Noun phrase chunking and categorization for authoring aids

Mahlow, C ; Piotrowski, M

Noun Phrase Chunking and Categorization for Authoring Aids

Cerstin Mahlow and **Michael Piotrowski**
Institute of Computational Linguistics
University of Zurich
Switzerland
{mahlow,mxp}@cl.uzh.ch

Abstract

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

1 Introduction

In the LingURed project[1], we are implementing functions to support writers when revising and editing German texts. For example, when a writer chooses to use a different verb, the case of the noun phrase governed by the verb may also have to be changed; since the constituents of a German noun phrase agree in case, number, and gender, the writer must move through the noun phrase and make the necessary adjustments for each word form. It frequently happens that writers forget to make some or all of the required modifications since they are focusing on the change in the verb—which may also require other modifications in distant parts of a sentence, such as the addition or deletion of a separable prefix.

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

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

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

Another factor influencing the design and implementation of language-aware editing functions

[1] LingURed stands for "Linguistically Supported Revising and Editing," see http://lingured.info.

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

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

2 Noun phrase chunking for unrestricted German text

2.1 Motivation

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

A function for highlighting specific types of phrases is an example of an information function; an author may, for instance, call such a function to identify potential stylistic or grammatical trouble spots. A function for jumping to the next NP is an example of a movement function; it requires detecting the next NP after the current cursor position and moving the cursor to the start of the first word of this NP. A function for modifying the grammatical case of an NP is an example of an operation: The author places the cursor on a noun and calls the operation, indicating the desired case; the operation then calls a resource to extract the needed information and makes the necessary changes to the text.

Thus, we are interested in extracting chunks to serve as resource for higher-level functions, we are not interested in checking and correcting agreement or spelling of the elements of a chunk. For information and movement functions, we have to identify the words belonging to a certain type of phrase—the usual task for a chunker. For this paper, we take into account NPs as important structural elements of natural-language text and thus a target for information functions, movement functions, or operations. For operations like pluralizing an NP, changing the case of an NP, or replacing an NP by a pronoun, we have to extract the NP and to determine the category of the phrase, i.e., the morphosyntactic properties[2]. For German NPs these are *case*, *number*, *gender*, and *definiteness*[3].

2.2 Requirements

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

Availability For LingURed, we use the XEmacs editor[4], which is open-source. We aim to distribute all functions we implement as open-source, too. Therefore all involved resources should be freely available.[5]

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

Programming Interfaces and Further Processing The results of the chunking will be used in higher-level functions. Therefore they have to be delivered in a format suitable for further processing. The chunker will take input from and deliver results to a calling Emacs Lisp function, so it should offer programming interfaces to allow seamless integration.

[2] We refer to "morphosyntactic properties" as "category," while the process of determining this category is called "categorization."

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

[4] `http://xemacs.org`

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

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

2.3 Pragmatic definition of *noun phrases*

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

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

(1) $[_{\text{NP}}$ Der Traum$]$ $[_{\text{NP}}$ vom Essen$]$ ohne $[_{\text{N}}$ Reue$]$ beschert $[_{\text{NP}}$ der Nahrungsmittelindustrie$]$ $[_{\text{NP}}$ schöne Perspektiven$]$.

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

We concentrate on extracting contiguous base NPs for two reasons. First, there is a simple test to determine what to include in an NP when considering changing case or number of an NP: All word forms not affected by the change do not belong to the NP. In German, it is possible to embed complex phrases into an NP, as in *eine für die Verhältnisse hohe Qualität* ('a high quality with respect to the circumstances', literally: 'a for the circumstances high quality'):

(2) $[_{\text{NP}}$ eine $[_{\text{PP}}$ für $[_{\text{NP}}$ die Verhältnisse$]$ $]$ hohe Qualität$]$

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

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

(3) $[_{\text{ART}}$ Eine$]$ gemischte Crew
'a mixed crew'
$[_{\text{ART}}$ der$]$ transatlantischen Fusion
'of the transatlantic fusion'
$[_{\text{APPRART}}$ beim$]$ Sozialminister
'at the minister of social affairs'
$[_{\text{PPOSAT}}$ unserem$]$ zeitgeschichtlichen Bewusstsein
'our sense of contemporary history' (dative)
$[_{\text{PIDAT}}$ beide$]$ Polizisten
'both policemen'
$[_{\text{ART}}$ die$]$ $[_{\text{PIDAT}}$ beiden$]$ Polizisten
'these two policemen'
$[_{\text{PIAT}}$ einige$]$ Automobilhersteller
'some car manufacturers'
$[_{\text{PDAT}}$ diese$]$ heiklen Verfahren
'these critical processes'
$[_{\text{PPOSAT}}$ seines$]$ $[_{\text{ADV}}$ besonders$]$ religiösen $[_{\text{KON}}$ oder$]$ $[_{\text{ADV}}$ besonders$]$ homosexuellen Gehalts
'of its especially religious or especially homosexual content'

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

2.4 Related work

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

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

The exact definition of an NP also varies and clearly depends on the intended application; for example, the TreeTagger chunker uses a definition similar[7] to ours (Schmid and Schulte im Walde, 2000); YAC (Kermes and Evert, 2002), on the other hand, is intended for corpus preprocessing and querying and outputs recursive chunks.

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

3 The NPcat Chunker

For the LingURed project, we decided to implement an NP chunker to identify NPs and determine their categories according to the definition of NPs given above. The implementation is called *NPcat* and is based on part-of-speech tagging and morphological analysis.

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

As a morphological resource we use GERTWOL (Koskenniemi and Haapalainen, 1996). As Mahlow and Piotrowski (2009) show, it is currently the only morphological system for German available[8] that meets the requirements for integration into real-world applications and delivers high-quality results. GERTWOL is shipped as shared library with a C API for integration into applications.

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

3.1 Implementation details

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

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

As an example, let us consider the following sentence: *Nur wenn dieses strikte Verbot gelockert werde, heißt es in einer Studie der DG-Bank, könne über eine bessere Aufklärung der Verbraucher das brachliegende Potenzial konsequent erschlossen werden.* ('Only if this strict ban were lifted, a study of DG-Bank says, the untapped potential could systematically be exploited through better counseling of consumers'). Mbt delivers the tags presented in (4) below. Note that *gelockert*, *DG-Bank* and *Potenzial* are not in the lexicon, and the unknown words case base was used to predict the tags. We use the tags from the Stuttgart-Tübingen Tagset (STTS) (Schiller et al., 1999).

(4) $[_{ADV}$ Nur$]$ $[_{KOUS}$ wenn$]$ $[_{PDAT}$ dieses$]$
$[_{ADJA}$ strikte$]$ $[_{NN}$ Verbot$]$ $[_{VVPP}$ gelockert$]$
$[_{VAFIN}$ werde$]$ $[_{\$,}$,$]$ $[_{VVFIN}$ heißt$]$ $[_{PPER}$ es$]$
$[_{APPR}$ in$]$ $[_{ART}$ einer$]$ $[_{NN}$ Studie$]$ $[_{ART}$ der$]$

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

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

$[_{NN}$ DG-Bank$]$ $[_{\$,}$,$]$ $[_{VMFIN}$ könne$]$ $[_{APPR}$ über$]$
$[_{ART}$ eine$]$ $[_{ADJA}$ bessere$]$ $[_{NN}$ Aufklärung$]$
$[_{ART}$ der$]$ $[_{NN}$ Verbraucher$]$ $[_{ART}$ das$]$
$[_{ADJA}$ brachliegende$]$ $[_{NN}$ Potenzial$]$
$[_{ADJD}$ konsequent$]$ $[_{VVPP}$ erschlossen$]$
$[_{VAINF}$ werden$]$ $[_{\$.}$.$]$

The following NPs are then extracted from this sentence:

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

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

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

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

3.2 Experiments

To evaluate the appropriateness of our approach, we carried out some experiments. Some of these experiments were also intended to get an impression of

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

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

morphosyntactic features of German NPs, in order to decide whether functions involving extracting NPs and determining their category can be of any use at all. The quality of the results delivered by NPcat clearly depends on the quality of the tagging and the quality of the morphological analysis.

3.2.1 Quality of the tagging

We decided to use Mbt for tagging as it is open-source software and can be used interactively. When using Mbt, it has to be trained on an annotated corpus. The currently available annotated corpora for German with an appropriate size to be used as training set are NEGRA, TIGER, and TüBa-D/Z. Of these, TüBa-D/Z is being actively maintained and enhanced. However, all of these corpora contain almost exclusively texts written according to spelling rules *before* the 1996 spelling reform. There seem to be some articles in the TIGER written according to current spelling rules. However, this is not mentioned in the release notes. Both NEGRA and TüBa-D/Z do not include texts written according to current spelling rules. Thus, these corpora do not represent the *current* spelling and are, strictly speaking, not suitable

to be used as resource for any application dealing with current texts.

To our knowledge there is only one annotated resource available written in current German spelling: The two small German corpora in the SMULTRON treebank (Gustafson-Čapková et al., 2007). However, with around 520 sentences each[9], they are too small to serve as a resource for training Mbt. They also lack morphological information (there is information on gender only) and thus cannot be used as a gold standard for morphological analysis and NP categories.

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

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

3.2.2 Quality of noun chunks

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

As noted above, this primarily depends on the quality of the POS tagging—clearly, if a noun is mistagged as a verb, our rules cannot find the corresponding NP. The question is thus how well the tagger is able to identify the constituents of NPs; this question is not answered by general accuracy numbers, but would require comparison to a gold standard. While annotated corpora usually include annotations for NPs or noun chunks, the underlying definition of noun chunks does not necessarily correspond to our definition. We would thus have to create a gold standard ourselves—something we have not yet done at the time of this writing, thus we cannot provide evaluation results for this aspect.

3.2.3 Categories of noun chunks

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

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

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

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

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

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

[9] 7,416 tokens (529 sentences) taken from the novel "Sophie's World" and 10,987 tokens (518 sentences) taken from three business texts.

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

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

Table 1: Categories of NPs

ambiguous intermediate results of NLP resources may require interaction with the user, which could be counterproductive.

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

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

For *der Reparaturwerkstatt* ('to/of the garage') we obtain the two categories `FEM SG GEN` and `FEM SG DAT`. The plural forms of these categories are *der Reparaturwerkstätten* and *den Reparaturwerkstätten*—here, the user either has to identify the category of the original NP or has to choose between the two possible plural NPs. 41,802 of all NPs (11.5%) are categorized as `FEM SG GEN` and `FEM SG DAT`.

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

3.2.4 Quality of categorization

Finally, which quality can we expect for the categories of the identified NPs? The ambiguity of NPs clearly influences the interaction with the user when

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

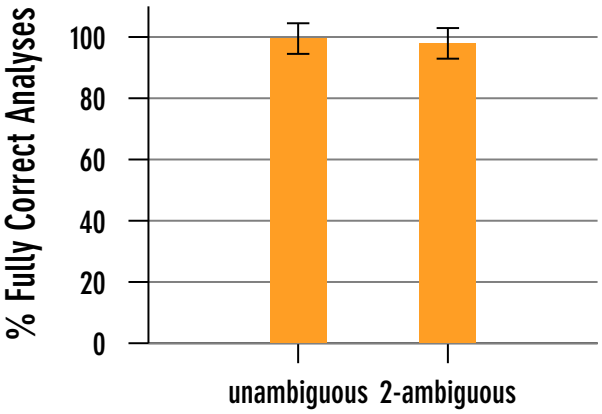

Figure 1: Percentage of completely correct analyses with a confidence interval of 5%

used in operations as shown above. However, we need some confidence about the correctness of the determined category of a certain NP, since users should know whether they can trust the changes made by operations based on NP chunking. If the correctness is insufficient, users would have to check—and possibly revise—all changes and there would be no benefit in using such an operation.

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

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

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

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

(6) a. * deren Freundin: `FEM SG GEN`
'whose girlfriend'

b. * deren Schwester: FEM SG GEN
'whose sister'

In both cases, *deren* ('whose') was incorrectly tagged as PDAT instead of as PRELAT. In fact, both NPs are ambiguous with respect to case. This type of problem may be reduced by improving the training of the tagger.

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

(7) a. * der Zivilflugzeuge: MASC SG NOM, NEU PL GEN
'(of) the airplanes'

b. * seinem Urteil: MASC SG DAT, NEU SG DAT
'his decision'

c. * vielen Straßenkämpfen: MASC PL DAT, NEU SG DAT
'many riots'

d. * möglichen Punkten: MASC PL DAT, NEU SG DAT
'possible points'

e. * kurzen Hauptsätzen: MASC PL DAT, NEU SG DAT
'short main clauses'

4 Conclusion

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

In this paper, we outlined requirements for an NP chunker and categorizer to be used as resource for language-aware editing functions to support authoring of German texts. Currently available chunkers do not meet these requirements and we therefore had to implement our own solution—NPcat—on the basis of existing resources for tagging and morphological analysis. We showed that NPcat meets the usual quality criteria for NP chunking of German texts.

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

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

References

Kenneth W. Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136–143, Morristown, NJ, USA. Association for Computational Linguistics.

Alan Cooper, Robert Reimann, and David Cronin. 2007. *About Face 3: The Essentials of Interaction Design*. Wiley, Indianapolis, IN, USA, 3rd edition.

Walter Daelemans, Jakub Zavrel, Antal van den Bosch, and Ko van der Sloot. 2010. MBT: Memory-Based Tagger version 3.2 reference guide. Technical report, Induction of Linguistic Knowledge Research Group, Department of Communication and Information Sciences, Tilburg University, June.

Stefan Evert. 2004. The statistical analysis of morphosyntactic distributions. In *LREC 2004 Fourth International Conference on Language Resources and Evaluation*, pages 1539–1542.

Michael Good. 1981. Etude and the folklore of user interface design. In *Proceedings of the ACM SIGPLAN SIGOA symposium on Text manipulation*, pages 34–43, New York, NY, USA. ACM.

Sofia Gustafson-Čapková, Yvonne Samuelsson, and Martin Volk. 2007. SMULTRON (version 1.0) – The Stockholm MULtilingual parallel TReebank.

Erhard W. Hinrichs. 2005. Finite-state parsing of German. In Antti Arppe, Lauri Carlson, Krister Lindén, Jussi Piitulainen, Mickael Suominen, Martti Vainio, Hanna Westerlund, and Anssi Yli-Jyrä, editors, *Inquiries into Words, Constraints, and Contexts. Festschrift for Kimmo Koskenniemi on his 60th Birthday*, CSLI Studies in Computational Linguistics ONLINE, pages 35–44. CSLI Publications, Stanford, CA, USA.

Hannah Kermes and Stefan Evert. 2002. YAC – a recursive chunker for unrestricted German text. In M. G. Rodriguez and C. P. Araujo, editors, *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 1805–1812.

Kimmo Koskenniemi and Mariikka Haapalainen. 1996. GERTWOL – Lingsoft Oy. In Roland Hausser, editor, *Linguistische Verifikation. Dokumentation zur Ersten Morpholympics 1994*, chapter 11, pages 121–140. Niemeyer, Tübingen.

Sandra Kübler, Kathrin Beck, Erhard Hinrichs, and Heike Telljohann. 2010. Chunking German: An unsolved problem. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 147–151, Uppsala, Sweden, July. Association for Computational Linguistics.

Cerstin Mahlow and Michael Piotrowski. 2008. Linguistic support for revising and editing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 9th International Conference, CICLing 2008, Haifa, Israel, February 17–23, 2008. Proceedings*, pages 631–642, Heidelberg. Springer.

Cerstin Mahlow and Michael Piotrowski. 2009. A target-driven evaluation of morphological components for German. In Simon Clematide, Manfred Klenner, and Martin Volk, editors, *Searching Answers – Festschrift in Honour of Michael Hess on the Occasion of his 60th Birthday*, pages 85–99. MV-Verlag, Münster, October.

Cerstin Mahlow, Michael Piotrowski, and Michael Hess. 2008. Language-aware text editing. In Robert Dale, Aurélien Max, and Michael Zock, editors, *LREC 2008 Workshop on NLP Resources, Algorithms and Tools for Authoring Aids*, pages 9–13, Marrakech, Morocco. ELRA.

Donald A. Norman. 1981. Categorization of action slips. *Psychological Review*, 88:1–15.

Michael Piotrowski and Cerstin Mahlow. 2009. Linguistic editing support. In *DocEng'09: Proceedings of the 2009 ACM Symposium on Document Engineering*, pages 214–217, New York, NY, USA, September. ACM.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Somerset, New Jersey. Association for Computational Linguistics.

Michael Schiehlen. 2002. Experiments in German noun chunking. In *Proceedings of the 19th international conference on Computational Linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Stuttgart.

Helmut Schmid and Sabine Schulte im Walde. 2000. Robust German noun chunking with a probabilistic context-free grammar. In *Proceedings of the 18th conference on Computational linguistics*, pages 726–732, Morristown, NJ, USA. Association for Computational Linguistics.

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

Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2009. Stylebook for the tübingen treebank of written German (TüBa-D/Z). Technical report, Universität Tübingen, Seminar für Sprachwissenschaft.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*, pages 127–132, Morristown, NJ, USA. Association for Computational Linguistics.

Michael Lee Van De Vanter. 1995. Practical language-based editing for software engineers. In *Software Engineering and Human-Computer Interaction*, Lecture Notes in Computer Science, pages 251–267. Springer.