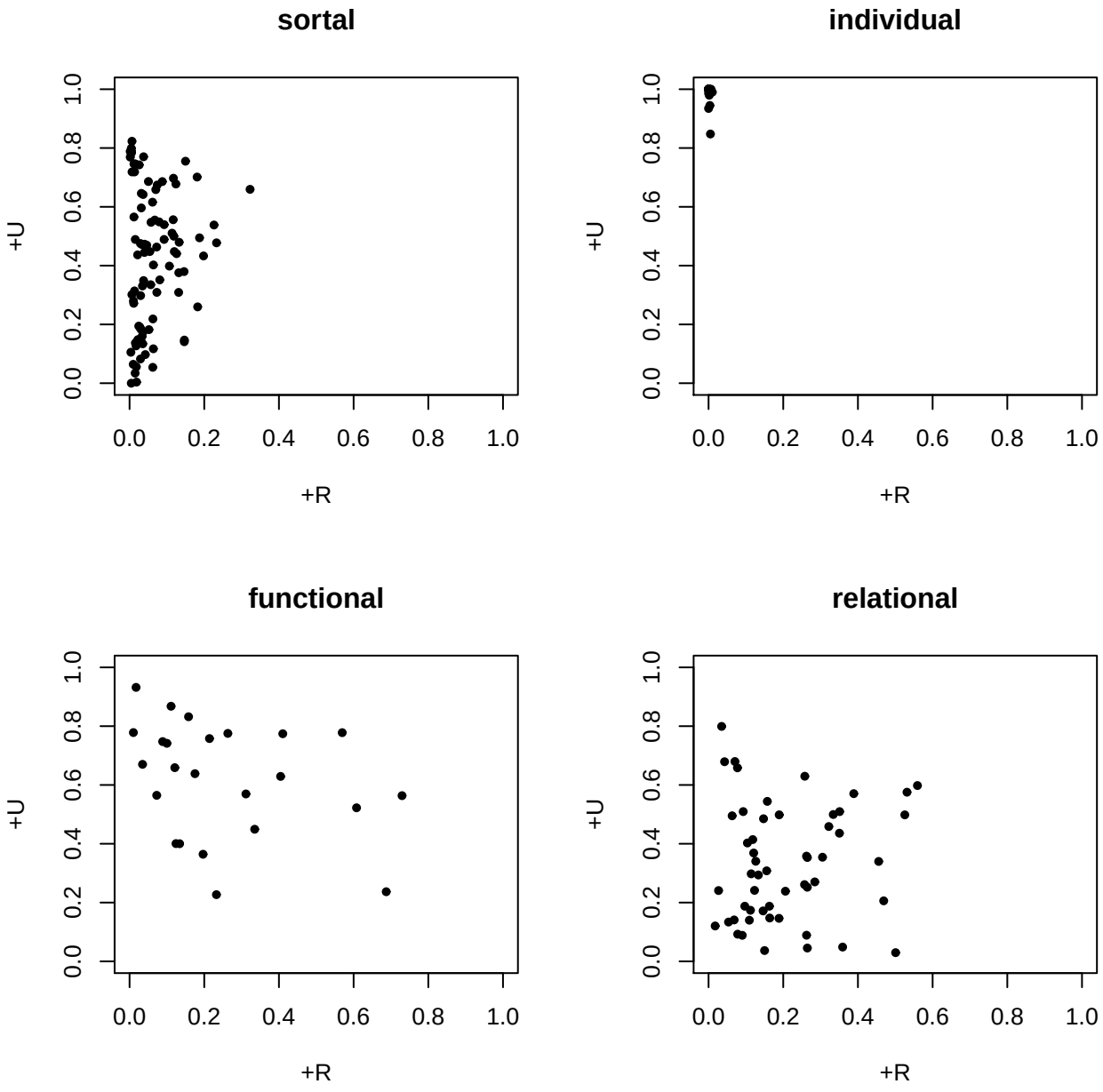


Figure 3: $\mathrm{DET}_{\pm U}$/$\mathrm{DET}_{\pm R}$ distribution plots for 217 lemmata grouped by their lexical determination

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *EMNLP-CoNLL*, pages 1455–1465.

Duden. 1997. *Duden Universalwörterbuch A-Z*. Bibliographisches Institut und Brockhaus, Mannheim, 3 edition. Electronic version.

Kari Fraurud. 1990. Definiteness and the processing of noun phrases in natural discourse. *Journal of Semantics*, (7):395–433.

Oliver Hellwig and Wiebke Petersen. 2014. Detecting relational constructions in German texts automatically. In Josef Ruppenhofer and Gertrud Faaß, editors, *Proceedings of the 12th edition of the KONVENS conference*, pages 40–47.

Christian Horn and Nicolas Kimm. 2014. Nominal concept types in German fictional texts. In Thomas Gamerschlag, Doris Gerland, Rainer Osswald, and Wiebke Petersen, editors, *Frames and Concept Types. Applications in Language and Philosophy*, volume 94 of *Studies in Linguistics and Philosophy*, pages 343–362. Springer Verlag.

Per Anker Jensen and Carl Vikner. 2004. The English pre-nominal genitive and lexical semantics. In J.Y Kim, Y.A. Lander, and B.H. Partee, editors, *Possessives and beyond: semantics and syntax*, number 29 in University of Massachusetts Occasional Papers in Linguistics. GLSA Publisher, Amherst.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Sebastian Löbner. 2011. Concept types and determination. *Journal of Semantics*, 28:1–55.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Johanna Nichols. 1992. *Linguistic Diversity in Space and Time*. The University of Chicago Press, Chicago.

Malvina Nissim. 2004. Lexical information and choice of determiners. In J.Y Kim, Y.A. Lander, and B.H. Partee, editors, *Possessives and beyond: semantics and syntax*, number 29 in University of Massachusetts Occasional Papers in Linguistics, pages 133–152. GLSA Publisher, Amherst.

Albert Ortmann. 2015. Uniqueness and possession: Typological evidence for type shifts in nominal determination. In Martin Aher, Daniel Hole, Emil Jeřábek, and Clemens Kupke, editors, *Logic, Language, and Computation*, volume 8984 of *Theoretical Computer Science and General Issues*, pages 234–256. Springer, Berlin, Heidelberg.

Barbara Partee and Vladimir Borschev. 2003. Genitives, relational nouns, and the argument-modifier ambiguitiy. In Ewald Lang, Claudia Maienborn, and Cathrine Fabricius-Hansen, editors, *Modifying Adjuncts*, pages 67–112. Walter de Gruyter, Berlin.

Barbara Partee. 1986. Noun phrase interpretation and type-shifting principles. In J. Groenendijk, D. de Jongh, and M. Stokhof, editors, *Foundations of pragmatics and lexical semantics*, pages 115–143. Foris, Dordrecht.

Anna N. Rafferty and Christopher D. Manning. 2008. Parsing three German treebanks: Lexicalized and unlexicalized baselines. In *Proceedings of the Workshop on Parsing German*, pages 40–46, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Renata Vieira. 1998. *Definite description processing in unrestricted text*. Ph.D. thesis, University of Edinburgh.

C. Vikner and P. A. Jensen. 2002. A semantic analysis of the English genitive. Interaction of lexical and formal semantics. *Studia Linguistica*, 56(2):191–226.

Digitale Kuratierungstechnologien

Verfahren für die effiziente Verarbeitung, Erstellung und Verteilung qualitativ hochwertiger Medieninhalte

Georg Rehm
DFKI GmbH
Forschungsbereich Sprachtechnologie
Alt-Moabit 91c
10559 Berlin
georg.rehm@dfki.de

Felix Sasaki
DFKI GmbH
Forschungsbereich Sprachtechnologie
Alt-Moabit 91c
10559 Berlin
felix.sasaki@dfki.de

1 Einleitung: Digitale Kuratierung

Das Kuratieren digitaler Informationen, Daten, Meldungen und Medieninhalte hat sich in den vergangenen Jahren als eine grundlegende Tätigkeit mit neuen Anforderungen herauskristallisiert, die von handelsüblichen Content-Management-Systemen schon längst nicht mehr abgedeckt werden. Kuratieren ist ein komplexer wissens- und zeitintensiver Prozess, in dem Redakteure oder interdisziplinäre Teams aus heterogenen Quellen ein neues, abgestimmtes Gesamtwerk entwickeln, das auf einen spezifischen Fokus ausgerichtet ist. Die hierzu erforderlichen Arbeiten umfassen das Auswählen, Zusammenfassen, zeitliche Einordnen, Internationalisieren, Anreichern, Visualisieren und Erklären der verschiedenen Inhalte, wobei insbesondere zu berücksichtigen ist, dass Geschwindigkeit, Volumen und Anzahl der Quellen (Online-Zeitungen, Nachrichtenportale, Twitter, Facebook, Instagram etc.) sowie der zu verarbeitenden Information stetig anwachsen. Ein Beispiel ist die Entwicklung eines interaktiven Exponats für ein Besucherzentrum, das bei Ausgrabungen gefundene Objekte mit Fotos, Beschreibungen und Zeitangaben auf einer Karte visualisiert und die Auswahl geeigneter Objekte, Erstellung entsprechender Inhalte, Gestaltung der Karte und Festlegung thematischer Perspektiven erfordert.

2 Projektüberblick

Dieser Beitrag gibt einen kurzen Überblick über das Verbundprojekt „Digitale Kuratierungstechnologien“, an dem die vier in Berlin ansässigen Unternehmen art+com AG, Condat AG, 3pc GmbH und kreuzwerker GmbH sowie das DFKI als Forschungspartner teilnehmen und das voraussichtlich ab dem 1. September 2015 vom Bundesministerium für Bildung und Forschung (BMBF) gefördert wird.[1] Das Ziel des zweijährigen Vorhabens ist es, die komplexen, von Redakteuren und Wissensarbeitern durchgeführten digitalen Kuratierungsprozesse durch Sprach- und Wissenstechnologien zu unterstützen.

Das DFKI wird Komponenten aus diesem Bereich einbringen und weiterentwickeln und gemeinsam mit den vier KMU-Partnern zu einer Plattform für digitale Kuratierungstechnologien ausbauen, die Funktionen zur Recherche, Anreicherung, Analyse, Kombination (z.B. thematisch, chronologisch, räumlich), Zusammenfassung und Internationalisierung von Inhalten umfasst. Branchen- und Plattformtechnologien werden die Realisierung branchenspezifischer Workflows und skalierbarer Anwendungen in den jeweiligen Branchen vereinfachen. Die Plattform ermöglicht den Industriepartnern, innovative und effizienz- sowie qualitätssteigernde Lösungen für vier unterschiedliche Branchen (Museen und Showrooms; TV-/Radio und Web-TV; Verlage und Medienhäuser; Archive und Bibliotheken) effizienter zu entwickeln, zu betreiben, zu integrieren und zu verwerten.

Die vom Forschungspartner DFKI eingebrachten Technologien umfassen Methoden, Komponenten und Ansätze aus dem Gebiet der Sprach- und Wissenstechnologien, die im Rahmen zahlreicher Projekte wie z.B. ATLAS, COLLATE, LT Web, META-NET, QTLaunchPad, EuroMatrix, EuroMatrixPlus und Trendminer (BMBF, BMWi,

[1] Siehe http://artcom.de, http://condat.de, http://3pc.de, http://kreuzwerker.de sowie http://dfki.de/lt.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 138–139, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

EU/EC etc.) entwickelt wurden. Diese Methoden können den folgenden drei Bereichen zugeordnet werden:

1. *Semantische Analyse*: Tiefe Analyse mit hoher Präzision und der Möglichkeit zur Adaption an verschiedene Domänen am Beispiel von Informationsextraktion (Zeiten, Orte, Themen, generische benannte Entitäten), automatische Textzusammenfassung, Sentiment-Analyse sowie Klassifikation und Clustering von Informationen.
2. *Semantische Generierung:* Unterstützung des Storytellings durch Text-, Hypertext- und Reportgenerierung für ausgewählte Typen von Dokumenten auf Basis von Verfahren zur Informationsextraktion und generischen Textschemata, die z.B. als thematische oder textsortenspezifische Strukturgrammatiken repräsentiert werden.
3. *Mehrsprachige Technologien*: Robuste und adaptierbare Komponenten für maschinelle Übersetzung sowie Integration verschiedener Wissensquellen unter Berücksichtigung der Kuratierungs-Workflows bei den Industriepartnern für eingehende (Inbound-Translation) und zu publizierende Dokumente (Outbound-Translation) sowie Integration mono- und multilingualer Linked-Open-Data-Quellen (LOD).

3 Technologieplattform

An dieser Stelle soll die Bedeutung der Technologieplattform im Rahmen der Wertschöpfungskette hervorgehoben werden. Je nach Anwendungsfall und Branche fällt die Wertschöpfungskette zur Kuratierung von Inhalten unterschiedlich aus. Es sind drei Arten von Akteuren zu unterscheiden:

- die kuratierende Institution, z.B. Museum, Fernsehsender, Verlag oder Archiv;
- Dienstleister/Agenturen, die für die kuratierende Institution Inhalte und Technologien bereit stellen bzw. Komplettlösungen entwickeln (z.B. die vier KMU-Partner);
- an der Kuratierung beteiligte Redakteure und Wissensarbeiter, z.B. interne Mitarbeiter oder Dienstleister, aber auch externe Wissenschaftler, Experten oder Freiberufler.

Die Plattform für digitale Kuratierungstechnologien soll diesen unterschiedlichen Akteuren eine umfassende Menge von Funktionalitäten bieten, die den Kuratierungsprozess unterstützen. Durch den Einsatz von Sprach- und Wissenstechnologien können einzelne, bisher noch rein manuell bzw. intellektuell durchgeführte Kuratierungstätigkeiten (teil-)automatisiert werden. Die Akteure können durch die Nutzung der Plattform größere Mengen an Inhalten schneller sichten und weiterverarbeiten. Mit der Technologieplattform wird eine deutliche Effizienzsteigerung und Kostensenkung des Kuratierungsprozesses angestrebt – bei gleichbleibender oder sogar verbesserter Qualität. Dabei können, wie in Abb. 1 dargestellt, vier Schichten unterschieden werden, nämlich Kernkomponenten der Sprach- und Wissenstechnologie, allgemeine Kuratierungskomponenten, allgemeine Plattformkomponenten sowie Branchentechnologien.

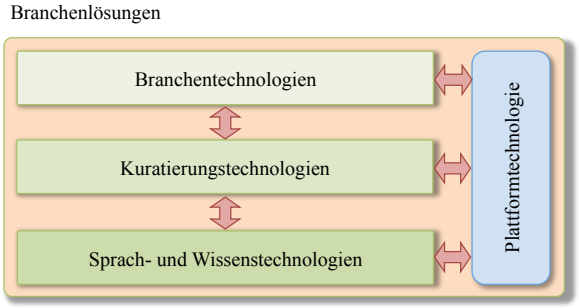


Abbildung 1: Plattform für Kuratierungstechnologien

Das DFKI wird die bereits vorhandenen Komponenten so weiter entwickeln, dass sie den Anforderungen der anderen Schichten genügen und von den Industriepartnern in ihre jeweiligen Branchenlösungen integriert werden können. Die Industriepartner wiederum konzipieren und entwickeln generische Technologiekomponenten, die für ihre Branchenanwendungen benötigt werden, aber auch in anderen Lösungen eingesetzt werden können. Wir fokussieren insbesondere die folgenden Zielmerkmale der Plattform:

- Vollintegrierte robuste, performante und skalierbare Komponenten mit offenen APIs für eine effiziente Einbettung in branchenspezifische Kuratierungs-Workflows;
- Einfache Nutzbarkeit der Cloud-Plattform durch browserbasierte SaaS-Webarchitektur;
- Anwendungsorientierte Branchenlösungen mit hoher Usability (User Interfaces, Interaktionsdesign, Informationsvisualisierung).

4 Schlussfolgerungen

Unsere grundlegende Arbeitshypothese ist, dass der gezielte Einsatz sprachtechnologischer Verfahren digitale Kuratierungsprozesse deutlich effizienter und produktiver gestalten kann und sind überzeugt, diese Hypothese gemeinsam mit den KMU-Partnern im Rahmen des hier knapp skizzierten Vorhabens verifizieren zu können.

Entering Appointments: Flexibility and the Need for Structure?

Karola Pitsch
University of Duisburg-Essen
Communication Studies
karola.pitsch@uni-due.de

Ramin Yaghoubzadeh
CITEC, Bielefeld University
Social Cognitive Systems
ryaghoubzadeh@uni-bielefeld.de

Stefan Kopp
CITEC, Bielefeld University
Social Cognitive Systems
skopp@techfak.uni-bielefeld.de

Abstract

Initial evaluation of human-agent-interaction reveals how the system's dialog strategies shape the user's input.

1 Introduction

Human-machine interfaces based on natural communication become increasingly important e.g. for assisting elderly or other people with specials needs in maintaining their daily routines. We have begun to develop a calendar and remainder application, which allows users to enter their appointments into a digital calendar by talking to an embodied conversational agent (ECA), which is presented on a large TV-screen alongside a weekly calendar. Users can interact freely with the system by using means of verbal (and in future: multimodal) communication. The system is set up to work autonomously and to extract information about date (D), time (T) and activity (A) of an appointment from the user's spontaneous speech. It uses dedicated modules for speech recognition and a multi-layered approach of securing understanding as shown in Yaghoubzadeh, Pitsch, and Kopp (2015). While the first trials are highly promising, we will be interested here in those cases in which wrong information is entered into the system as this raises questions about (i) understanding the linguistic and multimodal structure of the users' input and (ii) the ways in which the user utterances might be shaped through the conversational strategies deployed by the agent. We will present initial observations from a user study, which allows to observe the implications of a linear approach to extracting information from a user's input which was designed for the important benefit of allowing for flexibility in managing understanding and organizing repair activities. To which extent would this need to be completed with deeper information about structuring information?

2 Quantification: Success & Failure

Six senior citizens (age: 77 to 86 years) were asked to enter appointments in the digital calendar by talking freely to the ECA and using an A4 sheet with pictograms of potential events as inspiration. Quantitative analysis reveals …

- a high amount of correct entries with only one user (04) experiencing difficulties.
- if problems occur, they constitute a failure in the slot 'activity' or an abortion.
- the mean time of entering an appointment to be around 30'' for 2 seniors (which corresponds to young control users) and less than 60'' for 3 seniors.

User	Σ	Correct E	False Entry			Aban doned	Ø-Time
			D	T	A		
01	**9**	**5**		2*	3*		00:58
02	**8**	**7**			1		00:37
03a	**3**					3	
03b	**8**	**7**				1	00:36
04	**6**	**1**			3	2	02:31
05	**9**	**8**			1		00:57
06	**8**	**7**			1		00:51

Figure 1: Success/Failure of entering appointments. (*) = two failures in one entry. User 3a/b is the same person the trial of whom was interrupted.

Despite the encouraging results the question arises: How are the faulty 'activity'-entries produced in the interaction between user and system?

3 Exploration: Discursive strategies

Two cases from user 01 (`[A]`, `[B]`) are examined.

3.1 Extracting information dynamically

After the system (S) initiates the sequence, users (U) formulate an appointment using natural speech (often with specific gaze coordination):

```
01 S: do you have another appointment,    [A]
02 U: yes- I have another appointment
03 U: =on WEDnesday, (.) i want to go to
04 U: the restaurant, at 15 o'clock
```


Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 140–141, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

The system shows its understanding of the user's input step by step both through talk and by highlighting the corresponding slot in the calendar (Fig. 2a, b). The user takes up this stepwise procedure either ratifying each step individually ([A] 07, 09, 12) or adding missing pieces of information dynamically ([B] 06, 08).

```
05 (2.0)                                          [A]
06 S: [#2a] then on wednesday, (.) right,
07 U: yes;
08 S: [#2b] then at 15 o'clock, (.) right,
09 U: yes;
10 S: good; there then is restaurant, (.)
11 S: right,
12 U: yes;
```

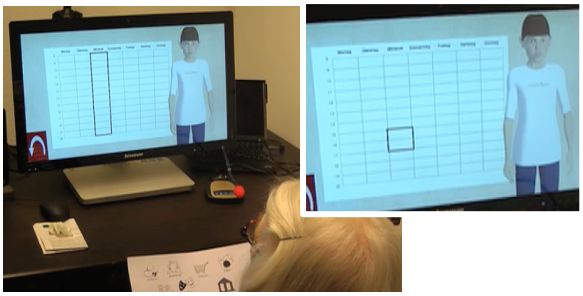


Figure 2a and 2b: Setup and highlighting slots

```
03 S: saturday, right;                            [B]
04 U: yes;
05 S: good;
06 U: at 13 o'clock
07 S: then at 13 o'clock, right,
08 U: yes; for lunch;
09 S: okay;
```

This flexible design of the dialog allows, in these cases, to extract all relevant information.

3.2 Expansions: Co-construction and the emergent nature of the user's input

Despite the availability of the correct information, the final calendar entries contain a faulty 'activity'. Analysis reveals these problems to result from users expanding their utterance incrementally adding more information about the activity. In [A], the agent ratifies the last entry (activity) (13: "good"). The user treats the ensuing 1.0-second pause and absence of information uptake on the screen as a moment to add an increment to his utterance (15:"in <city>"), which is repeated by the system (16) and ratified (17).

```
13 S: good; (1.0),                                [A]
14 (1.0)
15 U: in <city>
16 S: then there is <city>; right,
17 U: ye=hes;
```

In [B], the system has problems understanding "lunch" as the desired activity and seeks for clarification (10). The user again offers this piece of information (12) and the system picks it up (13). The system does not show immediate uptake in the calendar, which the user (in combination with the repeated questions) treats as a moment to expand her utterance in two steps (14-15). The system picks up the last piece of information "his vacations" (16) which is ratified (17).

```
10 S: okay; what did you plan there,              [B]
11 S: what did you plan there;
12 U: lunch;
13 S: then there is lunch; right,
14 U: lunch, yes:; (-) and a film screening;
15 U: (.) of his vacations;
16 S: then there is his vacations; right,
17 U: yes:;
```

This way, the agent's verbal conduct and the calendar display contribute to inviting the user to expand her utterance and to co-construct an input, which is difficult to handle for the system.

3.3 Overwriting entries & its acceptance

In both cases, the initially correctly extracted information (slot 'activity') is overwritten in the final entry by the information extracted from the users' expanded utterance ([A],19, [B],19-20).

```
18 S: okay; then I add the following              [A]
19 S: wednesday at 15 o'clock is <city>;
20 S: right,
20 U: .hh ye=hes;
```

```
18 S: okay; then I add the following              [B]
19 S: saturday at 13 o'clock there is his
20 S: vacations; right,
21 U: ye=hes;
```

While it has not been transparent to the user – when locally ratifying the 'activities' "<city>" ([A],15) and "his vacations" ([B],17) – that these would overwrite the 'activity'-information provided before ([A],10, [B],8) – it is noteworthy here that the user accepts both calendar entries.

4 Implications & Discussion

These observations point to investigating further (i) strategies for technical system to shape user conduct, and (ii) procedures for detecting completeness of information and managing the end of sequences. This will require also a discussion about linking such type of structural information with generic flexible computational approaches.

References

Ramin Yaghoubzadeh, Karola Pitsch and Stefan Kopp. 2015. Adaptive grounding and dialogue management for autonomous conversational assistants for elderly users. *Proceedings IVA 2015*.

Konventionalisierung und Interaktion - das *Pledari Grond Online*

Claes Neuefeind
Sprachliche Informationsverarbeitung,
Institut für Linguistik, Universität zu Köln
Albertus-Magnus-Platz, 50923 Köln
c.neuefeind@uni-koeln.de

Abstract

Online-Wörterbücher dienen vor allem dem täglichen Gebrauch. Für Kleinsprachen übernehmen sie darüber hinaus jedoch auch weitere Funktionen: Als gemeinsame Plattform, als normierende Quelle, und nicht zuletzt auch als strukturierte Ressource, die als Grundlage für weitergehende sprachtechnologische Anwendungen herangezogen werden kann, welche selbst wieder einen Beitrag zur Stärkung von Kleinsprachen leisten. Mit dem *Pledari Grond*[1] stellen wir hier ein in diesem Sinne ausgedeutetes Online-Wörterbuch für das Bündnerromanische vor, das die Einbindung der Sprachgemeinschaft und die lexikographische Arbeit einer betreuenden Redaktion in einer computerlinguistisch motivierten technologischen Umsetzung kombiniert.

1 Einleitung

In der EU gibt es offiziell 24 Amtssprachen. Zählt man jedoch auch die Minderheitensprachen hinzu, so werden in Europa mehr als 100 Sprachen gesprochen.[2] Auch wenn die meisten Mitgliedstaaten die "Europäische Charta für Minderheitensprachen"[3] ratifiziert und damit einen grundsätzlichen Schutzauftrag für Kleinsprachen übernommen haben, müssen kleinere Sprachgemeinschaften zusätzliche Wege für den Erhalt und die Pflege ihrer Sprache entwickeln. Dies trifft auch für das Bündnerromanische in der Schweiz zu, das offiziell als bedrohte Sprache gilt – wenngleich es im Gegensatz zu den meisten anderen Minderheitensprachen als vierte Landessprache eine gezielte institutionelle Unterstützung seitens der Eidgenossenschaft erfährt.

In diesem Zusammenhang spielt die lexikographische Aufbereitung des Wortschatzes eine zentrale Rolle: Aus Sicht von Kleinsprachen ist der Wortschatz im nämlichen Sinne ein Schatz, den es zu hüten, auszustellen und zu pflegen gilt – ist er doch in besonderem Maße Ausdruck kultureller Identität, die in den spezifischen Unterschieden gegenüber anderen, insbesondere dominierenden Sprachen zum Ausdruck kommt.

In Bezug auf Kleinsprachen ist die Aufgabe des Lexikographen damit mindestens eine doppelte: Zum einen muss er Archivar und akribischer Dokumentar sein, präzise und umfangreich zugleich; zum anderen muss er im Hinblick auf die Nutzung eine zu hohe Komplexität vermeiden, um ein möglichst nutzerfreundliches Wörterbuch für den täglichen Gebrauch bereitstellen zu können. Ziel des Pledari Grond ist es, diese beiden Perspektiven zusammen zu bringen und technologisch zu unterstützen.

2 Das Pledari Grond Online

Das Pledari Grond (PG) wurde in enger Zusammenarbeit mit der linguistischen Abteilung der Lia Rumantscha[4] entwickelt und umfasst derzeit etwa 224.000 Einträge. Vordergründig ist das PG ein einfaches Online-Wörterbuch für das Rumantsch Grischun (RG), vergleichbar zu LEO[5], dict.cc[6] oder auch Linguee[7], das wie seine ‚großen Geschwister' bei der Erweiterung und Aktualisierung der Datenbasis auf die Mitarbeit der Nutzer setzt. Der wesentliche Unterschied liegt in der speziellen Sprachsituation Graubündens: Zwar fungiert das RG als offizielle Amtssprache des Kantons (neben Deutsch und Italienisch), es ist jedoch eine reine Schriftsprache, die erst 1982 im Zuge sprachplanerischer

[1] http://pledarigrond.ch
[2] http://www.eurominority.eu
[3] http://conventions.coe.int/Treaty/ger/Treaties/Html/148.htm
[4] http://liarumantscha.ch
[5] http://www.leo.org/
[6] http://www.dict.cc/
[7] http://www.linguee.de/

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 142–143, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

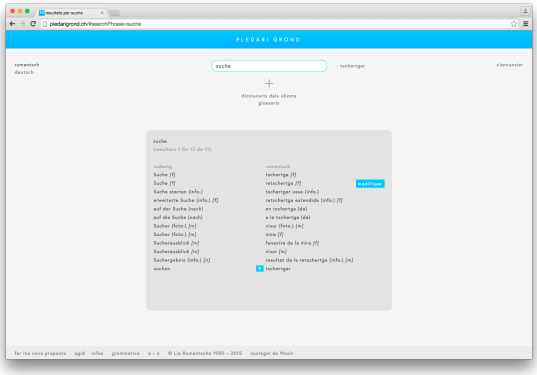


Figure 1: Das Suchinterface des Pledari Grond: Nutzer können Änderungen und Ergänzungen direkt im Suchergebnis vorschlagen.

Maßnahmen konzipiert wurde (Schmid, 1982).[8] Aus dieser spezifischen Konstellation ergibt sich, dass das PG als *die* zentrale Ressource des RG vor allem auch den Aufbau einer erweiterten lexikographischen Datenbasis leisten muss, die als Grundlage für Druckfassungen wie auch für weitergehende sprachtechnologische Anwendungen wie Orthographie-Korrektur, Maschinelle Übersetzung, etc. dienen kann.

Um gleichermaßen der Funktion als Online-Wörterbuch, dem Anspruch einer Konventionalisierung des im Aufbau befindlichen Wortschatzes, sowie der Funktion als standardisierte Datenquelle gerecht werden zu können, bietet das PG unterschiedliche Perspektiven auf die Daten: Während die Einträge in der Nutzerperspektive mit vereinfachten lexikographischen Angaben dargestellt werden und direkt im Suchergebnis ergänzt und korrigiert werden können (siehe Abb. 1), ist das PG für die redaktionelle Bearbeitung zusätzlich mit einem umfangreichen Editor-Backend ausgestattet, über das Vorschläge und vorhandene Lemmata nach erweiterten lexikographischen Kriterien angereichert werden können (siehe Abb. 2).[9]

Um darüber hinaus möglichst flexibel auf zukünftige Entwicklungen reagieren zu können, gestattet die technologische Umsetzung mit einer *NoSQL*-Datenbank im Zusammenspiel mit *Spring* und *GWT* die bedarfsabhängige Erweiterung um zusätzliche lexikographische Angaben, sowohl in den Daten als auch in Front- und Backend.

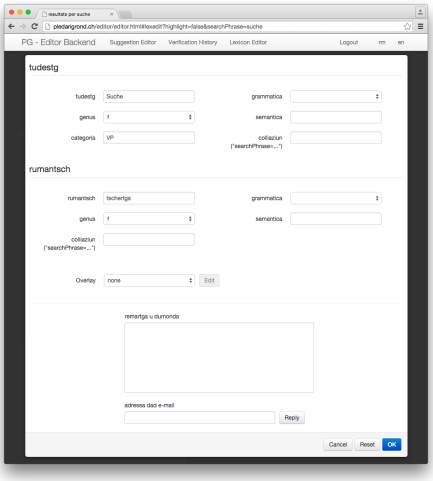


Figure 2: Im Editor-Backend werden sämtliche Einträge von den Redaktoren der Lia Rumantscha bearbeitet und lexikographisch angereichert.

3 Erweiterungen und Perspektiven

In der hier beschriebenen Umsetzung ist das PG bereits seit April 2014 online und erfüllt dabei die oben formulierte Mehrfachfunktion: Intern wird eine standardisierte strukturierte Ressource für das RG aufgebaut und kontinuierlich durch die Lexikographen der Lia Rumantscha erweitert; gleichzeitig dient es nach außen als Online-Wörterbuch für den täglichen Gebrauch, das dem Nutzer eine auf das Wesentliche reduzierte Wortliste präsentiert und eine barrierefreie und unmittelbare Nutzerbeteiligung ermöglicht. Durch die Einbindung von entsprechenden Analysetools für Nutzerverhalten und -herkunft gibt diese Konstellation zudem Einblick in wertvolle Daten; längerfristig bietet sich so eine einzigartige Chance, Prozesse der Konventionalisierung und Interaktion einer kleinen Sprachgemeinschaft – gewissermaßen ‚in vivo' – zu untersuchen.

[8] Tatsächlich umfasst das Bündnerromanische fünf Idiome, die aktiv gesprochen werden: Sursilvan, Sutsilvan, Surmiran, sowie Puter und Vallader, siehe dazu u.a. (Liver, 2010)

[9] Neben grammatischen Angaben umfasst dies Besonderheiten der Wortbildung, Verwendungsbeispiele und Angaben zur Semantik. Die Angaben dienen zudem als Filterkriterien für den flexiblen Export beliebiger Teilmengen der Daten.

References

Ricarda Liver. 2010. *Rätoromanisch. Eine Einführung in das Bündnerromanische*. Gunter Narr, Tübingen.

Heinrich Schmid. 1982. *Richtlinien für die Gestaltung einer gesamtbündnerromanischen Schriftsprache Rumantsch Grischun.* Societad Retorumantscha, Chur.

Towards parsing language learner utterances in context

Christine Köhn and Wolfgang Menzel
Department of Informatics
Universität Hamburg
{ckoehn, menzel}@informatik.uni-hamburg.de

Abstract

Intelligent computer-assisted language learning (ICALL) systems give feedback on grammatical errors based on the syntactic analysis of an utterance. However, parsing a language learner sentence in isolation can result in multiple interpretations, which even humans may not be able to resolve. For deriving a helpful error diagnosis, an interpretation that is consistent with the intended meaning is needed. To achieve this, we propose an approach that models information about the context of a sentence by means of conceptual relationships and integrates them into the decision procedure of a syntactic-semantic parser. This way, we obtain a syntactic structure which complies best with a given context model. We identify and categorize cases where the context influences both the structure and the error diagnosis derived from it.

1 Introduction

For diagnosing grammatical errors in a language learner sentence, its syntactic structure is needed. Based on the syntactic structure, an error diagnosis and a correction can be inferred. A syntactic analysis based on the sentence alone is not sufficient for this purpose, because the context in which a language learner utters a sentence sometimes plays a crucial role for grammatical error diagnosis. Depending on the context, a sentence can be judged as correct or erroneous. Furthermore, among the different error diagnoses for a faulty utterance, the context can help to choose the ones which are more likely than the others.

Different types of context information can be exploited to interpret a language learner utterance such as the learner's native language and the exercise. In this paper, we focus on a subset of the context, namely the situation the learner is writing about. In the following, examples will be given to illustrate different cases where the context of an utterance influences the error diagnoses.

The German[1] sentence in Example 1 is well-formed. However, *See* is ambiguous with respect to grammatical gender and if *See* is meant to refer to a lake instead of the sea, the grammatical gender of *See* is masculine. Therefore, the corresponding article needs to be masculine as well and the sentence is erroneous. Example 2 shows the corrected sentence.

E1	Das	Haus	liegt	an	**der**$_{\text{f,dat}}$	See$_{\text{f,dat}}$
	The	house	is located	by	the	sea
E2	Das	Haus	liegt	an	**dem**$_{\text{m,dat}}$	See$_{\text{m,dat}}$
	The	house	is located	by	the	lake

Assuming that *See* denotes a lake in Example 1, two probable error diagnoses are available: The student inflected the article correctly (dative case) but assumed the wrong grammatical gender for the word *See* (*der*: feminine, *dem*: masculine), or the student chose the correct gender (masculine) but did not use the right case (*der*: nominative, *dem*: dative). Which of these error diagnoses expresses the student's misconception best cannot be determined without further background knowledge but could be achieved through student modeling. This, however, is not considered in this paper.

A clearly erroneous sentence is shown in Example 3 where the article *die* needs to be replaced. Depending on whether *See* denotes the sea or a lake, it can be corrected into Example 1 or Example 2, respectively. If *See* means the sea, the student probably selected the appropriate gender (feminine) but the wrong case (*die*: nominative or accusative, *der*: dative). If *See* means a lake, not only the case is wrong (same as before) but also the gender (*die*: feminine, *dem*: masculine).

E3	*Das	Haus	liegt	an	**die**$_{\text{f,nom/acc}}$	See$_{\text{f/m,dat}}$
	The	house	is located	by	the	sea/lake

[1] All of the examples given in this paper are in German.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 144–153, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

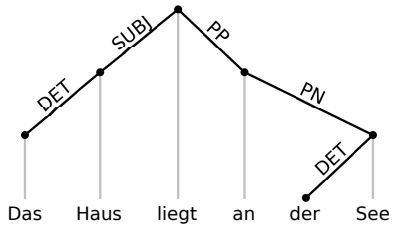
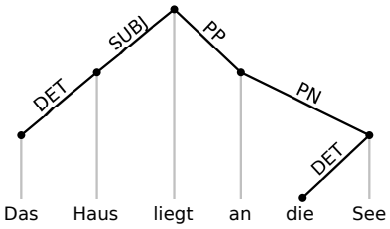
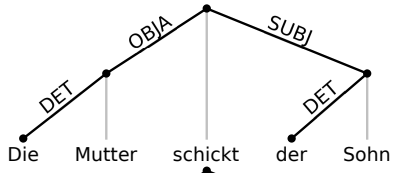
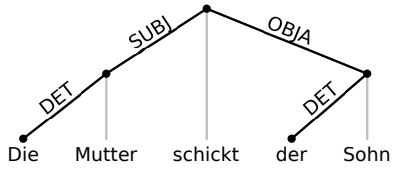
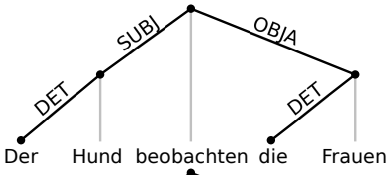
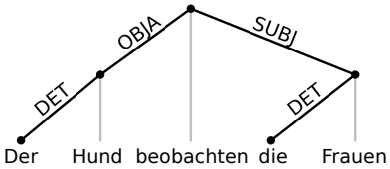
Synt. structure	Distinctions induced by the context: Erroneous vs. correct	Distinctions induced by the context: Different diagnoses and corrections
invariant to context	 Example 1	 Example 3
dependent on context	  Example 4	  Example 6

Table 1: Overview of the example sentences **E1**, **E3**, **E4** and **E6** illustrating the influence of the utterance context on syntactic parsing and error diagnosis (see text). Labels: `DET` (determiner of a noun), `SUBJ` (subject of a verb), `OBJA` (accusative object) and `PP`/`PN` (prepositional phrase/complement)

These examples illustrate that the context can help to distinguish between erroneous and correct sentences as well as narrow down the set of error diagnoses. The syntactic structures for the above examples are shown in Table 1.[2] For examples 1 to 3, the syntactic structures are the same, regardless of the sea/lake distinction. In other cases, the syntactic structure of a sentence might depend on the context: Example 4 shows a sentence with a structural ambiguity. It is correct but only as long as we assume that the mother is the object of the sentence and the son is the subject.

E4

(*)Die$_{\text{nom/}\mathbf{acc}}$	Mutter$_{\text{nom/}\mathbf{acc}}$	schickt	**der**$_{\text{nom}}$	Sohn$_{\text{nom}}$
The	mother	is sending	the	son

(The son is sending the mother)

If the sentence in Example 4 is uttered in a context where the mother is sending the son, then the mother is the subject and the son the object (see Table 1 for the syntactic structures). Under this reading, the sentence becomes erroneous: Being the object, *der Sohn* has to be accusative case but it is nominative case. Therefore, the article needs to be changed to *den* (Example 5).

E5

Die$_{\text{nom}}$	Mutter$_{\text{nom}}$	schickt	**den**$_{\text{acc}}$	Sohn$_{\text{acc}}$
The	mother	is sending	the	son

Cases where the sentence is obviously erroneous are similarly difficult. Then, different syntactic interpretations for that sentence might be possible, and, consequently, the error diagnoses inferred from them may differ. Example 6 shows such a case: A syntactic structure which assigns dog as the subject and (several) women as the object implies a verb form error because the subject *Hund* and the verb *beobachten* do not agree in number (*Hund*: singular, *beobachten*: plural). To correct the error, the verb has to be changed to singular (Example 7). Given a context in which the women are watching a dog, another syntactic structure which assigns dog as the object and women as the subject (see Table 1) is more adequate. Based on this interpretation, a different error diagnosis can be obtained: The object *der Hund* has the wrong case (nominative instead of accusative). Thus, the article *der* needs to be corrected to *den* (Example 8).

[2]The syntactic structures are displayed as dependency trees. We apply the annotation scheme by Foth (2006). A short description of the labels in English can be found in Foth et al. (2014).

E6 *Der$_{sg,nom}$ Hund$_{sg,nom}$ beobachten$_{pl}$ die$_{pl,nom/acc}$
The dog are watching the
Frauen$_{pl,nom/acc}$
women

E7 Der Hund **beobachtet$_{sg}$** die Frauen
The dog is watching the women

E8 **Den$_{acc}$** Hund$_{acc}$ beobachten die Frauen
The dog are watching the women
(The women are watching the dog)

The examples show that the context of an utterance influences its grammaticality judgment as well as the error diagnoses. This paper focuses on situations where the context gives rise to different syntactic structures and to different error diagnoses, which can be derived from them. We will adopt a method for modeling the context information and integrating it into the decision procedure of a syntactic parser to obtain the syntactic structure that is most plausible with respect to the given context.

The remainder of this paper is structured as follows: The parsing formalism (Section 3) and the context models (Section 4) we work with will be explained. Section 5 will illustrate, by means of an example, an approach for integrating context information into parsing. Section 6 will describe one core mechanism of the context integration process, the selection of referents. In Section 7, we will systematically identify and categorize cases where context integration could help to distinguish different syntactic structures. Limitations of this approach will be discussed in Section 8. Section 9 will conclude the paper.

2 Related Work

When parsing language learner utterances, the parser has to be error-tolerant, and in addition, its output should contain information that can be used for generating an error diagnosis. Several approaches exist for parsing language learner sentences in order to diagnose grammatical errors, e. g., Heift (2003), Reuer (2003), Bender et al. (2004), Fortmann and Forst (2004) and Boyd (2012).

To our knowledge, there are only two approaches which integrate context information into the parsing procedure itself when analyzing language learner sentences: An early experiment was conducted by Menzel and Schröder (1999) where domain knowledge was integrated into syntactic parsing of artificially distorted sentences originating from a single sentence. More recently, Antonsen et al. (2009) have used a parser which implements the Constraint Grammar framework (Karlsson et al., 1995) for analyzing learner sentences in an ICALL system where the user answers questions asked by the system. The parser uses two rule sets: The first set disambiguates the input partially in order to find appropriate readings for erroneous input. The second set contains rules that flag errors in the input. Context information such as verb tense and case of the interrogative determiner from the question restricts the interpretation of the user's answer. In contrast to Antonsen et al. (2009), we do not expect the input to be questions and answers but free-form text from writing exercises such as picture description tasks where a question is not necessarily available for the interpretation of the input.

Some ICALL applications extract a semantic interpretation from the syntactic structure of a sentence for further processing but the semantic information is not used for parsing itself. Hahn and Meurers (2012), for example, evaluate the meaning of short answers. For this purpose, MaltParser (Nivre et al., 2007), a dependency parser, obtains a syntactic structure from which a semantic representation is derived.

3 Parsing with constraint relaxation

A dependency parser constructs a structural description, the dependency tree, of an input sentence by assigning each word (the dependent) to a regent (either another word or the special root node). The parser labels these dependencies to characterize the relationship between dependent and regent. For parsing, the Weighted Constraint Dependency Grammar (WCDG) parser is used (Foth et al., 2004). In the WCDG formalism, well-formedness conditions for dependency trees are expressed as constraints. Each constraint stipulates a condition which a set of edges should satisfy. In addition, it is graded with a weight from $[0,1]$, which indicates how severe it is if a dependency tree violates the particular constraint: The closer the weight is to 0, the more severe the violation. For example, having two subjects as dependents of a verb is more severe than assigning a subject to a verb which does not agree with it. The WCDG parser finds the structural description that best adheres to all constraints defined in the grammar: The best structure s is defined as

$$s = \arg\max_{s'} \prod_{c \in \text{Constraints}} \text{weight}(c)^{\text{n}(c,s')} \quad (1)$$

where weight(c) is the weight of constraint c and $\mathrm{n}(c, s')$ is the number of times c is violated in the structure s', i. e. the number of times the parser had to relax c to obtain s'. The parser performs a heuristically-guided transformation-based search, which starts with an initial structure and tries to resolve constraint violations iteratively.

The constraint relaxation mechanism makes the parser robust to ill-formed input (Foth et al., 2005). As a byproduct, the unresolved constraint violations can be used to infer error diagnoses for the input sentence.

WCDG achieves a labeled attachment accuracy on the learner corpus CREG-109 (Ott and Ziai, 2010) of 79.28% (Krivanek and Meurers, 2011). On newspaper data, an accuracy of 81.42%[3] (Krivanek and Meurers, 2011) was measured on the TüBa-D/Z corpus (Telljohann et al., 2004) and 91.0% (Foth and Menzel, 2006) on the NEGRA corpus (Brants et al., 1999). Currently, the German grammar contains more than 1000 handwritten constraints.

In the WCDG formalism, a dependency structure may span multiple levels of analysis. In addition to the syntactic level, we will use a semantic level, which contains a semantic analysis in the form of thematic roles. This level serves as an interface for the context integration.

4 Modeling context

A context model specifies high-level information about the situation, in which a sentence was uttered. In this paper, we follow McCrae (2010) and model only part of the context, i. e., actions and their participants and relationships between them. Later on, we will show that this part is sufficient to influence the syntactic analysis of possibly erroneous utterances.

A context model consists of two parts: situation-invariant knowledge (the T-Box) and situation-dependent knowledge (the A-Box), which we specify using the description logic OWL (Web Ontology Language, Bechhofer et al. (2004)).

4.1 T-Box

The T-Box (terminological box) is an ontology, which defines classes and relationships between them, e. g., MOTHER $\xrightarrow{\text{is\_a}}$ WOMAN. The IS-A relations form a hierarchical taxonomy. Each class has one or more lexicalisations, which specify the words and phrases that can be used to denote individuals of that class. Every individual described in the situation-dependent knowledge instantiates one of the concepts from the T-Box.

The T-Box can be manually created for the domain as was done in McCrae (2010) but a readily-available ontology such as GermaNet (Hamp and Feldweg, 1997) could also be exploited for that purpose.

4.2 A-Box

The A-Box (assertional box) contains individuals, which are instances of concepts in the T-Box. The A-Box defines relationships between individuals or assigns properties to them. An individual represents an object, an action or a participant who engages in an action. McCrae (2010) defines several relations which can hold between an action and its participants in terms of the thematic role the participant fulfills in that action. In Figure 1b, for example, WOMAN_1 is modeled as the AGENT of the sending action.

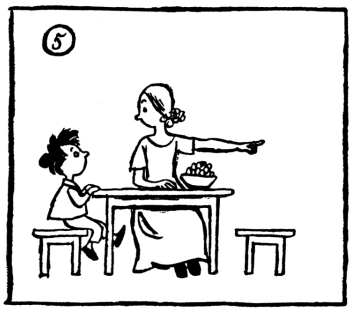
The A-Box can either be manually created as in McCrae (2010) or semi-automatically using the parser again: Simple sentences particularly authored for that purpose can be analyzed in order to extract individuals and the relationships between them automatically.

McCrae (2010) and Baumgärtner et al. (2012) use the A-Box to describe a visual scene and therefore limit the context model to include only visually perceivable information or indirectly derived information inferred from the visual input. Other scenarios are also conceivable, where the context information is derived from a textual description accompanying an exercise or from the preceding discourse.

5 Obtaining a context-induced syntactic structure

Guided by Example 4, this section will explain how the parser obtains a syntactic structure that is compatible with a given utterance context. Parsing the sentence using solely linguistic features results in a syntactic structure where the mother is the object and the son is the subject (first analysis for Example 4 in Table 1) because a parser usually prefers a structure which is as well-formed as possible. Thus, judged by this structure, the sentence has to be considered correct German. However, if the sentence

[3] The low accuracy might be caused by mismatches in the annotation scheme.



(a)

WOMAN_1	is_instance_of	WOMAN
BOY_1	is_instance_of	BOY
SEND.SB_1	is_instance_of	SEND.SB
WOMAN_1	is_AGENT_for	SEND.SB_1
BOY_1	is_THEME_for	SEND.SB_1

(b)

Figure 1: An image from a picture story (Ohser, 1993) and the A-Box of its context model

was uttered while describing a situation where actually the mother is sending the son, it is erroneous. Interchanging the subject and object in the syntactic structure so that the structure complies with the described situation would reveal that the wrong case was chosen for *der Sohn*. Therefore, we want the parser to output a context-induced syntactic structure where the mother (subject) is sending the son (object).

To deal with this problem, the parser has to be enabled to choose one syntactic structure in one context and another in a different context for the same sentence, even though the sentence might contain errors in one of these interpretations. For this purpose, we adopt a model developed by McCrae (2009) and Baumgärtner et al. (2012): By mapping propositions about the context to syntactic relationships in the sentence, the parser is guided towards a syntactic structure which conforms to a given visual context. So far, this model has only been applied to syntactically ambiguous well-formed German sentences with the objective of modulating the syntactic structure with context information, e. g., the attachment of prepositional phrases. Our goal is to analyze learner utterances, whereas the goal of the aforementioned work was to disambiguate correct German sentences. Nonetheless,

"Die" *The*	is_conceptualised_by	{}
"Mutter" *mother*	is_conceptualised_by	{MOTHER}
"schickt" *is sending*	is_conceptualised_by	{SEND.SB}
"der" *the*	is_conceptualised_by	{}
"Sohn" *son*	is_conceptualised_by	{SON}

(a) Words mapped to concepts

"Die"	matches	{}
"Mutter"	matches	{WOMAN_1}
"schickt"	matches	{SEND.SB_1}
"der"	matches	{}
"Sohn"	matches	{BOY_1}

(b) Candidate referents for words

Figure 2: Finding referents for words in the context model

their model is also capable of resolving some of the ambiguities caused by errors. When we apply that model, we are able to obtain a context-induced syntactic interpretation of the example sentence.

The sentence in Example 4 is the beginning of a genuine language learner sentence written by a learner when asked to describe the content of the picture in Figure 1 as part of a picture story. We assume that the parser receives a context model for the picture as input, i. e., a high-level description of the situation depicted (Figure 1): A woman (WOMAN_1) and a boy (BOY_1) are engaging in a sending action (SEND.SB_1) where the woman is the agent of that action (the one that is sending somebody) and the boy is the theme (the one that is being sent). Each of the individuals (WOMAN_1, BOY_1, SEND.SB_1) are instances of classes of the ontology in the T-Box.

First, a connection between the sentence and the context model is established. For each word, a matching individual (the word's referent) is searched in the A-Box: The word's lemma is compared to the lexicalisations of each class in the ontology, obtaining a set of activated concepts for each word (Figure 2a). The set of candidate referents for each word is built by adding each individual whose class is close to a concept activated by the word (Figure 2b). As a result, not only direct matches (SEND.SB_1 for "schickt") but also conceptually related individuals (WOMAN_1 for *Mutter* and BOY_1 for *Sohn*) are selected as candidate referents. Among all possible assignments of candidates to words, a scoring mechanism deter-

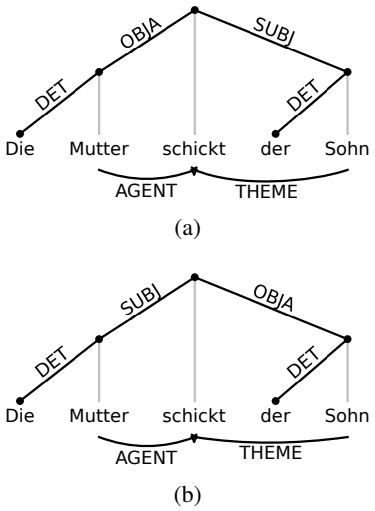


Figure 3: Analyses of Example 4

mines the best assignment. For our example, this would result in the mapping given in Figure 2b.

The information from the context model is integrated into parsing by means of constraints that access the information made available by the linking of words to their referents. One constraint could, for example, require that an agent (or theme) relation between two words in the analysis must also exist between their referents in the A-Box. By applying such constraints to the semantic level of the analysis, the attachments on this level can be pushed to reflect the relationships the word's referents engage in the A-Box. In our example, the *Mutter* would be attached to *schickt* by an `AGENT` edge, and *Sohn* would be attached as the `THEME` (Figure 3a).

If the parser's grammar contains constraints that each describe a condition either for the syntactic level or the semantic level, the parser would output the structure in Figure 3a: It complies best with the constraints on the syntactic level and on the semantic level in isolation, but the syntactic and semantic analysis contradict each other. To obtain the desired structure where *Mutter* is the subject and *Sohn* is the object, constraints which mediate between the syntactic and the semantic level need to be employed to influence the syntactic structure. Interchanging subject and object in Figure 3a can be achieved by adding the following constraints: The `THEME` should be the accusative object `OBJA` and the `AGENT` the subject `SUBJ` (in an active-voice sentence). If the weights of these constraints penalize their violation more heavily than the ones on the syntactic level, the parser obtains the structure in Figure 3b: The syntactic level is not well-formed anymore but complies with the context model. Based on this structure, a case error can be diagnosed: *der Sohn* (nominative) has to be changed to *den Sohn* (accusative).

6 Selecting Referents

Candidate referents for a word are found by comparing the lexicalisations of the concepts in the T-Box to the lemma of the word. Every individual in the A-Box which is an instance of a matching concept or an instance of a concept close to a matching concept is a candidate referent for the word. To select adequate referents for the words in the sentence, each possible assignment of referents to words is rated and the assignment with the highest score is selected.

Several measurements contribute to the overall score of an assignment (Baumgärtner et al., 2012):

Distance between concepts The suitability of a referent for a word depends on the distance of the concept instantiated by the referent to the concept activated by the word in the taxonomy. The closer the concepts, the higher the rating for assigning a referent to a word.

Incorporating the conceptual distance makes the matching process more robust because it allows to match concepts which are on different levels in the taxonomy (e. g., MOTHER and WOMAN) but even if there is no super- or subclass relationship between the two concepts it may be reasonable to establish a match if the concepts are close enough (e. g., CUP and MUG).

Similarity between word and lexicalisation The more similar the lemma of the word and the lexicalisation of the referent are on the character level, the better the assignment is rated. This makes the matching robust to slight spelling and typing errors.

Connections between referents A referent is rated higher if it is related to an individual in the A-Box which has been selected as a referent for another word.

This increases the likelihood that the appropriate referents are chosen in cases where the conceptual distance and the character similarity do not distinguish between two candidate referents. For example: If there were another woman, WOMAN_2, in the A-Box of Figure 1b, WOMAN_1 would be rated higher than WOMAN_2 in a sentence where

	SUBJ	OBJA	OBJD	GMOD	S
OBJA	•				–
OBJD	•	•			–
DET	•	•	•		–
GMOD	•	•	•		–
APP	•	•	•	•	–
REL	–	–	–	–	•

Table 2: Confusion matrix of syntactic functions. "•": Confusion is possible. "–": No example has been found. Please note that the table is symmetric: The entry for (DET, OBJA), e. g., can be found in cell (OBJA, DET). Labels: APP (apposition), REL (relative clause), S (sentence); for the other labels see Table 3.

Information in context model	Promoted syntactic attachment
AGENT	SUBJ (subject of a verb)
THEME	DET (determiner of a noun) OBJA (accusative object of a verb) SUBJ (subject of a verb)
RECIPIENT	OBJD (dative object of a verb)
OWNER	DET (determiner of a noun) GMOD (genitive modifier of a noun)

Table 3: Syntactic attachments which are promoted by the relations in the context model

SEND.SB_1 has already been selected as a referent for another word.

Influence from the syntactic structure Relationships expressed in the syntactic structure of a sentence may also be present in the A-Box. For example: If the syntactic structure assigns an attribute (e. g. "big") to a noun (e. g. "dog"), an individual which exhibits the same property is a more likely referent for that word than another individual (a dog who is big as opposed to any other dog).

The selection process does not require that all individuals from an A-Box have to be referents for words, and vice versa, not every word has to have a referent in the A-Box. Additionally, the context model does not restrict the word order and allows for a variety of word choices, since referents can be matched via the taxonomy in the T-Box. As a result, an A-Box is more general than a sentence with the textual description of the A-Box: One A-Box can be matched to numerous sentences.

The selection of referents is an iterative process, since the referents and the parser's analysis mutually influence each other. After the selection, the referents for each word are fed back to the parser, which reanalyzes the sentence. Whenever the parser finds an analysis with a higher score with respect to Equation (1), the selection of referents is renewed.

7 Structurally Ambiguous Cases

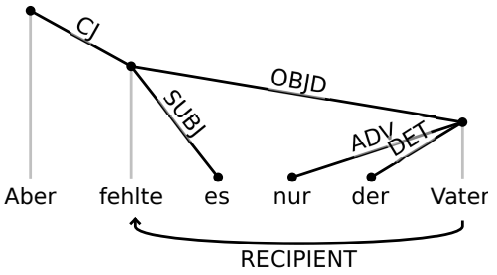
If errors are present in a sentence, the syntactic structure might be ambiguous in a way that the ambiguity can only be resolved by integrating context information into syntactic parsing (cf. Example 4 and 6 in Section 1). We systematically analyzed common language learner error types such as case and gender selection errors (Rogers, 1984) as to when they cause ambiguities with respect to syntactic functions.

We have grouped the error-induced ambiguities into confusion classes: Table 2 shows the syntactic functions (denoted by the name of the label) that could be confused with each other if no biasing context information is available. Confusions are not limited to the exchange of syntactic functions, e. g. the subject SUBJ against the accusative object OBJA, but can also affect the attachment of a word: Confusing the dative object OBJD and the genitive modifier GMOD, e. g., implies that the word's regents differ, since dative objects are attached to a verb, and genitive modifiers to a noun.

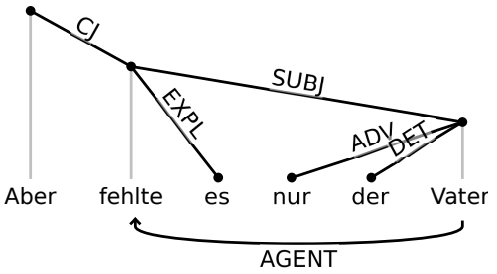
For each confusion of syntactic functions, we have identified disambiguating context information in the form of thematic roles. Four of the six thematic roles employed by McCrae (2010) are relevant for disambiguating the structural ambiguities from Table 2: AGENT, THEME, RECIPIENT and OWNER. Each role influences the attachment of words in the sentence (by means of the mediating semantic level of the analysis) in a different way, e. g., an AGENT of an action is more likely to be the subject of the sentence in active voice, whereas the RECIPIENT is more likely to be the dative object. Table 3 gives an overview of the syntactic attachments which are promoted by the relationships in the A-Box[4].

Figure 4 shows two analyses of an erroneous real-world learner sentence, which exhibits the

[4]Example sentences which exhibit the confusions of Table 2 and show the disambiguating capabilities of the thematic roles from Table 3 can be found in Köhn and Menzel (2015), the extended version of this paper.



(a) Interpreting "father" as dative object implies a word order and a case error with respect to the target hypothesis "Aber **es fehlte** nur **dem** Vater" (But only the father missed it).



(b) Interpreting "father" as the subject implies a word order error with respect to the target hypothesis "Aber **es fehlte** nur der Vater" (But only the father was missing).

Figure 4: Two different analyses of the same learner sentence (Literally: But missed it only the father).

SUBJ/OBJD confusion. Depending on the context, the father is either the subject or the object. Without any context information, both WCDG and the machine learning-based TurboParser[5] (Martins et al., 2009; Martins et al., 2013) obtain the analysis in Figure 4a (without the RECIPIENT edge). However, the learner wrote about a picture where the father was missing. Therefore, the context information "the father is the agent of the 'missing' action" can be used to guide the parser to the appropriate analysis (Figure 4b).

8 Limitations

The main limitation of the approach is that it is only of benefit when strong context information is available (while writing and parsing). For example, if the topic of an essay is the only available context information, parsing would not profit from context integration. However, since the interpretation of learner utterances without context information is not reliable in general, Ott et al. (2012) recommend to collect language learner data with explicit task contexts.

Another disadvantage of the approach is its dependence on the verb of the sentence. If the link between the verb and the action in the A-Box cannot be established, the context information cannot be used to influence attachments to the verb. This could happen, e. g., if the verb is misspelled or inflected incorrectly or if it is mistaken for another verb. If the verb is missing completely, no verb-related error diagnoses such as case errors can be derived from the parser's output.

To make the matching of words to individuals more robust in general, a component needs to be added to the scoring mechanism for selecting referents in the A-Box (Section 6): This component could model typical errors such as false friends and dictionary errors and it could also deal with spelling errors more accurately, e. g., using an approach similar to King and Dickinson (2014).

If the content of the A-Box diverges from what the learner wants to express, integrating such context information either has no influence on syntactic parsing or, worse, the syntactic structure does not reflect the learner's intention. To mitigate these problems, several A-Boxes can be defined, and sentences will be parsed with each model separately. The model which yields the syntactic structure with the highest score can then be chosen as the most appropriate one.

9 Conclusions and Outlook

We have shown that the context of an utterance can influence the diagnosis of grammatical errors in several ways. Cases where the syntactic structure and the error diagnoses differ depending on the context have been identified and systematically categorized into confusion classes. We have proposed to use a model for integrating context information into syntactic parsing of language learner utterances to resolve these error-induced ambiguities.

In future work, we will evaluate this approach by parsing erroneous as well as well-formed sentences to find out whether context integration benefits parsing of faulty sentences and whether it deteriorates parsing of error-free sentences. For this purpose, we have collected texts written by learners of German, which describe the content of picture stories.

[5]TurboParser was trained on the first 100 000 sentences of part A of the Hamburg Dependency Treebank (Foth et al., 2014), the largest genuine dependency treebank for German.

References

Lene Antonsen, Saara Huhmarniemi, and Trond Trosterud. 2009. Constraint Grammar in Dialogue Systems. In Eckhard Bick, Kristin Hagen, Kaili Müürisep, and Trond Trosterud, editors, *Proceedings of the NODALIDA 2009 workshop Constraint Grammar and robust parsing*, volume 8 of *NEALT Proceedings Series*, pages 13–21.

Christopher Baumgärtner, Niels Beuck, and Wolfgang Menzel. 2012. An Architecture for Incremental Information Fusion of Cross-Modal representations. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 498–503.

Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein, 2004. *OWL Web Ontology Language Reference*. World Wide Web Consortium (W3C). `http://www.w3.org/TR/2004/REC-owl-ref-20040210/`.

Emily M Bender, Dan Flickinger, Stephan Oepen, Annemarie Walsh, and Timothy Baldwin. 2004. Arboretum: Using a precision grammar for grammar checking in CALL. In *Proceedings of InSTIL/ICALL 2004 – NLP and Speech Technologies in Advanced Language Learning Systems*, Venice.

Adriane Amelia Boyd. 2012. *Detecting and Diagnosing Grammatical Errors for Beginning Learners of German: From Learner Corpus Annotation to Constraint Satisfaction Problems*. Ph.D. thesis, Ohio State University.

Thorsten Brants, Wojciech Skut, and Hans Uszkoreit. 1999. Syntactic Annotation of a German Newspaper Corpus. In *Proceedings of the ATALA Treebank Workshop*, pages 69–76, Paris, France.

Christian Fortmann and Martin Forst. 2004. An LFG Grammar Checker for CALL. In *Proceedings of InSTIL/ICALL 2004 – NLP and Speech Technologies in Advanced Language Learning Systems*, Venice.

Kilian Foth and Wolfgang Menzel. 2006. Hybrid Parsing: Using Probabilistic Models as Predictors for a Symbolic Parser. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 321–328, Sydney.

Kilian Foth, Michael Daum, and Wolfgang Menzel. 2004. A broad-coverage parser for German based on defeasible constraints. In *KONVENS 2004, Beiträge zur 7. Konferenz zur Verarbeitung natürlicher Sprache*, pages 45–52, Wien.

Kilian Foth, Wolfgang Menzel, and Ingo Schröder. 2005. Robust Parsing with Weighted Constraints. *Natural Language Engineering*, 11(1):1–25.

Kilian A. Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. 2014. Because Size Does Matter: The Hamburg Dependency Treebank. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Language Resources and Evaluation Conference 2014*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Kilian A. Foth, 2006. *Eine umfassende Constraint-Dependenz-Grammatik des Deutschen*. Fachbereich Informatik, Universität Hamburg. URN: urn:nbn:de:gbv:18-228-7-2048.

Michael Hahn and Detmar Meurers. 2012. Evaluating the Meaning of Answers to Reading Comprehension Questions: A Semantics-Based Approach. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 326–336, Montréal, Canada. Association for Computational Linguistics.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid.

Trude Heift. 2003. Multiple learner errors and meaningful feedback: A challenge for ICALL systems. *CALICO journal*, 20(3):533–548.

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

Levi King and Markus Dickinson. 2014. Leveraging Known Semantics for Spelling Correction. In *Proceedings of the third workshop on NLP for computer-assisted language learning at SLTC 2014, Uppsala University*, pages 43–58.

Julia Krivanek and Detmar Meurers. 2011. Comparing Rule-Based and Data-Driven Dependency Parsing of Learner Language. In *Proceedings of the International Conference on Dependency Linguistics (Depling 2011)*, pages 128–132.

Christine Köhn and Wolfgang Menzel. 2015. Towards parsing language learner utterances in context. Technical report, Fachbereich Informatik, Universität Hamburg. URN: urn:nbn:de:gbv:18-228-7-212.

André Martins, Noah Smith, and Eric Xing. 2009. Concise Integer Linear Programming Formulations for Dependency Parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 342–350, Suntec, Singapore. Association for Computational Linguistics.

André Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the Turbo: Fast Third-Order Non-Projective Turbo Parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617–622, Sofia, Bulgaria. Association for Computational Linguistics.

Patrick McCrae. 2009. A model for the cross-modal influence of visual context upon language processing. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 230–235.

Patrick McCrae. 2010. *A Computational Model for the Influence of Cross-Modal Context upon Syntactic Parsing*. Ph.D. thesis, Fachbereich Informatik, Universität Hamburg.

Wolfgang Menzel and Ingo Schröder. 1999. Error Diagnosis for Language Learning Systems. *ReCALL*, (special edition, May 1999):20 – 30.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Erygit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135.

Erich Ohser. 1993. Der Schmöker. In *50 Streiche und Abenteuer*, volume 1. Südverlag GmbH, Konstanz.

Niels Ott and Ramon Ziai. 2010. Evaluating Dependency Parsing Performance on German Learner Language. In Markus Dickinson, Kaili Müürisep, and Marco Passarotti, editors, *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories*, volume 9 of *NEALT Proceeding Series*, pages 175–186.

Niels Ott, Ramon Ziai, and Detmar Meurers. 2012. Creation and Analysis of a Reading Comprehension Exercise Corpus: Towards Evaluating Meaning in Context. In Thomas Schmidt and Kai Wörner, editors, *Multilingual Corpora and Multilingual Corpus Analysis*, Hamburg Studies in Multilingualism (HSM), pages 47–69. Benjamins, Amsterdam.

Veit Reuer. 2003. Error Recognition and Feedback with Lexical Functional Grammar. *CALICO Journal*, 20(3):497–512.

Margaret Rogers. 1984. On major types of written error in advanced students of German. *International Review of Applied Linguistics in Language Teaching*, 22(1):1–39.

Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. 2004. The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*.

Recent Initiatives towards New Standards for Language Resources

Gottfried Herzog[1], Ulrich Heid[2], Thorsten Trippel[3], Piotr Bański[4], Laurent Romary[5], Thomas Schmidt[4], Andreas Witt[4], Kerstin Eckart[6]

[1]Deutsches Institut für Normung e. V., Berlin,
gottfried.herzog@din.de
[2]Universität Hildesheim, [3]Universität Tübingen,
[4]Institut für Deutsche Sprache, Mannheim, [5]Inria, [6]Universität Stuttgart

1 Introduction

This poster is aimed at providing an overview of three ongoing initiatives towards language resource (LR) standards coordinated and initiated by the German mirror group of ISO TC 37/SC 4[1] within DIN[2] (Deutsches Institut für Normung):

- ISOTiger, an XML serialization of proposals for the syntactic annotation of text corpora;
- "Transcription of spoken language", a set of guidelines for transcribing spoken utterances;
- "Corpus Query Lingua Franca", a meta-standard for the comparison of the formal properties of corpus query languages.

Coordinated by German experts, these upcoming international standards[3] are all part of initiatives to standardize data formats and procedures for language resources internationally. The present poster is intended not only to inform about the ongoing work, but also to initiate a discussion with additional experts to reflect the interests of the community.

Standards for LRs in the framework of ISO TC 37 cover several types of resources (text corpora, lexicons, terminology collections). Actors in computational linguistics and language technology cooperate and thus need to exchange data and technologies using comparable methods and formats, cf. (Eckart and Heid, 2014). Most of the proposed standards are guidelines on a meta-level, describing properties of representation formats, instead of prescribing a format. Examples of these are the *Lexical markup framework* (LMF, ISO 24613:2008), the *Terminological markup framework* (TMF, ISO 16642:2003) and the *Linguistic annotation framework* (LAF, ISO 24612:2012) for lexical or terminological entry representation and for the representation of annotated corpora, respectively.[4]

In corpus annotation, more specific standards have been developed, for linguistic annotation at the levels of morphosyntax (MAF, ISO 24611:2012) and syntax (constituency and dependency, SynAF, ISO 24615-1:2014), as well as for certain aspects of semantic annotation, such as e.g. the annotation of temporal expressions (SemAF-Time, ISO-TimeML, ISO 24617-1:2012).

The proposed paper describes work towards standards which will be integrated into the existing standards portfolio: ISOTiger (ISO/DIS 24615-2) is building on top of SynAF; the standard for transcription of audio- or video-recorded spoken interactions (*Transcription of spoken language*, ISO/CD 24624) fills a gap in the domain of the preparation of spoken corpora; and the third proposal (CQLF, *Corpus Query Lingua Franca*, ISO/CD 24623-1) targets properties of tools for querying corpora.

2 ISOTiger

ISOTiger is an XML serialization of the SynAF meta-model. For this serialization, TIGER-XML (König et al., 2003), a widely applied corpus encoding format, which originated form the German TiGer project (Brants et al., 2004), was enhanced to meet the SynAF requirements for a generic exchange format for syntactic annotations. This includes independence from a specific theoretical orientation or annotation scheme: there shouldn't e.g. be any preferences whether the annotation consists of constituency trees or dependency graphs, or whether the encoded information results from a deep or a shallow analysis. ISOTiger fits in with existing serializations for other annotation

[1]International Organization for Standardization, Technical Committee 37, Subcommittee 4: http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=297592

[2]http://www.nat.din.de/cmd?level=tpl-untergremium-home&committeeid=54739043&languageid=de&bcrumblevel=2&subcommitteeid=63074672

[3]They are now progressing to the level of "Draft International Standard", the last feedback option before publication.

[4]The appendix contains a reference list of all standards discussed in this paper.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology, pages 154–156, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

layers: morpho-syntactic annotations encoded according to a MAF serialization naturally constitute the leaves of ISOTiger-encoded syntax trees in a standoff annotation. Moreover, we are discussing to use the full power of feature structures, cf. (FSR, ISO 24610-1:2006), in ISOTiger, cf. (Bosch et al., 2014). Similar to LAF and MAF, SynAF separates the structure of the annotations from the semantics of the annotation categories, thus it is possible within ISOTiger to link elements of tagsets to external data categories describing their semantics, cf. (ISO 12620:2009).

3 Transcription of spoken language

The standard on *Transcription of spoken language* is motivated, similar to the corpus representation standards, by the need to compare, interchange and possibly combine transcriptions of spoken language; this also concerns tool environments for the creation, editing, publication and exploration (e.g. query) of transcribed data. The standard is based on a comparative study of state of the art tools and their formats, and it is compatible with widely used transcription systems. The standard is being developed in cooperation with TEI proposals in the field, cf. (Schmidt, 2011).
It addresses metadata (briefly, as more standards proposals for this domain are available in CMDI (ISO 24622-1:2015) and from the TEI), as well as the macro- and microstructure of transcriptions. The macrostructure involves the timeline, as well as single or grouped utterances and elements outside utterances (e.g. `<pause>` and `<incident>` items).
The microstructure proposals deal in depth with the annotation of tokens, pauses, audible or visible non-speech events, punctuation, as well as units above and below the level of utterances. It also includes recommendations concerning the handling of uncertain cases, alternatives, incomprehensible or omitted passages. The appendices contain an ODD specification and a fully encoded example.

4 CQLF: Corpus Query Lingua Franca

CQLF proposes a standardized metamodel for classifying the data models underlying different corpus query languages (=QLs). It distinguishes three levels of QL complexity and thereby opens up a space of properties of QLs. The first level covers query systems for linear annotation, i.e. plain text or simple annotations to segments. Level 2 in addition involves complex annotations, either hierarchical (as in constituent structures) or dependency-like. Level 3 adds concurrent annotations, i.e. cases where a given phenomenon has been annotated in multiple ways which may overlap, be intersecting or even in conflict. The current part I will be complemented by an ontology of QL features, guidelines for the development of customized QLs (part II), as well as an analysis of QLs for multimodal and parallel corpora (part III). The specification provides general guidelines on a rudimentary classification of QLs, together with several examples in the annex.

The poster will present key elements of the three initiatives; all of them are thoroughly documented, and interested parties should not hesitate to contact the German experts on the DIN committees with comments and suggestions to the proposals. Work on all three proposals will continue in 2015 and early 2016, and at least for CQLF over a longer time frame.

References

Sonja Bosch, Kerstin Eckart, Gertrud Faaß, Ulrich Heid, Kiyong Lee, Antonio Pareja-Lora, Laurette Pretorius, Laurent Romary, Andreas Witt, Amir Zeldes, and Florian Zipser. 2014. From <tiger2> to ISOTiger – Community Driven Developments for Syntax Annotation in SynAF. In Verena Henrich, Erhard Hinrichs, Daniël de Kok, Petya Osenova, and Adam Przepiórkowski, editors, *Proceedings of the Workshop on Treebanks and Linguistic Theories*.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620.

Kerstin Eckart and Ulrich Heid. 2014. Resource interoperability revisited. In Josef Ruppenhofer and Gertrud Faaß, editors, *Proceedings of KONVENS*, volume 1, pages 116–126. Universität Hildesheim.

Esther König, Wolfgang Lezius, and Holger Voormann, 2003. *TIGERSearch 2.1 User's Manual. Chapter V*. IMS, Universität Stuttgart.

Thomas Schmidt. 2011. A TEI-based approach to standardising spoken language transcription. *Journal of the Text Encoding Initiative*, Issue 1. [Online], `http://jtei.revues.org/142`.

Appendix A. Reference list of mentioned standards and standard proposals

ISO 16642:2003	Computer applications in terminology – Terminological markup framework
ISO 24610-1:2006	Language resource management – Feature structures – Part 1: Feature structure representation
ISO 24611:2012	Language resource management – Morpho-syntactic annotation framework (MAF)
ISO 24612:2012	Language resource management – Linguistic annotation framework (LAF)
ISO 24613:2008	Language resource management – Lexical markup framework (LMF)
ISO 24615-1:2014	Language resource management – Syntactic annotation framework (SynAF) – Part 1: Syntactic model
ISO/DIS 24615-2	Language resource management – Syntactic annotation framework (SynAF) – Part 2: XML serialization (ISOTiger)
ISO 24617-1:2012	Language resource management – Semantic annotation framework (SemAF) – Part 1: Time and events (SemAF-Time, ISO-TimeML)
ISO 24622-1:2015	Language resource management – Component Metadata Infrastructure (CMDI) – Part 1: The Component Metadata Model
ISO/CD 24623-1	Language resource management – Corpus Query Lingua Franca (CQLF) – Part 1: Metamodel
ISO/CD 24624	Language resource management – Transcription of spoken language

Sentiment Uncertainty and Spam in Twitter Streams and Its Implications for General Purpose Realtime Sentiment Analysis

Nils Haldenwang
University of Osnabrück, Germany
nils.haldenwang@uos.de

Oliver Vornberger
University of Osnabrück, Germany
oliver@uos.de

Abstract

State of the art benchmarks for Twitter Sentiment Analysis do not consider the fact that for more than half of the tweets from the public stream a distinct sentiment cannot be chosen. This paper provides a new perspective on Twitter Sentiment Analysis by highlighting the necessity of explicitly incorporating uncertainty. Moreover, a dataset of high quality to evaluate solutions for this new problem is introduced and made publicly available[1].

1 Introduction

As a field of research Twitter Sentiment Analysis has gained much attention recently. For a multitude of applications such as sales prediction (Asur and Huberman, 2010), stock market prediction (Bollen et al., 2011) or political debate analysis (Diakopoulos and Shamma, 2010) it has been shown to generate practical value. Twitter Sentiment Analysis denotes the task of assigning a given tweet a sentiment label of either *positive* or *negative* and is an integral part of many practical applications. Few methods consider *neutral* as a third class. However, defining a neutral class is a hard task. Pak and Paroubek (2010) for example label tweets of popular news sites as neutral. This assumption is not always true. The headline "Multiple children were killed in the attack." would be labeled as *negative* by most human labelers. Thus, we propose an alternative approach to this problem. Its basic idea is the explicit incorporation of sentiment uncertainty.

2 The State of the Art and Its Shortcomings

SemEval-2014 Task 9 (Rosenthal et al., 2014) provides a widely used state of the art benchmark for Twitter Sentiment Analysis and compares the performance of many current approaches. From a dataset collected from January 2012 to January 2013, popular topics have been extracted through identification of frequently mentioned named entities. Only tweets scoring above a certain polarity threshold determined by a sentiment lexicon were considered to ensure the inclusion of a sentiment. The labels included in the dataset are *positive*, *negative* and *neutral*, determined by a majority vote of five labelers who were told to vote for the sentiment they perceive as strongest, when in doubt. This assigns tweets to the classes *positive* and *negative* which do not carry a distinct sentiment. Methods performing well on this dataset are shown to be able to distinguish between positive and negative sentiment under the assumption that all tweets can be assigned one of these labels. Moreover, all test tweets include popular named entities of the time. As the authors themselves noted: The dataset is biased. Moreover, the majority vote along with the treatment of ambiguity adds noise to the dataset. While providing a dataset of high quality for the desired purpose, the general composition of the public Twitter stream is not represented by the dataset. Hence, only part of the problems arising in practical analysis of the live stream are addressed with the related research.

3 A General Purpose Dataset

When analysing the Twitter stream we are interested in the "Electronic Word of Mouth"(Jansen et al., 2009), i.e. the personal opinions of private Twitter users. While labeling tweets, we noticed that a relatively high percentage of tweets are spam, advertising or marketing messages which we are not interested in. Those tweets shall be labeled *spam*. Moreover, it became obvious that for the remaining tweets only a small fraction can be distinctly labeled as *positive* or *negative*. The remaining tweets may still include polarity and can often

[1] http://project2.cs.uos.de/TweeDOS

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 157–159, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

not be labeled *neutral* while being neither *positive* nor *negative*. Hence, we propose the new category *uncertain*. Tweets labeled as *neutral* can be assigned to the class *uncertain* too, as they provide no additional information for sentiment analysis and can be treated in the same way as tweets of *uncertain* sentiment. This approach reduces the noise for the sentiment bearing classes which is a desirable feature if political or business decisions are supposed to be supported by the analysis results.

To acquire a representative view on the label composition of the public Twitter stream, we randomly sampled our dataset from a collection of about 43 million tweets with their creation dates ranging from June 2012 to August 2013 to minimize topical bias. Each tweet was labeled by two human labelers who had to assign it one of the labels *positive*, *negative*, *uncertain* or *spam*. In total 14506 tweets have been labeled by 27 labelers. The labelers consisted of master's students from the University of Osnabrück, Germany and researchers from our group.

The distribution of labels is shown in figure 1. There is a total of 9356 (64.5% of total tweets labeled) tweets to which both human labelers assigned the same label. Of these tweets 15% are *spam* and 55% are labeled *uncertain*. A definite sentiment label could only be assigned to 30% of tweets with 13% being positive and 17% being negative.

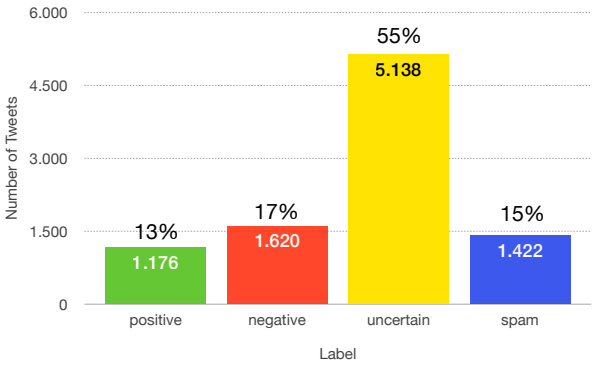


Figure 1: Distribution of labels for tweets which both labelers agreed upon.

These results provide evidence for our claim that one has to deal with uncertainty in sentiment analysis when working with the public Twitter stream.

To assess the inter annotator agreement we computed Fleiss' Kappa (Fleiss, 1971) resulting in a value of $\kappa = 0.45$ which can be interpreted as *moderate agreement* (Landis and Koch, 1977). At first sight this value seems to be rather low but when considering the disagreement matrix shown in table 1 the claim of the necessity to deal with uncertainty is further strengthened.

	positive	negative	uncert.	spam
positive	1176	106	1666	143
negative		1620	2263	58
uncert.			5138	914
spam				1422

Table 1: Disagreement matrix showing the absolute number of label combinations.

Labelers seem to have a very good understanding of what distinguishes the classes *positive* and *negative*, only 106 tweets have been assigned both these labels. The disagreement for *positive/spam* and *negative/spam* is of similar or even smaller magnitude. Looking at these tweets we noticed that the disagreement is mainly related to misunderstanding of the labeling instructions or probably accidentally clicking the wrong label. Hence, these tweets should be omitted from the test set when evaluating methods for reliable Twitter Sentiment Analysis.

However, the disagreement between *positive/negative* and *uncertain* is relatively large. These tweets make up about 76% of the tweets to which the two labelers assigned different labels. This indicates that in many cases not even two humans can agree upon whether a tweet contains a distinct sentiment or should be labeled *uncertain*. Systems aiming to perform reliable sentiment analysis of the public Twitter stream should be able to deal with these tweets. While not strictly belonging to the category *uncertain* they should still be labeled as such or at least not be considered for sentiment analysis. Another possible approach can be to interpret them as *rather positive* or *rather negative*, depending on the amount of reliability the respective application requires.

Moderate disagreement (914 tweets) can be noted for the classes *uncertain* and *spam*. Since these tweets may still contain useful information in the sense of answering the question "What do people talk about?" they probably should not be considered spam. However, they also should not be assigned a sentiment. A system labeling these as *uncertain* will still produce reliable results with regard to sentiment analysis.

As a first approach one can make use of just the tweets with two identical labels to asses methods for reliable sentiment analysis of the public Twitter stream. However, it should be considered that in practice the tweets upon which the labelers disagreed can also appear in the stream and have to be handled to provide reliable sentiment results. To enable researchers to develop systems which meet all the aforementioned requirements the complete dataset including the tweets disagreed upon is publicly available.

4 Conclusion and Outlook

When performing analysis on the public live stream of Twitter with regard to sentiment, it needs to be considered that more than half of the tweets cannot be assigned a distinct sentiment. These tweets have to be filtered or explicitly dealt with before sentiment analysis takes place. Moreover, one has to deal with spam tweets. Spam adds unwanted noise by polluting topics with artificially injected tweets. Most of the work on spam detection on Twitter focusses on catching the users generating the spam by looking at the accounts' behaviour over time (Grier et al., 2010; Lin and Huang, 2013). When performing realtime analysis, a given tweet has to be determined to be spam or no spam by looking at its content and meta data only as there is no time to examine the author's account in detail. New methods have to be developed which are able to deal with sentiment uncertainty and spam if reliable representations of the public opinion are to be acquired from the Twitter stream. The dataset presented in this paper can be used to develop and evaluate methods for reliable Twitter Sentiment Analysis.

References

Sitaram Asur and Bernardo A Huberman. 2010. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 492–499. IEEE.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Nicholas A Diakopoulos and David A Shamma. 2010. Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1195–1198. ACM.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. 2010. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 27–37. ACM.

Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Po-Ching Lin and Po-Min Huang. 2013. A study of effective features for detecting long-surviving twitter spam accounts. In *Advanced Communication Technology (ICACT), 2013 15th International Conference on*, pages 841–846. IEEE.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the 7th International Conference on Language Resources*, volume 10, pages 1320–1326.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Visuelle Mehrsprachigkeit in der Metropole Ruhr: Aufbau und Funktionen der Bilddatenbank „Metropolenzeichen“

Tirza Mühlan
Universität Duisburg-Essen
Institut für Germanistik
Universitätsstraße 12
45141 Essen
tirza.muehlan@uni-due.de

Frank Lützenkirchen
Universitätsbibliothek Duisburg-Essen
Universitätsstraße 9-11
45141 Essen
frank.luetzenkirchen@uni-due.de

Abstract

Sichtbare Mehrsprachigkeit zeigt sich im öffentlichen Raum auf Hinweis-, Informations- und Geschäftsschilden sowie auf Graffiti und Aufklebern und gibt Aufschluss über Migration, Kultur- und Konsumtourismus. Das von MERCUR (Mercator Research Center Ruhr) geförderte Projekt „Metropolenzeichen: Visuelle Mehrsprachigkeit in der Metropole Ruhr“ ist eine als Querschnittstudie für die Städte Duisburg, Essen, Bochum und Dortmund angelegte Untersuchung und behandelt die Präsenz sichtbarer Mehrsprachigkeit bezogen auf den öffentlichen Raum der Metropole Ruhr als bundesweit wichtigste Metropole für Arbeitsmigration. Grundlage des Projekts bildet ein Korpus geokodierter Bilddaten, die in einer digitalen Bilddatenbank archiviert und mit einer geographischen Karte verlinkt sind. Die ca. 25.000 Bilder sind formal und inhaltlich erschlossen, d.h. für die weitere Analyse nach verschiedenen Kategorien (Sprache, Diskurstyp, Name, Erscheinungsform u.a.) verschlagwortet. Entsprechend dieser inhaltlichen und formalen Verschlagwortungskriterien können die Bilddaten quantitativ und qualitativ untersucht werden.

1 Visuelle Mehrsprachigkeit in der Metropole Ruhr

Der Vortrag stellt das digitale, geokodierte Bilddaten-Korpus „Metropolenzeichen“ vor, das in einer Bilddatenbank archiviert ist und die Basis für das von MERCUR (Mercator Research Center Ruhr) geförderte Forschungsprojekt „Metropolenzeichen: Visuelle Mehrsprachigkeit in der Metropole Ruhr“ bildet. Das von Linguisten, Integrationsforschern und Stadtsoziologen der Universitäten Duisburg-Essen und Bochum betriebene Forschungsprojekt beschäftigt sich mit der Präsenz sichtbarer Mehrsprachigkeit im öffentlichen Raum der Metropole Ruhr. Visuelle Mehrsprachigkeit zeigt sich in Form von Informations-, Geschäfts- und Straßenschildern, aber auch in Graffitis und transgressiven Aufklebern. Sie steht in engem Zusammenhang mit Migration, Kultur- und Konsumtourismus sowie auch mit Regionalisierungstendenzen, d.h. der Inanspruchnahme kleinräumigerer kultureller Identifikationssymbole wie etwa regionalen Varietäten. In einem interdisziplinären und multiperspektiven Zugriff werden stadtsoziologische, sprachwissenschaftliche und integrationstheoretische Aspekte behandelt, d.h. die städteräumliche Verteilung, formale Ausgestaltung, funktionale Bedeutung und gesellschaftliche Bewertung visueller Mehrsprachigkeit behandelt.

Die Untersuchung visueller Mehrsprachigkeit stellt ein junges, stark diskutiertes internationales Forschungsfeld dar, das unter dem Etikett der „linguistic landscapes“ firmiert und die Sichtbarkeit, Verteilung und Situierung von geschriebener Sprache im öffentlichen Raum thematisiert. Zentrale Annahme ist, dass öffentlich sichtbare Mehrsprachigkeit relevante Hinweise auf die Kultur des Zusammenlebens in einer mehrsprachigen Gesellschaft liefert (Cenoz/Gorter, 2006). Neuartig ist die systematische Untersuchung einer Me-

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 160–161, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

tropolregion, die nicht durch offizielle Mehrsprachigkeit, sondern durch Migration bedingte Mehrsprachigkeit gekennzeichnet ist.

Im Rahmen einer Querschnittstudie wurden in jeweils zwei Stadtteilen der vier Städte Duisburg, Essen, Bochum und Dortmund systematisch Bilddaten gesammelt und in eine Bilddatenbank importiert. Bemerkenswert ist die Zweiteilung dieser Städte durch den sog. „Sozialäquator A 40" (Kersting et al., 2009), der die Städte der Metropole Ruhr in „ethnisch divers und weniger divers", „arm und weniger arm" und „gebildet und weniger gebildet" einteilt. Diese ethnisch-soziale Segregation lässt eine besonders interessante vergleichende Analyse der Bilddaten zu, denn ein Ziel des Projektes ist es, symbolische Sichtbarkeit kultureller Diversitäten im Kontext ethnisch-sozialer Siedlungsstrukturen und funktionsräumlicher Aspekte zu bestimmten. Aber auch die gesellschaftliche Bedeutung von Mehrsprachigkeit als Index für Beheimatung und gesellschaftlicher Anerkennung soll herausgearbeitet werden.

2 Aufbau und Funktion der Bilddatenbank „Metropolenzeichen"

Die insgesamt 25.595 in den Stadtteilen südlich und nördlich der A 40 aufgenommenen Bilddaten der Ruhr-Städte wurden in eine Bilddatenbank importiert, die auf Basis der Open Source Repository-Software „MyCoRe" implementiert ist. Sie bietet folgende Grundfunktionen: dezentraler, passwortgeschützter Zugriff, Serienimport, Eingabe- und Suchmaske, integriertes Image-Viewer-Modul (inkl. Zoom-Funktion), Galerie- und Detailansicht mit Metadatenliste, Filter- und Sondierfunktion.

Für die quantitative und qualitative linguistische Analyse wurden die Bilddaten nach inhaltlichen und formalen Kriterien verschlagwortet. Dabei wurden folgende Kategorien bedient: Ort (Stadt, Stadtteil, Einrichtung wie z.B. Bahnhof, Bürgerbüro), Diskurstyp (z.B. infrastrukturell, kommerziell), Sprache (z.B. Deutsch, Englisch, Türkisch, Polnisch, Nonstandard), Anzahl der Sprachen (z.B. monolingual, bilingual), Name (z.B. Person, Institution, Toponym), Informationsmanagement (z.B. komplett, teilweise), Erscheinungsform (z.B. Aufkleber, Schild), semiotische Kodierung (z.B. Bild, Text), Größe (z.B. –1 m², –10 m²) und Typografie (z.B. Arabisch, Latein). Bezüglich der Verschlagwortungspraxis können einige der Kategorien nur einem Typ zugewiesen werden (Ort, Diskurstyp, Erscheinungsform, semiotische Kodierung, Anzahl der Sprachen), die Sprachwahl und die Kategorie „Name" kann dagegen mehrfach vergeben werden, je nachdem wie viele Sprachen oder Namen sich auf einem Item befinden.

Auf der anderen Seite stellen die Kategorien „Name" sowie „Informationsmanagement" fakultative Kategorien dar, die nur vergeben werden, wenn auf den Items z.B. ein Personenname oder ein Firmenname bzw. eine erweiterte, komplette oder teilweise übersetzte Information vorliegt.

Neben der formalen und inhaltlichen Erschließung soll auch die Geovisualisierung der Daten dargestellt werden. Dazu wurden die Geokoordinaten der Bilder beim Import aus den EXIF-Daten extrahiert. Die Ergebnismenge jeder Suchanfrage kann mittels des OpenLayers Frameworks auf einer zoombaren Karte visualisiert werden, die so einen Einblick in die symbolische Sichtbarkeit und lokale Verteilung kultureller Diversität gibt.

Bibliographie

Peter Backhaus. 2007. *Linguistic Landscape. A Comparative Study of Urban Multilingualism in Tokyo*. Clevedon, Buffalo, Toronto.

Jasone Cenoz und Durk Gorter. 2006. Linguistic landscape and minority languages. *International Journal of Multilingualism* 3(1): 67-80.

Ibrahim Cindark und Evelyn Ziegler. i. Dr.: Mehrsprachigkeit im Ruhrgebiet: Zur Sichtbarkeit sprachlicher Diversität in Dortmund. Ptashnyk, Stefaniya et al. (Hg.): *Gegenwärtige Sprachkontakte im Kontext der Migration.* Heidelberg: Winter.

Volker Kersting et al.. 2009. Die A 40 – Der „Sozialäquator" des Ruhrgebiets. *Atlas der Metropole Ruhr*. Essen: 142-145.

Ron Scollon und Suzie Wong Scollon. 2003. *Discourses in Place: Language in the Material World*. London.

Elena Shohamy. 2006. *Hidden agendas and new approaches.* New York.

IDaSTo – Ein Tool zum Taggen und Suchen in historischen Paralleltexten

Rahel Beyer
Institut für luxemburgische Sprach- und Literaturwissenschaft
Universität Luxemburg
L-4366 Esch-Belval, Luxemburg
rahel.beyer@uni.lu

Abstract

Ein integriertes Datenbank-, Such- und Tagging-Tool (IDaSTo) wird vorgestellt, das sich besonders für Variablenanalysen, für Paralleltexte und für diachronische Untersuchungen eignet. Relevante Kategorien bzw. Variablen können individuell definiert, Tags frei im Text und auf verschiedenen Wegen gesetzt und ihre Häufigkeiten in den verlinkten Statistiken direkt abgerufen werden.

1 Einleitung[1]

Die historische Soziolinguistik greift zunehmend auf (große) Korpora zurück und führt an ihnen variablenanalytische Untersuchungen u.a. im Kontext der Erforschung der Sprachstandardisierung durch (vgl. z.B. Durrell et al., 2008; Elspaß, 2005; Vosters et al., 2012). Zu den Arbeitsschritten gehört es, Varianten in den Texten ausfindig zu machen, diese zu sammeln und auszuzählen. Von besonderem Interesse sind dabei die Entwicklungen von Strukturen auf allen sprachlichen Ebenen im Laufe von teils großen Zeiträumen und ihr soziohistorischer Kontext. Dementsprechend wichtig ist es, das Gesamtkorpus auf der Grundlage von Subperioden zu analysieren und ggf. Ergebnisse verschiedener Textsorten voneinander getrennt zu halten, jedoch jeweils die übergreifenden Verhältnisse präsent zu haben. Zu einem solchen Kontext gehören auch funktionale Aspekte bzw. Metadaten den Text als Ganzes bzw. seine äußeren Merkmale betreffend wie Entstehungsjahr, Drucker(haus), Unterzeichner u.ä. Diese Rückbindung ist v.a. dann von Relevanz, wenn es um Sprachwandel und die Auswirkungen von und Wechselwirkungen zwischen sprachlichen und gesellschaftspolitischen Faktoren geht.

Außerdem gewinnen in der historischen Soziolinguistik Paralleltextkorpora immer mehr an Aufmerksamkeit (vgl. Claridge, 2008). Diese sind für Untersuchungen in Sprachkontaktkontexten umso attraktiver als dass gerade auf der Ebene von grammatischen Phänomenen ein Nachweis von kontaktinduzierten Veränderungen immer wieder als problematisch diskutiert wird (vgl. Heine, 2009). In Paralleltexten lassen sich direkte Abgleiche zwischen zwei oder mehreren Sprachversionen vornehmen und auf diese Weise z.B. Konvergenzen erkennen. Aber auch im Hinblick auf funktionale Erkenntnisinteressen bieten sie vielversprechende Ansatzpunkte für Untersuchungen. Wurden beide Sprachversionen auf ein Papier gedruckt, so ist z.B. ihre Abfolge von Bedeutung, d.h. welcher Text links (bei horizontaler Anordnung) bzw. oben (bei vertikaler Anordnung) steht.

Aus der beschriebenen Vorgehensweise bei der Variablenanalyse lässt sich der Mehrwert maschineller Unterstützung unschwer ableiten. Gerade für eine datengesteuerte und explorative Identifizierung von Variablen bedarf es jedoch flexibler, nicht voreingestellter Annotationskategorien. Bestehende Annotationsprogramme fokussieren sich in der Regel auf spezifische sprachliche Aspekte bzw. lassen sich toolabhängig für die Aufbereitung nur bestimmter sprachlicher Phänomene bzw. vordefinierter Ebenen (mittels vorinstallierter Tagsets) einsetzen. Web-Anno (Yimam et al., 2013) bietet zwar auch die Einrichtung von benutzerdefinierten Annotationsebenen an, allerdings beschränkt sich die Funktionalität – wie bei vielen Annotationsprogrammen – auf eine Anreicherung mit linguistischen Informationen innerhalb von Texten. Ihre Auswertung muss dementsprechend extern

[1] Ich danke den anonymen Gutachtern sowie Instituts- und Projektkkollegen für hilfreiche Kommentare und Hinweise.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology, pages 162–169, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

geschehen.[2] Eine variablenanalytische Untersuchung bedarf jedoch einer zentralen Verwaltung inklusive einer Übersicht der Phänomene (Variablen) und ihrer Varianten sowie der Häufigkeiten, von der aus zudem Belegstellen nach Bedarf aufgerufen werden können. Auch Standardmethoden der Korpuslinguistik wie Frequenzlisten und Konkordanzen sollten unmittelbar mit einem variablenanalytischen Tool verknüpft sein. Eine solche Kombination verschiedener Funktionen in einem integrierten Tool gibt es jedoch bislang nicht.[3] Die Suche in Paralleltexten bringt weitere Anforderungen mit sich, die von gängigen Konkordanzprogrammen ebenfalls nicht erfüllt werden. tlCorpus[4] z.B. kann weder sprachspezifisch suchen noch den Suchbegriff als Variante eines (sprachlichen) Merkmals auszeichnen. Außerdem ist das Filtern der Texte, die durchsucht werden sollen, aufwändiger und wenig flexibel.

Der vorliegende Beitrag stellt eine Anwendung vor, die nun den genannten Bedürfnissen Rechnung trägt. Dabei handelt es sich weniger um ein klassisches computerlinguistisches Tool (z.B. stehen weder Lemmatisierung noch Parts-of-speech- noch Morphologie-Tags o.ä. zur Verfügung), vielmehr wurde das Programm speziell auf variablenanalytisches Arbeiten zugeschnitten, bei dem abgrenzbare Konstruktionen und Phänomene in den Fokus genommen werden (z. B. orthografische Variation, afinite Nebensätze, der realisierte Kasus nach bestimmten Präpositionen oder Entwicklungen im Wortschatz). Zudem wurde es zu Beginn eines historisch-soziolinguistischen Projekts entwickelt, so dass eine zügige Einsatzmöglichkeit erforderlich war. Letzten Endes konnte eine pragmatische Lösung gefunden werden, die jedoch hochspezifiziert für diachrone Variablenanalyse an Paralleltexten ist. Gegenstand des besagten Projekts ist die Standardisierung des Deutschen in einem Mehrsprachigkeitskontext. Variation bzw. Variantenreduktion soll hier v.a. anhand von historischen, zweisprachigen Paralleltexten untersucht werden.[5] Durch die Nutzung parallel zur Entwicklung konnten konkrete Bedürfnisse ermittelt und im Programm berücksichtigt werden.

[2] Vgl. etwa die Annotation in CorA und die anschließend notwendige Importierung der Transkripte in ANNIS zur Durchsuchung und Visualisierung (Bollmann et al., 2014).
[3] Vgl. z.B. die Zusammenstellung unter https://www.linguistik.hu-berlin.de/institut/professuren/korpus-linguistik/links/software. Aus Platzgründen kann eine ausführliche Evaluation jedes der dort aufgeführten Tools an dieser Stelle nicht stattfinden.
[4] http://tshwanedje.com/corpus/.
[5] S. das Datenbeispiel in Abb. 2.

2 Beschreibung des Tools

Grundsätzlich handelt es sich um eine Mischung aus Datenbank, Suchprogramm und Tagging-Tool. Neben der Funktion als Datenbank, aus der entsprechend den eingegebenen Filterkriterien verschiedene Datensätze aufgerufen und angesehen werden können, sollten auch in den einzelnen Texten Tags frei verteilt werden, Tokens im Fließtext gesucht und die Suchergebnisse ebenfalls getaggt werden können.

Außerdem galt es zu berücksichtigen, dass das Projekt an verschiedenen Standorten bearbeitet wird. Dementsprechend wird eine web-basierte Architektur[6] verwendet, d.h. alle Daten sowie die Anwendung werden auf einem Server gespeichert, der für alle Benutzer über einen beliebigen Internetbrowser zugänglich ist. Dadurch kann problemlos von jedem Computer darauf zugegriffen werden, es bedarf keiner lokalen Installation und alle Benutzer arbeiten an derselben Version.

2.1 Datenbank

Von der Anmeldung gelangt man zunächst automatisch auf die Startseite, auf der alle Datensätze aufgelistet werden.[7] In dem konkreten Projekt[8] handelt es sich dabei um öffentliche Bekanntmachungen, die mehrheitlich als zwei, parallel auf einem Dokument angeordnete Sprachversionen vorliegen. Diese wurden zunächst mit einem Großformatscanner erfasst. Informationen über die Bilddateien sowie deren Metadaten (Signatur, Datum, Titel, Verantwortlicher und gebrauchte Sprach(en)) wurden ebenfalls in einer Tabellenkalkulation festgehalten. Anschließend wurden die Dateien manuell text-digitalisiert und in ein XML-Format überführt. Dabei blieb die Originaltextstruktur durch Taggen der essentiellen Merkmale wie Überschriften, Sprachenwechsel, Schriftarten (Französisch ist in Antiqua, Deutsch meistens in Fraktur) und Groß- und Kleinschreibung nach den internationalen Standards der Text Encoding Initiative (TEI)[9] erhalten.

Durch das Ausfüllen der Datenfelder am Anfang der Seite können jeweils bestimmte Datensätze herausgefiltert werden.

[6] Dabei wurde in PHP programmiert und auf das Webentwicklungs-Framework Symfony 2 bzw. für die Benutzerschnittstelle auf Bootstrap und jQuery zurückgegriffen.
[7] Hier wurde MariaDB als Datenbank-Verwaltungssystem eingebaut.
[8] S. auch http://infolux.uni.lu/standardization/.
[9] TEI Consortium (2015).

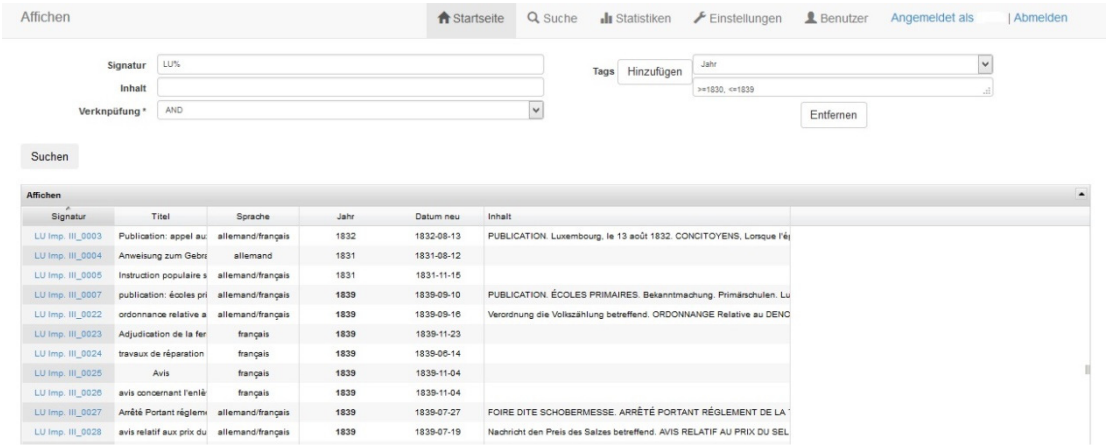


Abb. 1: Screenshot der Startseite

So können z.B. mithilfe des Wildcard-Zeichens ‚%' alle Dokumente mit demselben Signaturanfang, d.h. eines Subkorpus' selektiert und aufgelistet werden. Zur Filterung können jegliche Dokumentenmerkmale, d.h. Metadaten herangezogen werden, die in einem vorangegangenen Schritt definiert und ihre Ausprägungen für die einzelnen Dokumente notiert wurden.[10] Mithilfe von Vergleichsoperatoren können außerdem Zeitspannen ausgewählt werden. Ferner können mehrere Auswahlkriterien miteinander kombiniert werden. In Abbildung 1 bspw. wurden alle Dokumente der Signatur „LU%" aus den Jahren 1830-1839 gesucht.

2.2 Taggen in einzelnen Dokumenten

Durch Anklicken der Signatur öffnet sich das jeweilige Dokument in einer neuen Registerkarte. Hier sind der Original-Scan, die Metadaten zu dem Dokument (grundlegende Merkmale wie Signatur, Sprachen, Titel, Erstellungsjahr usw.) und der Text einzusehen. Werte von Metadaten können an dieser Stelle korrigiert oder neue Metadaten hinzugefügt, d.h. der Text als Gesamtes getaggt werden. Die Metadaten aus dem Kopf der Dokumente finden sich in den Spalten der Datenbank auf der Startseite wieder.[11]

Vor dem Hintergrund der zweisprachigen Ausgabe der Texte und des intendierten (punktuellen) Abgleichs der beiden Sprachversionen[12] war die Trennung und parallele Darstellung der beiden Sprachen von entscheidender Bedeutung. Dieser Schritt steigert die Leserfreundlichkeit und beschleunigt damit den Bearbeitungvorgang. Zu diesem Zweck wurden die Texte zuvor absatzweise auf einem N-Gram-basierten Verfahren automatisch kategorisiert,[13] d.h. in diesem Fall einer Sprache zugeordnet. Für den Fall, dass eine automatische Zuordnung nicht möglich war oder nicht der tatsächlich geschriebenen Sprache entspricht, können die einzelnen Absätze innerhalb des Tools von Hand korrigiert werden. Beide Sprachversionen können wahlweise untereinander in einer Spalte oder in getrennten Spalten nebeneinander, wie in Abbildung 2, angezeigt werden. Für die Auswahl gibt es eine entsprechende Schaltfläche in der rechten Leiste. In den Texten selbst werden per Mouseover auf die Token Links zu verschiedenen Wörterbüchern angeboten.

Vor allem aber können sämtliche Token (d.h. Wortformen sowie Interpunktionszeichen) des Textes ausgewählt und anschließend mit verschiedenen Informationen und hinsichtlich verschiedener Aspekte getaggt werden. Bezogen auf das korpuslinguistische bzw. variablenanalytische Vorgehen kann also das Vorkommen einer Variante eines relevanten sprachlichen Phänomens (Variable) mittels Tags dokumentiert werden (z.B. die <ä>-Variante von z.B. *Kerker*). Genauso können jedoch auch durch Aktivieren der Steuerungstaste mehrere Token gleichzeitig ausgewählt und somit mehrteilige Ausdrücke oder Phrasen, die z.B. eine bestimmte Spracheinstellung zum Ausdruck bringen (z.B. „unsere geliebte Muttersprache"), getaggt werden. Ein Tag ist folglich immer als Merkmal-Wert-Paar aufgebaut, bei dem der Tagname bzw. die Variable dem Merkmale entspricht und als Wert der

[10] Diese Meta-Daten können entweder aus einer Tabellenkalkulation importiert oder in IDaSTo eingegeben werden (s. Abschnitt 2.2).

[11] Die Anzeige der Spalten kann individuell nach Bedarf des Benutzers in den „Einstellungen" angepasst werden, d.h. Spalten mit Metadaten können hinzugefügt oder ausgeblendet werden.

[12] Zur Aufdeckung von Replikationen und Transferenzen.

[13] Der Algorithmus wurde basierend auf Cavnar und Trenkle (1994) gebildet.

konkrete Beleg bzw. die vorzufindende Variante eingesetzt wird. Es gibt keine vordefinierten oder obligatorischen Tags oder Tagkategorien; vielmehr können Tagname (z.B. „<ä>-<e>-Variation“ oder „Spracheinstellung“) und zugehöriger Wert (d.h. Variante, z.B. <ä> oder „emotionales Motiv“) individuell und flexibel erstellt werden. Da für den Wert ein freies Textfeld zur Verfügung steht, können z.B. auch notizartige Teilsätze als Werte eingegeben und die Tagfunktion kann als eine Art Lesezeichen eingesetzt werden. Bereits im Laufe des Projekts erstellte Tagkategorien können indes aus einem Dropdown-Menü ausgewählt werden. Außerdem wird der zuletzt vergebene Tag (d.h. Tagname und Wert) im Tagging-Fenster aufgeführt und kann direkt angewendet werden (s. Abbildung 3). Dabei kann jedes Token mit beliebig vielen Tags ausgezeichnet werden. Bereits ausgezeichnete Wortformen sind grün unterstrichen und bei ihrer Auswahl werden vergebene Tags inklusive Werten in der rechten Leiste angezeigt (vgl. Abbildung 2).

Nicht zuletzt kann für ein ausgewähltes Token direkt aus dem Dokument eine Suche erstellt werden, d.h. ein mit dem betreffenden Token als Suchbegriff, ihm zugewiesene Tags und zugeordnete Sprache vorausgefülltes Suchformular öffnet sich in einem neuen Tab.

Abb. 2: Screenshot der Dokumentenansicht

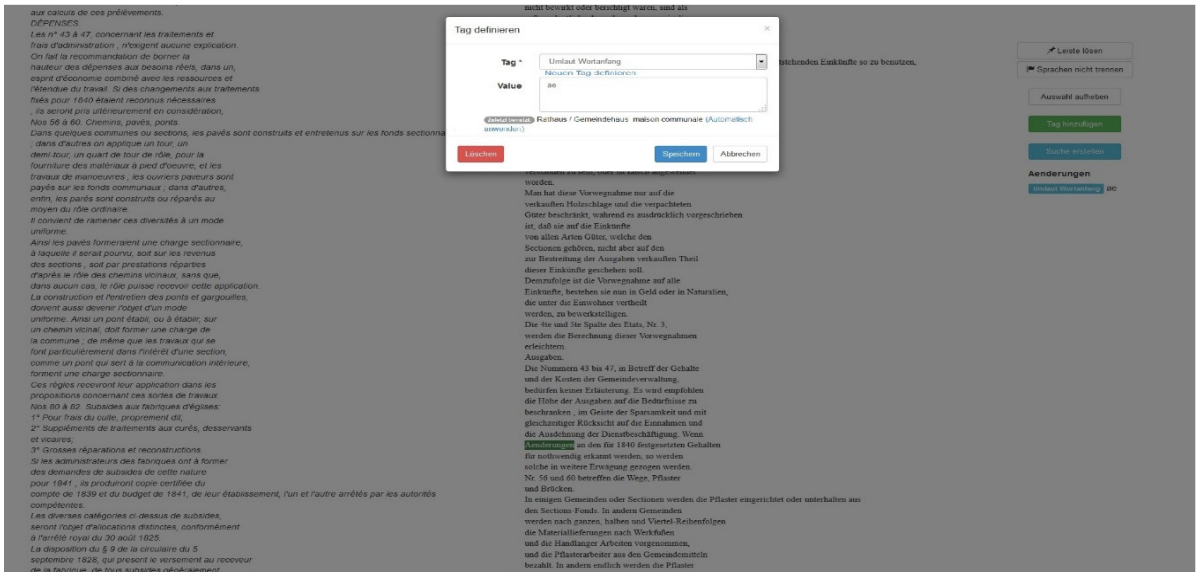


Abb. 3: Screenshot des Tagging-Fensters

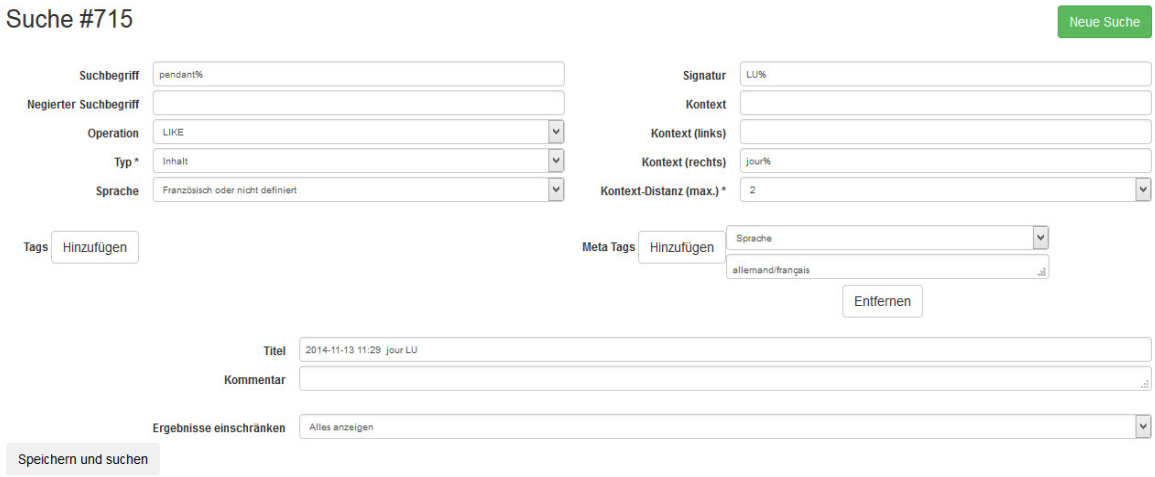


Abb. 4: Screenshot der Suchspezifizierungen

2.3 Das Suchmenü

Unter dem Menüpunkt „Suche“ werden zunächst alle bereits ausgeführten Suchen aufgelistet. Alternativ kann eine neue Suche gestartet werden.

Der Suchbereich für eine beliebige Zeichenkette kann hinsichtlich verschiedener Aspekte eingeschränkt bzw. präzisiert werden. Darunter befinden sich Sprache, Subkorpus (d.h. Signaturanfang), evtl. gespeicherte Tags sowie flexibel zuwählbare Metadaten. Bezüglich der Operationen stehen die exakte Suche nach einem String und die Suche mit vordefinierten Zeichen (LIKE [nur mit Wildcard-Zeichen ‚%‘] oder reguläre Ausdrücke) zur Auswahl. Des Weiteren kann man sich für verschiedene Inhaltstypen entscheiden („Inhalt“ [ignoriert sowohl Groß- und Kleinschreibung als allerdings auch Sonderzeichen, z.B. Umlaute], „Groß-/Kleinschreibung beachten“ sowie „Normalisierter Inhalt“).

Um über einzelne Wörter hinaus Wortfolgen ausfindig machen zu können, wurde eine Suche im (rechten, linken oder unspezifizierten) Kontext des eigentlichen Suchbegriffs implementiert. Diese Option wird relevant, wenn z.B. die Realisierungen bestimmter Mehrwortlexeme oder Nomen-Verb-Verbindungen überprüft werden sollen. Vorteil dieser Lösung ist, dass die Distanz zwischen Such- und Kontextbegriff über ein entsprechendes Datenfeld einzelfallspezifisch bestimmt werden kann. So kann z.B. nach der Konstruktion *pendant X jour* gesucht werden, wobei *pendant%* Suchbegriff ist und *jour%* im Kontext mit einer maximalen Distanz von zwei Wörtern zum Suchbegriff stehen soll (s. Abbildung 4).

Unter den Datenfeldern erscheinen nach Beendigung des Suchlaufs die Suchergebnisse in einem integrierten Fenster (s. Abbildung 5). Das Fenster selbst entspricht dem typischen Aufbau von Konkordanzprogrammen. Zusätzlich lassen sich auf der äußersten rechten Seite einzelne Suchergebnisse deaktivieren bzw. nach Deaktivierung bei Bedarf wieder aktivieren. Auf diese Weise können Suchresultate, die nicht dem relevanten Phänomen entsprechen, aussortiert und von einer weiteren Verarbeitung ausgeschlossen werden. Optional können in den „Einstellungen“ weitere Spalten mit Metadaten dazugeschaltet werden.

Die Suchergebnisse lassen sich über drei Wege weiterverarbeiten, d.h. taggen. Erstens kann jede der Belegstellen des Suchbegriffs direkt im Fenster mit den Suchresultaten einzeln ausgewählt und wie in Abschnitt 2.2 beschrieben getaggt werden. Zweitens gelangt man durch das Anklicken der Signatur zum Belegdokument, das beim Öffnen direkt zur Fundstelle des Suchbegriffs springt. Dort kann man ebenfalls nach demselben Verfahren Tags vergeben. Für den Fall, dass alle (nach der ‚Bereinigung‘ übriggebliebenen) Suchresultate denselben Tag und dazugehörigen Wert bekommen, kann die Option „Tag auf die Suchergebnisse anwenden“ gewählt werden. Beide Vorgänge können genauso für Meta-Tags durchgeführt werden.

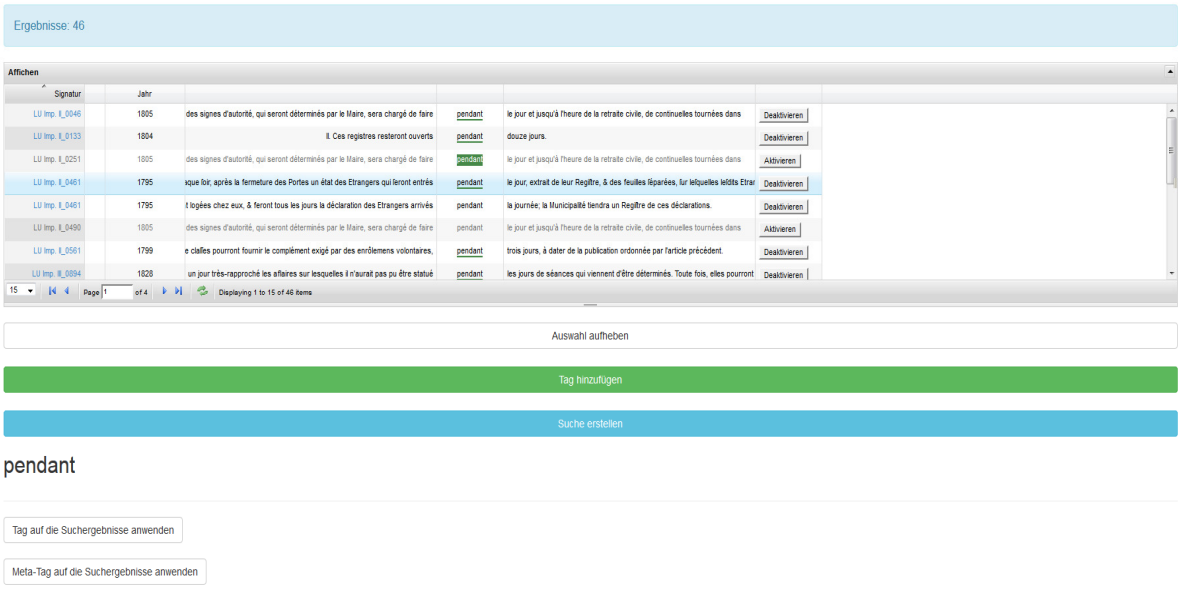


Abb. 5: Screenshot des integrierten Fensters mit der Auflistung der Suchergebnisse

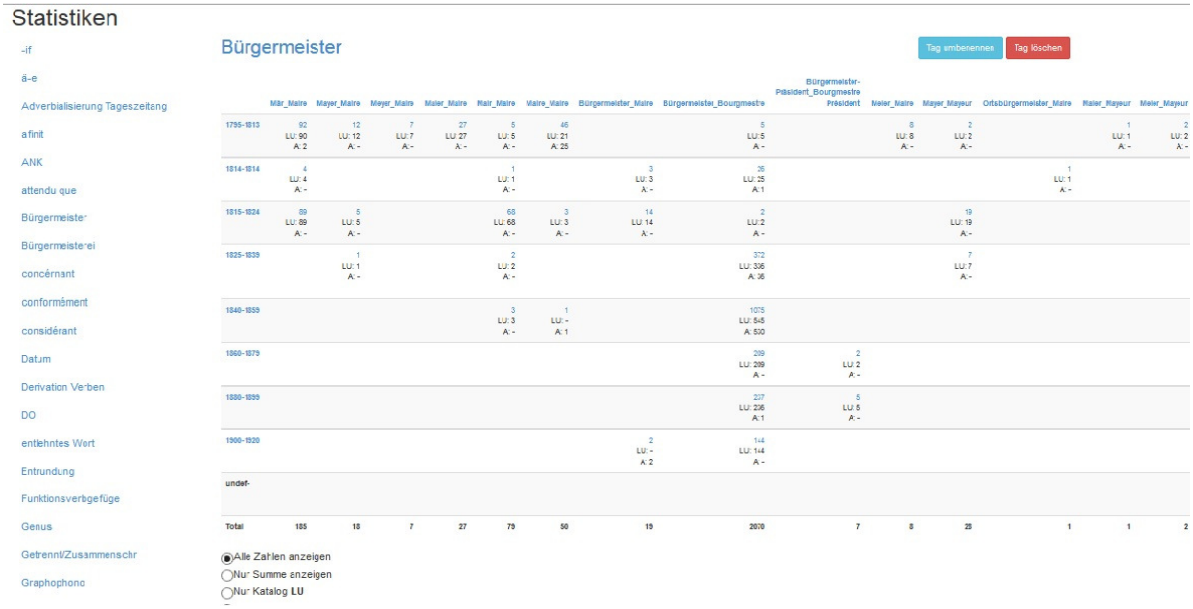


Abb. 6: Screenshot der Statistik für die Variable BÜRGERMEISTER

2.4 Statistiken

Im Menüpunkt „Statistik“ werden definierte Tagkategorien (ähnlich einem Inhaltsverzeichnis) aufgelistet, vergebene Werte gezählt und für jede Tagkategorie (d.h. Variable) in einer Tabelle präsentiert.[14]

Auf diese Weise sind die Tags systematisch gesammelt und müssen nicht von Hand, Dokument für Dokument herausgesucht und ausgezählt werden. Da diese Funktion in das Tagging-Tool integriert ist, ist somit kein Export der getaggten Texte erforderlich. Dieser Menüpunkt liefert somit automatisch die quantitative Auswertung der Variablenanalyse. Da nicht nur die Gesamtzahl aller vergebenen Tags angegeben wird, sondern ihre (absolute) Häufigkeiten nach Zeitintervallen aufgeschlüsselt werden, kann der Statistik die qualitative und quantitative Verteilung von Varianten im Verlauf des Untersuchungszeitraums direkt entnommen werden. Somit wird ein entscheidender Schritt diachroner Analysen maschinell unterstützt. Die Statistik in Abbildung 6 bspw. führt sämtliche Bezeichnungen des Amtes des Bürgermeisters, die in den Bekanntmachungen gefunden und als Werte dieser Variable via Taggen festgehalten wurden, in der Kopfzeile und darunter ihre Belegzahlen auf.

[14] Für die Tabellen wurde das Plugin Flexigrid verwendet.

Die einzelnen Zeitabschnitte können dabei individuell in den „Einstellungen“ bestimmt und geändert werden.

Standardmäßig werden alle Zahlen angezeigt, d.h. sowohl getrennt für die jeweiligen Korpora als auch die Summen für jede Zelle. Unterhalb der Tabelle gibt es jedoch die Möglichkeit, die Zahlen nur eines der Korpora (d.h. nur einen der Signaturanfänge) oder nur die Summen dargestellt zu bekommen. Die Tabelle kann des Weiteren via Copy&Paste-Verfahren in ein Programm zur Tabellenkalkulation übertragen werden.

Außer zur Übersicht über die Zahlen gelangt man über das Statistik-Menü zu den Belegstellen der Varianten. Die Werte (Varianten), Zeitabschnitte sowie Summen jeder einzelnen Zelle sind anklickbar und führen zu den jeweiligen Beleglisten. Diese entsprechen grundsätzlich dem Fenster mit den Resultaten des Suchmenüs. Durch das Anklicken der Signatur kommt man wiederum zum Belegdokument, das beim Öffnen direkt an die Stelle des Tokens springt. Hier ergibt sich ggf. die Möglichkeit zur Korrektur. Andererseits ist die Auflistung der Belegstellen eine wichtige Funktion für die (qualitative) Interpretation der quantitativen Ergebnisse. So lässt sich z.B. feststellen, ob bei dem einen oder anderen Phänomen lexemspezifische Realisierungen bzw. Entwicklungen vorzufinden sind.

Schließlich dient dieser Menüpunkt in gewisser Weise der Variablenverwaltung. So befindet sich an der linken Seite eine Art Inhaltsverzeichnis, das sämtliche definierte Tagkategorien auflistet. Des Weiteren kann man auf dieser Seite auch Tags umbenennen oder löschen.

3 Zusammenfassung

Es wurde ein Such- und Tagging-Tool vorgestellt, das sich besonders für die Durchführung von quantitativen Analysen, besonders für Paralleltexte und besonders für historische Untersuchungen eignet. Es dient weniger einer linguistischen Aufbereitung im Sinne einer Anreicherung von Texten mit linguistischer Information; vielmehr lassen sich relevante Kategorien bzw. Variablen selbst definieren, Tags frei im Text und auf verschiedenen Wegen setzen sowie ihre Werte frei formulieren. Deren Häufigkeiten können in den verlinkten Statistiken direkt abgerufen werden. Die Anwendung zeichnet sich somit durch Flexibilität und Teilautomatisierung aus. So können Autovervollständigungsfunktionen bei Datenfeldern, (teilweise) vorausgefüllte Suchformulare und die Angabe des zuletzt vergebenen Tags genutzt werden. Die Flexibilität bezieht sich auch auf benutzerspezifische Einstellungen der angezeigten Informationen und der zeitlichen Unterteilung des gesamten Untersuchungszeitraums. Verschiedensprachige Versionen desselben Inhalts können nebeneinander angezeigt werden. Nichtzuletzt können neben systemlinguistischen auch textbezogene (und u.U. auch diskursanalytische) Aspekte bearbeitet werden. Das Programm unterstützt Korpuslinguisten somit auf vielfältige Weise und erleichtert ihnen das empirische Arbeiten in vielerlei Hinsicht.

Literatur

Marcel Bollmann, Florian Petran, Stefanie Dipper und Julia Krasselt (2014): „CorA: A web-based annotation tool for historical and other non-standard language data”. In: *Proceedings of the EACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH), Gothenburg, Sweden*. Seiten 86-90.

William B. Cavnar und John M. Trenkle (1994): “N-Gram-Based Text Categorization”. In: *Proceedings of SDAIR-94 (3rd Annual Symposium on Document Analysis and Information Retrieval)*. Seiten 161-175.

Claudia Claridge (2008): “Historical Corpora”. In: Anke Lüdeling und Merja Kytö (Hrsg.): *Corpus Linguistics. Handbücher zur Sprach- und Kommunikationswissenschaft*. Berlin: Mouton de Gruyter. Seiten 242-259.

Martin Durrell, Astrid Ensslin und Paul Bennett (2008): “Zeitungen und Sprachausgleich im 17. und 18. Jahrhundert“. In: Werner Besch und Thomas Klein (Hrsg.): *Der Schreiber als Dolmetsch: Sprachliche Umsetzungstechniken beim binnensprachlichen Texttransfer in Mittelalter und Früher Neuzeit*. Berlin: Erich Schmidt. Seiten 263-279.

Stephan Elspaß (2005): *Sprachgeschichte von unten. Untersuchungen zum geschriebenen Alltagsdeutsch im 19. Jahrhundert*. Tübingen: Niemeyer.

Bernd Heine (2009): “Identifying instances of contact-induced grammatical replication”. In: Samuel G. Obeng (Hrsg.): *Topics in Descriptive and African Linguistics: Essays in Honor of Distinguished Professor Paul Newman.* Munich: LINCOM EUROPA. Seiten 29-56.

TEI Consortium (Hrsg.) (2015): *TEI P5*: *Guidelines for Electronic Text Encoding and Interchange*. [Version 2.8.0]. http://www.tei-c.org/P5/.

Rik Vosters, Gijsbert Rutten und Wim Vandenbussche (2012): “The sociolinguistics of spelling. A corpus-based case study of orthographical varia-

tion in nineteenth-century Dutch in Flanders". In: Ans M.C. van Kemenade und Nynke de Haas (Hrsg.): *Historical Linguistics 2009: Selected papers from the 19th International Conference on Historical Linguistics, Nijmegen, 10-14 August 2009*. Amsterdam/Philadelphia: John Benjamins. Seiten 253-274.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho und Chris Biemann (2013): "WebAnno: A flexible, web-based and visually supportedsystem for distributed annotations". In: *Proceedings of ACL 2013 System Demonstrations*. Seiten 1-6.

<JACK:lin> − Linguistische Module für das E-Assessment mit JACK

Tim Kocher **Ulrike Haß** **Bernhard Schröder**

Germanistik/Linguistik // Universität Duisburg-Essen

{tim.kocher, ulrike.hass, bernhard.schroeder}@uni-due.de

Abstract

In unserem Demo-Beitrag stellen wir Module für linguistische E-Klausuren und Übungen mit automatischem Feedback im Test- und Prüfungssystem JACK vor.

1 Rahmenbedingungen

In den linguistischen Fachanteilen der philologischen Fächer sind Vorlesungsgrößen von mehreren hundert Hörerinnen und Hörern keine Seltenheit. Modulprüfungen mit entsprechenden Teilnehmerzahlen bedeuten einen hohen Organisations- und Korrekturaufwand. Papierklausuren bedeuten für die Korrektur einen hohen zeitlichen Aufwand, der i. d. R. auf mehrere Personen verteilt werden muss. Kleinere schriftliche Übungen sind mit dem Wegfall von Übungen bzw. Tutorien unter diesen Bedingungen höchstens stichprobenartig durchführbar, ein individuelles Feedback ist nicht möglich.

Um den Korrekturaufwand zu reduzieren und die Zeit zwischen Test bzw. Klausur und Rückmeldung der Ergebnisse an die Studierenden zu reduzieren, bietet sich der Umstieg auf elektronische Lösungen – E-Assessment – an. Vorteile sind, je nach Aufgabentyp, automatisierte Korrekturen von E-Klausuren einerseits, sowie um Hinweise erweiterbare Übungsaufgaben mit unmittelbarem Feedback andererseits. Für Tests mit automatisiertem Feedback und E-Klausuren werden verschiedene Systeme angeboten, die sich hinsichtlich ihrer Möglichkeiten und Stärken deutlich unterscheiden. Nach der Prüfung von Alternativen arbeiten wir derzeit mit einer Kombination der Systeme Moodle und JACK. Während mit Moodle v. a. Lernmaterialien bereitgestellt werden, lassen sich mit JACK Aufgaben für beide o. g. Szenarien erzeugen.

2.1 Die Lösung mit <JACK:lin>

Moodle einerseits hat den Vorteil, dass es als Lernplattform einen Online-Kursraum bietet, in welchem Literatur (verschiedene Textdatei-Formate), ergänzt durch unterschiedliche multimediale Inhalte (Video-/Audio-Dateien, Hyperlinks) einem definierten Nutzerkreis, i. d. R. Studierende eines Seminars bzw. einer Vorlesung, zur Verfügung gestellt werden kann. Darüber hinaus sind neben Befragungen auch Tests mit unterschiedlichen Aufgabentypen durchführbar (darunter Lückentext, Multiple Choice, Zuordnung). In unserem speziellen Einsatzkontext haben sowohl rechtliche als auch technische Bedenken uns von einer Nutzung als E-Klausur-System Abstand nehmen lassen.

Mit *JACK* steht andererseits ein E-Assessment-System zur Verfügung, das für Prüfungen genutzt werden kann und bereits an verschiedenen Universitäten entsprechend genutzt wird (Striewe/Goedicke 2013). Der ursprünglich intendierte Anwendungsbereich von JACK waren die MINT-Fächer, insbesondere die Informatik und die Mathematik. Typische Aufgabenarten sind entsprechend Programmierübungen (erst in Java, mittlerweile auch in C++ (Striewe et al. 2008)) sowie formelbasierte Aufgaben zur Mathematik (ermöglicht durch die Einbindung von LaTeX).

Obwohl JACK also nicht ursprünglich für sprachwissenschaftlich orientierte AnwenderInnen konzipiert war, scheint es uns als Entwicklungsplattform für eine linguistische E-Klausur geeignet. Zwei Gründe sind dabei für unsere Bewertung ausschlaggebend:

1. JACK ist ein nach wie vor in der Entwicklung begriffenes System, das ständig verbessert und erweitert wird. In enger Zusammenarbeit mit den EntwicklerInnen kann der Funktionsumfang stetig angepasst werden.
2. Die Aufgabenerstellung ist ein modularisiertes System, bei dem über eine webgestützte graphische Benutzeroberfläche XML-Dateien zu (ggf.

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology, pages 170–171, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

aufeinander aufbauenden) Aufgaben verknüpft werden. Da das eigentliche Aufgabendesign innerhalb der XML-Dateien erfolgt, lassen sich – innerhalb der durch den Aufgabentyp gesetzten Grenzen – eine Vielzahl spezifisch sprachwissenschaftlicher Aufgaben erstellen. Dies ist von besonderem Interesse, da die zu erstellende Klausur eine große Bandbreite linguistischer Themengebiete abzudecken hat: Von Semiotik über Phonetik und Phonologie, Morphologie über Syntax zu Semantik und Pragmatik. So lässt sich z. B. die Darstellung relevanter IPA-Zeichen über Unicode-Referenzen in XML in den Aufgaben realisieren. Durch die Annotationsmöglichkeiten, die XML bietet, lässt sich ein Pool unterschiedlicher Satzgliedanalysen in Form XML-annotierter Sätze hinterlegen, aus welchem JACK dann randomisierte Aufgaben zur Satzgliedzerlegung und -klassifikation in Form eines speziellen Multiple-Choice-Aufgabentyps generieren kann.

Kurzfristig beschränken wir uns auf die Aufgabentypen Multiple Choice unter Einschluss des speziellen Subtyps Satzgliedanalyse und Lückentext, mittelfristig werden wir die Implementierung von Aufgaben zur Manipulation von Baumstrukturen anstreben. Für Aufgaben dieses Typs gibt es auch außerhalb der Linguistik, z. B. in der Informatik, Bedarf.

Ein weiterer Vorteil von JACK ist, dass sich mit JACK erstellte Aufgaben in die Moodle-Kurse einbetten lassen. Moodle ist bereits so etabliert, dass diese Einbettung die Akzeptanz von JACK seitens der Studierenden deutlich erhöhen wird. Wir könnten also unter Beibehaltung der gewohnten Lernplattform den Studierenden mit JACK ein Tool zur selbstständigen Lernfortschrittsevaluation an die Hand geben. In dieser Hinsicht ist JACK Moodle ebenfalls überlegen, da sich die Aufgaben in Schleifen anordnen lassen, die bei fehlerhaften Antworten (ggf. nach Informationsgehalt gestaffelt) Hinweise anzeigen, mit deren Hilfe die Aufgabe erneut bearbeitet werden kann.

2.2 Warum JACK?

Zwei alternative E-Assessment-Systeme wurden von uns getestet: Moodle und L-Plus. Wegen der zu geringen Detailtiefe der Protokollierung wird an unserer Universität aus rechtlichen Gründen derzeit von der Durchführung von Prüfungen mit Moodle abgeraten. Von L-Plus konnten wir uns im Rahmen eines mehrstündigen Lehrgangs einen Eindruck verschaffen. Die Eigenschaften, die unserer Ansicht nach für JACK sprechen, sind

1. die Einbindungsmöglichkeit von XHTM-Elementen: Tabellen, Formeln, Unicode;
2. flexiblere Auswertungsmöglichkeiten und
3. die XML-basierte Programmierbarkeit der Aufgaben.

Es ergibt sich in Summe ein Vorteil zugunsten von JACK, da die Aufgaben(-Typen) leichter variiert werden können und JACK die programmbasierte Generierung komplexer Spezifikationen ermöglicht. Letzteres ist für uns besonders für den fachspezifisch relevanten Aufgabentyp „Satzgliedanalyse“ interessant, da gerade dieser Aufgabentyp von den alternativen Programmen nicht oder nur mit hohem Aufwand überhaupt realisiert werden kann.

Literatur

Michael Striewe, Michael Goedicke and Moritz Balz. 2008. *Computer Aided Assessments and Programming Exercises with JACK.* Technical report 28, ICB, University of Duisburg-Essen.

Michael Striewe and Michael Goedicke. 2013. JACK revisited: Scaling up in multiple dimensions. *Proceedings of Eighth European Conference on Technology Enhanced Learning (EC-TEL),* Paphos, Cyprus, 635-636.

KoGraR: standardized statistical analyses of corpus counts

Sascha Wolfer
IDS Mannheim
R5, 6-13
D-68161 Mannheim
wolfer@ids-mannheim.de

Sandra Hansen-Morath
IDS Mannheim
R5, 6-13
D-68161 Mannheim
hansen@ids-mannheim.de

Hans-Christian Schmitz*
Fraunhofer FKIE
Fraunhoferstr. 20
D-53353 Wachtberg
hans-christian.schmitz@fkie.fraunhofer.de

Within the project "Corpus grammar" (Korpusgrammatik) at the Institute for the German Language (Institut für Deutsche Sprache, IDS) in Mannheim, techniques and tools are developed for the description of grammatical phenomena based on analyses of very large morpho-syntactically annotated corpora. The goal of the project is a corpus-based grammar that captures variations of grammatical structure in present-day German. In the first project phase, pilot studies were conducted (cf. Bubenhofer et al., 2014; Fuß, 2014; Konopka, 2014) to exploit and evaluate various methodological approaches to variation phenomena. For each research question, statistical analyses were chosen and customized. From these analyses, a subset was extracted as the methodological core of the project, with the aim of supporting methodological coherence, interoperability of sub-projects and, finally, the descriptive coherence of the project result, that is, the grammar. The methodological core has been made available to project members via an easy-to-use web front-end: the results of corpus queries and other, user-defined data tables can be uploaded and analyzed automatically. The web front-end is called KoGraR.

A tool like KoGraR has to meet several requirements: (1) The statistical analyses that are conducted have to be general enough to study a wide range of variation (lexical, morpho-syntactic and syntactic) phenomena. (2) The tool has to incorporate tests of statistical significance but also effect size. This is necessary because analyses based on very large corpora tend to show significant results while the size of the effects may be very small or even negligible. (3) The tool has to be easy to use, and documentations on the implemented statistical tests have to be accessible at a glance.

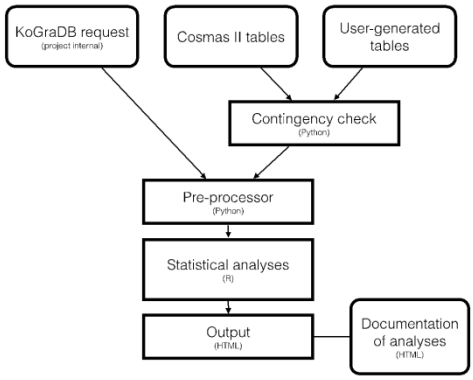


Figure 1: Schematic overview of KoGraR.

KoGraR is depicted schematically in Figure 1. As a basis for analyses, KoGraR takes frequency or contingency tables from various sources: firstly, KoGraR provides a direct interface to the KoGra database (KoGraDB) for processing the results of database queries (this resource is not open to the public). Secondly, arbitrary frequency tables can be entered manually ("User-generated tables" in Figure 1. Thirdly, frequency tables generated with Cosmas II can be uploaded.[1] It is possible to upload several tables at once and combine them in a multi-column table. The input is pre-processed, checked for contingency (which is most necessary for the user-generated tables), and transferred to a server-side installa-

* Hans-Christian Schmitz worked on KoGraR while he was a member of the IDS.

[1] Cosmas II is the Corpus Search, Management and Analysis System which makes huge portions of the IDS corpora available to the public (http://www.ids-mannheim.de/cosmas2/ [last access: May 7th, 2015]).

Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology,
pages 172–173, University of Duisburg-Essen, Germany, Sep 30–Oct 2 2015.

tion of R, an open environment for statistical computing and graphics (R Core Team, 2015). Currently, the following statistical procedures are applied on the tables: (1) output of tables and diagrams for raw data, normed and relative values, (2) a Chi-Square test as well as expected frequencies and standardized cell residuals, (3) Phi / Cramérs V association coefficient, (4) association and mosaic plots, (5) tables and diagrams for confidence intervals and (6) dispersion measures and plots with a focus on DP(norm) (Lijffijt & Gries, 2012). The set of procedures is open, thus, further analyses can be added easily on demand. For each test implemented in KoGraR, a short documentation with further information and help for the interpretation of the test is made available. The R code used to conduct the analyses on the server can be accessed directly and copied into a local installation of R (via copy & paste). The code to create the table objects in R is also included so that the user is able to retrace every step of the analyses and adapt the code where appropriate.

KoGraR has a standardizing influence on the collaborative work within the project "Corpus grammar". The implemented statistical analyses are used as the "standard catalogue" necessary for the conception of a monograph containing a corpus-based description of present-day German variation phenomena. All researchers currently working on the monograph are supposed to consult KoGraR with their empirical questions in order to assure methodological coherence of the grammar monograph.

The set of statistical analyses and the associated documentations can be useful for a variety of other linguistic projects that are working with large corpora. The only restriction (for KoGraR in its current state) is that the data has to be arranged in a frequency table in a meaningful way. Of course, this does not restrict the scope of KoGraR to linguistics. In principle, every researcher interested in the statistical analyses of contingency tables can use KoGraR.

Elementary knowledge of statistical methodology is expected from the potential users of KoGraR. To meet needs that exceed this basic statistical knowledge, the specific tests carried out are thoroughly documented and explained.

References

Noah Bubenhofer, Sandra Hansen-Morath, and Marek Konopka (2014). Korpusbasierte Exploration der Variation der nominalen Genitivmarkierung. *Zeitschrift für germanistische Linguistik 42* (3), S. 379-419.

Fred D. Davis (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly 13* (3), 319-340.

Eric Fuß (2014). Endungslose Genitive. In: grammis 2.0. – Korpusgrammatik. Electronical resource - Mannheim: Institut für Deutsche Sprache.

Stefan Th. Gries (2008): Dispersions and adjusted frequencies in corpora. *International Journal of Corpus Linguistics 13*(4). 403-437.

Marek Konopka (2014). Endungsvariation. In: grammis 2.0 – Korpusgrammatik. Variation der starken Genitivmarkierung. Electronical resource - Mannheim: Institut für Deutsche Sprache.

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org.`

Author Index