A Self-Learning Context-Aware Lemmatizer for German

Praharshana Perera and **René Witte**
Institute for Program Structures and Data Organization (IPD)
Universität Karlsruhe, Germany
perera|witte@ipd.uka.de

Abstract

Accurate lemmatization of German nouns mandates the use of a lexicon. Comprehensive lexicons, however, are expensive to build and maintain. We present a self-learning lemmatizer capable of automatically creating a full-form lexicon by processing German documents.

1 Introduction

Lemmatization is the process of deriving the base form, or *lemma*, of a word from one of its inflected forms. For morphologically complex languages like German this is not a simple task that can be solved solely through a rule-based algorithm: Performing an accurate lemmatization for German requires a lexicon. This can be either a lexicon containing all inflected forms of a word together with its base form *(full-form lexicon)*, or just the lemma together with a set of rules for creating its inflected forms *(base-form lexicon)* (Hausser, 2000).

Creating such a lexicon by hand, however, is expensive and time-consuming. Perhaps because of this there are currently no freely available lexical resources for German that include full case and inflection information.[1] Moreover, even a full-scale commercial lexicon can fail when encountering specialized terminology.

As a consequence, most systems processing German texts currently perform the much simpler task of *stemming*, which often generates stem forms of words that might not actually exist in the language (so-called *overstemming*). Stemming is frequently used for information retrieval (IR) tasks, an example being the German stemmer contained in the full-text search engine *Lucene*,[2] which is based on the algorithm described in (Caumanns, 1999). While overstemming is a feasible approach for text retrieval, a *text mining* system often needs to obtain a more precise lemma, for example, in order to perform a gazetteer lookup to identify named entities or for description logic (DL) queries within an ontology.

The goal of our work, therefore, is to allow the semi-automatic generation of a lexicon by mining full-text documents. Since there are currently no free lemmatization systems for German available,[3] all components have been developed for release as free, open-source software.

2 Lemmatization Algorithm

Our lemmatization system has two main components, an algorithm and a lexicon. The algorithm lemmatizes German nouns depending on morphological classes. The lexicon, which is described in Section 3, is generated from the nouns that have been processed by this algorithm, with some additional capabilities for self-correction.

The lemmatization algorithm considers the context and grammatical features of the language to lemmatize German words. It requires an additional POS tagger and an NP chunker, which are used as resources to extract the features of words and their surrounding context. It has been developed primarily for nouns but can also be extended to lemmatize adjectives and verbs.

2.1 Inflection of German Nouns

In German there are seven declensional suffixes for nouns: *-s*, *-es*, *-e*, *-n*, *-er*, and *-ern* (Caumanns, 1999). These suffixes are due to the morphological

[1] The free online dictionary *Wiktionary* (`http://de.wiktionary.org/`) had at the time of writing (May 2005) less than 5000 entries for German.

[2] `http://lucene.apache.org/`

[3] The *Morphy* system (Lezius et al., 1998) is described as "freely available," but in fact is closed-source, binary-only, non-changeable software. It is also no longer being maintained.

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 636–643, Vancouver, October 2005.

Class	Features	Remove Suffix
I	$\{Sg\} \wedge \sim \{Gen\}$ $\wedge \{Masc \vee Fem \vee Neut\}$	none
II	$\{Sg\} \wedge \{Gen\}$ $\wedge \{Masc \vee Neut\}$	*-es* or *-s*
III	$\{Pl\} \wedge \sim \{Dat\}$ $\wedge \{Masc \vee Fem \vee Neut\}$	*-e*, *-n*, *-en*, *-er*, or *-s*
IV	$\{Pl\} \wedge \{Dat\}$ $\wedge \{Masc \vee Fem \vee Neut\}$	*-n*, *-en*, *-ern*, or *-s*

Table 1: Lemmatization of German nouns based on morphological classes

features such as gender, number, and case (Vilares et al., 2004). A basic lemmatization algorithm would reduce the suffixes by analyzing these morphological features. The existence of these suffixes is caused by the following: (1) genitive form of the singular, masculine, or neuter nouns have the declensional suffixes *-es*, *-en*, or *-s*, e.g., *Kind* → *Kindes*; (2) plural nouns have the declensional suffixes *-en*, *-ern*, *-n*, or *-s*, e.g., *Frau* → *Frauen*; and (3) dative forms of plural nouns have the declensional suffixes *-s*, *-n*, *-en*, or *-ern*, like in *Kind* → *Kindern*.

A simple lemmatization algorithm has been developed to cutoff these suffixes taking the morphological features such as number, gender, and case into consideration. The values of these features often cannot be uniquely determined from the word form (Evert, 2004). Therefore, we developed an algorithm to classify the nouns into four different morphological classes, as shown in Table 1. Lemmatization can then be performed based on these morphological classes (Table 1, right column).

We now discuss the first step, finding the proper class for each noun.

2.2 Lemmatization Classes

The currently available POS taggers for German do not capture more complex morphological features like number or case. Thus, in order to lemmatize German nouns it is necessary to first categorize them into the classes defined above. Our algorithm achieves this by analyzing the grammatical features of a noun, based on the German grammar (Duden, 1995). Additionally, a stochastic case tagger has been developed as an additional resource to support the algorithm in the classification of nouns.

2.2.1 Nouns with a Determiner

Table 2 shows statistics for German noun phrases for different corpora (the size of each corpus can be found in Section 5). The percentage of nouns that have a determiner is around 34% (25% determiner only + 9% determiner and modifier). The morphological information that can be extracted from a determiner preceding a noun is very ambiguous. For example, the determiner *die* can be either singular or plural in number, nominative or accusative in case, and masculine, feminine, or neuter in gender. But some determiners can be used to classify nouns into morphological classes.

Corpus	Det Only	Mod Only	Det+Mod	None
Negra	25%	13%	9%	53%
Die Welt	26%	14%	9%	51%
AvFIS	22%	16%	8%	53%
Wikipedia	28%	15%	9%	48%

Table 2: Distribution of German noun phrases

Table 3 describes our algorithm for nouns that have a determiner. In the first step, we consider determiners that are singular and non-genitive. Therefore, they belong to class I and do not need to be lemmatized. Examples are *das Haus* → *Haus*, *dem Mann* → *Mann*, *eine Frau* → *Frau*.

Determiners in the second step are singular and genitive and the gender can be masculine or neuter. These nouns belong to class II and to find the lemma, the suffix *-s* or *-es* must be removed. Examples are *des Hauses* → *Haus*, *des Vaters* → *Vater*.

Determiners in the third step can be either singular or plural. The only possible way to differentiate this is when the noun has both a determiner and a modifier. The plurals have modifiers ending with *-en* and singulars with *-e*.

In the other steps, nouns cannot be directly classified. In the fourth step we apply additional heuristics and in the last step the statistical case tagger (described in Section 2.4) is being used.

In German, genitive is mostly used as the case of nominal modifiers and complement of prepositions (Hinrichs and Trushkina, 1996), which is used as a heuristic to find the singular determiners in the fourth step and in the same way another heuristic has been applied which finds singular determiners when they are followed by dative prepositions.

The determiner *den* in German can be either accusative or dative. In the dative case it is plural and in the accusative case it is singular and masculine in gender. Examples are *den Kindern* (dative plural)

Step	Determiner	Class
1	das, dem, ein, einem, ..., ihr, ihrem	Class I
2	des, eines, meines, deines, ..., ihres	Class II
3	die, meine, deine, ..., ihre	If modifier has the suffix *-e* → Class I If modifier has the suffix *-en* → Class III
4	der, meiner, deiner, ..., ihrer	If determiner is not followed by a genitive preposition or a noun phrase → Class I If determiner is followed by a dative preposition → Class I
5	den, meinen, deinen, ..., ihren	If case tagged by case tagger is accusative → Class I If case is dative → Class IV

Table 3: Lemmatizing German nouns that appear with a determiner

and *den Salat* (accusative singular). The fifth step has determiners that have this ambiguity, which is resolved using information given by the case tagger.

2.2.2 Nouns with a Modifier only

The morphological features of a noun that can be extracted from a modifier are less than those based on a determiner. According to the statistics in Table 2, around 14% of noun phrases come with a modifier only. However, it is sometimes possible to lemmatize nouns by looking at the modifiers' suffixes and the case information as given by the case tagger. Table 4 describes our algorithm for nouns that come solely with a modifier.

In German, when a noun exists without a determiner but with a modifier, the ending of the modifier changes according to the morphological features of the noun. For example, the noun phrase *dem kleinen Kind* without determiner becomes *kleinem Kind*. The suffix *-em* appears only for singular nouns, which do not need to be lemmatized.

A modifier with the suffix *-es* can be genitive, accusative, or nominative. A good example for this feature is *kleines Kind* and *kleines Kindes*. In the first case it is nominative or accusative and in the second case genitive. Here, we use the case information given by the case tagger to classify the noun.

Modifiers with the suffix *-en* are similar to the step with the determiner *den*. A modifier with suffix *-en* can be either singular or plural. In singular case it is accusative and in plural case dative; examples

Step	Modifier Suffix	Action
1	*-em*	Class I
2	*-es*	If case is not genitive → Class I If case is genitive → Class II
3	*-en*	If case is accusative → Class I If case is dative → Class IV
4	*-er*	If case is dative or nominative → Class I

Table 4: Lemmatizing German nouns with a modifier but without a determiner

for these cases are *guten Mann* (accusative, singular) and *guten Männern* (dative, plural).

Modifiers that have the suffix *-er* can be both genitive or non-genitive. In the non-genitive case they are singular and need not to be lemmatized. Examples for this are *kleiner Katze* (dative, singular), *kleiner Katze* (genitive, singular), and *kleiner Katzen* (genitive, plural).

2.2.3 Nouns without Modifier or Determiner

Nouns without modifier or determiner account for 51% of all NPs (Table 2). Most of these nouns cannot be directly lemmatized using methods as they have been applied above. The main reason for this is the unavailability of a tagger providing number and gender information for such nouns. Using only the case tagger it is not possible to classify all the nouns in this set. However, it is possible to capture some nouns in this set by applying a heuristic:

> If a noun follows the preposition *zum, zur, am, im, ins,* or *ans* ⟶ Class I.

The main idea behind this heuristic is a grammatical feature of the German language. In German, there exists a set of prepositions that are connected with a determiner, for example, *zum Bahnhof*, *zur Party*, and *ins Bett*. The main feature of nouns following such a preposition is that they are singular and thus do not need to be lemmatized.

2.2.4 POS-based Lemmatization

To maximize the number of nouns that can be lemmatized a heuristic has been added to capture nominative nouns, using the POS tagger *TreeTagger* (Schmid, 1995). The main idea behind this heuristic is to find the subject and main verb of a sentence. In German, the subject is always nominative and by looking at the suffix of the main verb, it is possible to determine the number of the subject.

This heuristic first finds the subject of the sentence based on the case tagger information. Then, based on the information from the POS tags the main verb is identified and checked whether it is a plural verb. The corresponding plural nouns are then lemmatized, whereas singular nouns remain unchanged.

2.3 Optimizations

To avoid some errors in the lemmatization algorithm and to increase the accuracy of lemmatization additional optimizations are needed. In German, many plural forms are built by changing a vowel to an Umlaut (Caumanns, 1999), like in *das Land* and *die Länder*. But this is not a static rule because there are some cases where the noun already has an Umlaut, like in *die Affäre* and *die Affären*. Here, it would not be correct to lemmatize *Affären* to **Affare*. As a solution, several possible lemma candidates are generated, for example, *Länder* → **Länd* and *Land*.

Another feature of German are nouns that are made up from adjectives. These nouns have different suffixes when they appear with definite or indefinite determiners and without determiners. An example is the noun *Abgeordnete*; in singular form it can appear in two ways, *der Abgeordnete* and *ein Abgeordneter*. It is also tricky in the dative singular case, where it has three forms, *Abgeordnetem*, *Abgeordneter* and *dem/der/einem/einer Abgeordneten*. Our algorithm thus generates the possible lemma candidates: *Abgeordneter* → *Abgeordneter*, *Abgeordnete*.

The main reason to generate lemma candidates for these nouns above is to store them in the lexicon. The correct lemma can then later be identified and the lexicon updated when the noun appears again in a different context.

2.4 The Case Tagger

As an additional resource to the lemmatizer we developed a stochastic case tagger. It has been built using the POS tags as features to train the model in order to predict the case of nouns. From the standard STTS tagset for German (Schiller et al., 1995), which has 54 POS tags, 38 tags[4] have been identified to train the model, based on an analysis of the grammatical structure of German sentences as defined in the German grammar (Duden, 1995).

2.4.1 Model

We apply a standard Hidden Markov Model (HMM), designed for the structure of the German language. A German sentence can be represented as a set of variable states, which can be nominative, accusative, dative, or genitive and a set of fixed states like finite verbs and conjunctions. For example, in the sentence *Die Mutter gibt den kleinen Kindern den Salat*, the phrases *Die Mutter* (nominative), *den kleinen Kindern* (dative) and *den Salat* (accusative) are the variable states and the finite verb *gibt* is a fixed state. In this manner, the whole sentence can be represented with the state sequence *nominative VVFIN* (finite verb) *dative accusative*. From the 38 tags that have been chosen for training, 10 tags[5] have been integrated with the nouns as variable states.

2.4.2 Tagging Algorithm

As an HMM tagger, our case tagger chooses the best sequence of tags for a given sequence of states (Jurafsky and Martin, 2000). In this model this can be expressed as choosing the best sequence of tags for the variable states in the sequence. The first stage of the algorithm selects the set of tags from the POS tags that are used for calculation and then it orders these tags into fixed and non-fixed states with respect to the grammatical case. The second stage of the algorithm calculates the most probable tag sequence using the Viterbi algorithm. The model is smoothed to avoid zero probabilities. In the worst case the complexity of this algorithm is $O(N^3)$ but here $N = 4$, the four grammatical cases.

3 Lexicon Generation

As discussed above, the lemmatization algorithm cannot be used alone to lemmatize all German nouns, as it cannot capture every noun in a text. However, a noun that could not be lemmatized within one text may well have enough context information for a precise lemmatization within another. Thus, our main idea here is to create a self-learning lexicon that evolves with the nouns processed by the algorithm, continuously learning the correct values for each lexical entry.

[4] These POS tags define the structure of the grammatical case in German sentences, for example, verbs and prepositions.

[5] Like for nouns, grammatical case is a morphological feature of these POS tags, for example, pronouns and adjectives.

3.1 Lexicon Entries

The lexicon stores the full form of a word with its base form and possible morphological features like number, gender, and case. This is different from a lexicon as it has been used for lemmatization, which only stores the base form for each word together with its inflection class (Lezius et al., 1998).

For example, the lexicon entries for the noun *Kind* are represented as:

Noun	Number	Gender	Case	Lemma
Kind	Sg	Neut	Nom.Akk	Kind
Kindes	Sg	Neut	Gen	Kind
Kinder	Pl	Neut	Nom.Akk	Kind
Kindern	Pl	Neut	Dat	Kind

3.2 Lexicon Generation

The lexicon grows by updating itself from the nouns that have been processed by the lemmatization algorithm. Additional functionality has been implemented in the lexicon, to allow it to evolve by assigning the correct lemma to the words that are inflected from the same lemma and correcting some errors that have been generated by the algorithm.

3.2.1 Evolving the Lexicon

If a word is scheduled for addition to the lexicon, it first checks whether it already exists. If this is the case, it compares each feature of the new word with the one already in the lexicon. If there is any difference, for example, if the word in the lexicon shows the number *Sg* and the new word has the number *Pl*, it adds both features to the lexicon entry. If a new word does not already exist in the lexicon it will just be added as a new entry. The following example illustrates this process:

Current state of the lexicon				
Menschen	Sg	Masc	Akk	Mensch
Mensch	Sg	Masc	Nom	Mensch
New Entry				
Menschen	Pl	Masc	Nom	Mensche.Mensch
State of the lexicon after update				
Menschen	Sg.Pl	Masc	Akk.Nom	Mensch
Mensch	Sg	Masc	Nom	Mensch

The assignment of the correct lemma *Mensch* is done by a procedure that will be discussed next.

3.2.2 Updating Lemmas

If a new word lemmatized by the algorithm that has more than one lemma candidate is to be added, the lexicon tries to assign the correct lemma for this new word by looking at the lemmas that are already in the lexicon. If one of the lemma candidates in the new word matches with a lemma stored in the lexicon, the lemma of the new word will be updated with the new information. This process is illustrated in the following example:

Current state of the lexicon (lemma only)	
Land	Land
Landes	Land
New Entry	
Länder	Lände.Länd.Lande.Land
State of the lexicon after update	
Land	Land
Landes	Land
Länder	Land

In the same way, if a new word that has been correctly lemmatized is to be entered to the lexicon, the lexicon tries to update the words in the lexicon that have more than one lemma using the lemma of the new word. If one of the lemma candidates of a word in the lexicon matches with the lemma of the new word, then the lemma of the word in the lexicon will be updated with the lemma of the new word:

Current state of the lexicon (lemma only)	
Länder	Lände.Länd.Lande.Land
Ländern	Länder.Lände.Länd.Lander.Lande.Land
New Entry	
Landes	Land
State of the lexicon after update	
Landes	Land
Länder	Land
Ländern	Land

3.2.3 Automatic Error Correction

The lemmatization algorithm may produce errors, for example, a plural noun wrongly tagged as singular may not be lemmatized, resulting in a wrong entry. While the lexicon evolves, such errors produced by the algorithm are corrected automatically.

As shown in the example below, the lexicon can have wrong entries and entering a word with more than one lemma, which is an inflectional form of a word that has a wrong entry, will not be assigned with the correct lemma because the procedure that updates the lemma will assign possible lemma candidates to this word. If a word that has a wrong entry in the lexicon will be entered again with the correct lemma, the word itself and all its inflectional forms will be updated with the correct lemma:

Current state of the lexicon (lemma only)	
Jahr	Jahr
Jahre	Jahre **(wrong)**
New Entry	
Jahren	Jahre.Jahr
State of the lexicon after update	
Jahr	Jahr
Jahre	Jahre **(wrong)**
Jahren	Jahre.Jahr **(two possibilities)**
New Entry	
Jahre	Jahr **(correct lemmatization)**
State of the lexicon after update	
Jahr	Jahr
Jahre	Jahr
Jahren	Jahr

4 Implementation

The lemmatization algorithm and the lexicon have been implemented based on the GATE architecture (Cunningham et al., 2002). GATE provides an infrastructure for developing and deploying software components that process human language. For the German POS tagger we currently use the TreeTagger (Schmid, 1995). The other main resource is a multi-lingual base NP chunker implemented within the JAPE language.

The Negra corpus version 2 (Skut et al., 1998) based on approximately 70 000 tokens tagged with morphological features has been used to train the case tagger. This corpus has been split into 50 000 training tokens and 20 000 tokens used for testing.

5 Evaluation

Evaluation was performed over four collections of texts: (1) a set of 350 articles from "Die Welt" newspaper containing 190 868 tokens (40 104 nouns); (2) the electronic version of the book "AvFIS"[6] containing 120 212 tokens (22 039 nouns); (3) six manually for lemma, case, and number annotated articles from the German *Wikipedia* containing 6580 tokens (1536 nouns); (4) 20 000 tokens (5023 nouns) from the Negra corpus version 2 (Skut et al., 1998), which contains morphological tags for case and number.

The lemmatization of German texts has been evaluated using both the algorithm and the lexicon separately and combined. Since the first two collections of texts are not annotated with lemmatization information, we evaluated the lemma produced by our algorithm or lexicon by comparing it with the one produced by the TreeTagger, which is based on an internal dictionary. Since the TreeTagger cannot produce the lemma for all nouns, we evaluated only that percentage of nouns for which the TreeTagger was able to produced a lemma, which is 88% for both "Die Welt" and the "AvFIS" book. In order to also evaluate our lemmatization independently from the lemma produced by the TreeTagger, we compared its results to a manually annotated set of articles from the Wikipedia.

Finally, the case and number taggers have also been evaluated separately using the manually annotated articles from the Wikipedia and the Negra corpus. For this evaluation, the lemmatization accuracy has been calculated by $accuracy = \frac{n(correct)}{n(lemmatized)}$.

Corpus	Nouns	Algorithm Only		Lexicon Only	
		Lemm.	Acc.	Lemm.	Acc.
Die Welt	35531	49%	0.88	67%	0.96
AvFIS	19394	40%	0.88	70%	0.97
Wikipedia	1536	49%	0.87	54%	0.97

Table 5: Lemmatization results, algorithm and lexicon tested in isolation

5.1 Algorithm Evaluation

Table 5 shows the results of lemmatization using only the lemmatization algorithm (i.e., no lexicon).

The number of nouns that our algorithm can lemmatize is just below 50%. This is mainly due to the large number of nouns, as shown in Table 2, that appear without a determiner or modifier, as well as some ambiguous cases where NPs with determiners and modifiers cannot be lemmatized directly.[7]

The accuracy of lemmatization based on this approach shows the irregular morphological features of the German language. 75% of the errors are due to irregular morphological variations in German. The algorithm does not change the vowels with Umlauts, therefore, all nouns which have a vowel with an Umlaut in plural are not lemmatized correctly. For example, the noun *Ländern* is lemmatized by the algorithm to **Länd* but the correct lemma is *Land*. Another peculiarity that causes errors in lemmatization are nouns that have been formed by adjectives. For example, a noun with a determiner like *ein Ab-*

[6] René Witte, *Architektur von Fuzzy-Informationssystemen*, BoD, 2002, `http://rene-witte.net`

[7] E.g., in the sentence *Ich sehe die Kinder der Frau* the two nouns *Kinder* and *Frau* cannot be lemmatized by the algorithm because in this context these nouns could be singular or plural.

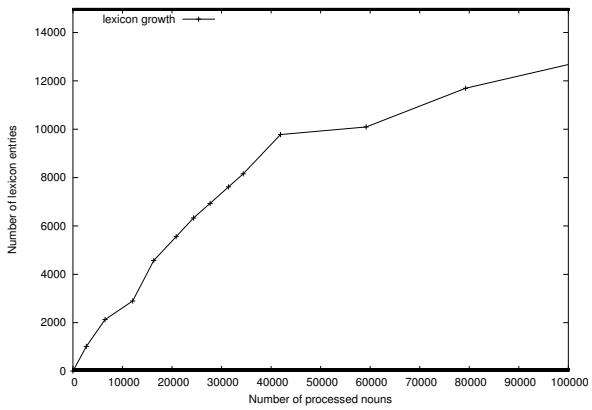

Figure 1: Lexicon growth

geordneter would not be lemmatized by the algorithm because it is singular and non-genitive. However, the correct lemma of this word is *Abgeordnete*. German also has nouns where the plural and the singular forms are equal. This is a situation in which the algorithm fails to generate the correct lemma. For example, the noun *Arbeiter* has the same singular *der Arbeiter* and plural *die Arbeiter* form. The algorithm lemmatizes *die Arbeiter* to **Arbeit* whereas the correct lemma is *Arbeiter*.

The remaining errors are due to mis-tagging, mainly by the case tagger, which can result in an error in lemmatization. For example, *den Kindern* has been tagged by the case tagger as **Akk* (correct *Dat*), so the lemmatization algorithm does not lemmatize this noun to *Kind* because the case is accusative and hence assumed to be singular.

5.2 Lexicon Evaluation

The lexicon was initially generated by applying the lemmatization algorithm on the "Die Welt" collection of texts. We then evaluated lemmatization based solely on the lexicon (not applying the algorithm) for these documents. Table 5 also shows the results for this collection of texts. The growth of the lexicon is shown in Figure 1; when we performed the evaluation it contained 12 858 entries for 10 251 lemmas.

The next test for lexicon evaluation has been done in two stages. First, the electronic book "AvFIS" (2) has been lemmatized using only the lexicon. Afterwards, we applied the lemmatization algorithm on the same book, generating new entries, and then evaluated the extended lexicon again on this book. Before processing the book, the lexicon was able

Corpus	Contribution			Results	
	Lex.	Alg.	Both	Lemm.	Acc.
Die Welt	27%	10%	39%	76%	0.94
AvFIS	33%	3%	37%	73%	0.96
Wikipedia	24%	19%	30%	73%	0.93

Table 6: Results using both algorithm and lexicon

to lemmatize 40% of all nouns with an accuracy of 0.98, whereas afterwards the lemmatization coverage increased to 70% with the accuracy dropping slightly to 0.97.

Both tests above have been done against the lemma generated by the TreeTagger. Additionally, we evaluated the lexicon on our manually annotated set of articles from the Wikipedia, which is also shown in Table 5.

As can be seen, in all tests the accuracy of lemmatization based on the lexicon is higher than that of the algorithm. The reason for this is the self-correcting feature of the lexicon discussed above: While the lexicon evolves it increasingly assigns the correct lemma for each noun.

Although the lexicon performs with a high accuracy, the remaining errors are due to various forms of the construction of words in German. For example, consider the two nouns *Sieger* (lemma *Sieger*) and *Sieg* (lemma *Sieg*). As the lexicon evolves, it assigns *Sieger* the lemma **Sieg* because it already exists as a lemma in the lexicon whereas the correct lemma is *Sieger*. Some remaining incorrect entries in the lexicon also result in errors. Such cases will need to be corrected manually.

The percentage of lemmatization is obviously high for texts which have been used to generate the lexicon. The difference can be clearly seen in the book example, where the number of nouns that could be lemmatized increased significantly after enhancing the lexicon from the same set of nouns.

5.3 Lexicon and Algorithm Evaluation

We evaluated lemmatization using both algorithm and lexicon combined on the same set of texts (Table 6, right side). The number of lemmatized nouns has clearly increased in the combined method. Here, a lemma produced by the lexicon takes precedence over the algorithms' one, if both were able to produce a lemma. Table 6 also shows the contribution of each method for lemmatization in the combined method (left side). The number of nouns lemmatized

by the lexicon is relatively higher than the algorithm on the first two texts because these texts were used to initially generate the lexicon.

When both algorithm and lexicon were able to produce a lemma, it agrees in 92% of all cases with an accuracy of 0.98.

One special case both fail to lemmatize correctly are foreign (e.g., Latin) words that do not follow German morphological rules (e.g., *Lexika* → *Lexikon*). These require manual correction or the development of specialized heuristics.

Finally, we evaluated the performance of the case and number taggers. While a detailed discussion of these results cannot be presented in this paper, the case tagger reaches an accuracy of 0.92 on the training data, 0.8 on the testing data, and 0.79 on the Wikipedia, while the number tagger has an accuracy of 0.93 on the training data, 0.9 on the testing data, and 0.91 on the Wikipedia corpus.

6 Conclusions and Future Work

In this paper we demonstrated a new algorithm for the lemmatization of German nouns. An important feature is the automatic construction of a lexicon from the processed documents, allowing it to continuously improve in both coverage and accuracy. The lemmatization system as well as a lexicon will be made available as free, open-source software, which will fill an important gap for the development of NLP systems dealing with German.[8]

The automatic generation and self-correction of a lexicon is a huge time-saver. Compared to the German Wiktionary, whose users needed a year to manually curate less than 5000 entries, we were able to compile the same amount of nouns within a matter of days.[9] Human intervention can be limited to the inspection and correction of wrong entries, which will allow the creation of specialized lexicons even for groups with limited resources. To increase the coverage of our lexicon, we currently employ a web crawler, which daily scans several German news sources for texts, which are then processed for lexical entries.

In the future, we plan to enhance the system to also deal with verbs, adjectives, and adverbs, as well as compound nouns.

Acknowledgments. This work is funded through the DFG project "Entstehungswissen."

[8] Dictionaries that are only accessible online, like Canoo.net (`http://www.canoo.net`) or Wortschatz Lexikon (`http://wortschatz.uni-leipzig.de`) we do not consider freely available, as the underlying databases and tools cannot be downloaded, modified, or integrated into NLP systems.

[9] The Wiktionary does have more information for each entry, however, some of these could also be automatically created in a similar fashion.

References

Jörg Caumanns. 1999. A Fast and Simple Stemming Algorithm for German Words. Technical report, Center für Digitale Systeme, Freie Universität Berlin.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proc. of the 40th Anniversary Meeting of the ACL*. `http://gate.ac.uk`.

Duden. 1995. *Grammatik der deutschen Gegenwartssprache*. Dudenverlag, Mannheim, 5^{th} edition.

Stefan Evert. 2004. The Statistical Analysis of Morphosyntactic Distributions. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal*.

Roland Hausser. 2000. *Grundlagen der Computerlinguistik*. Springer Verlag.

E. Hinrichs and J. Trushkina. 1996. Forging agreement: Morphological disambiguation of noun phrases. In *Proceedings of the First Workshop on Treebanks and Linguistic Theory*.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing*. Prentice Hall PTR.

Wolfgang Lezius, Reinhard Rapp, and Manfred Wettler. 1998. A Freely Available Morphological Analyzer, Disambiguator and Context Sensitive Lemmatizer for German. In *Proc. COLING-ACL*, pages 743–748.

A. Schiller, S. Teufel, and C. Thielen. 1995. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart and Tübingen.

H. Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A Linguistically Interpreted Corpus of German Newspaper Text. In *Proceedings of the ESS-LLI Workshop on Recent Advances in Corpus Annotation. Saarbrücken, Germany*.

Jesús Vilares, Miguel A. Alonso, and Manuel Vilares. 2004. Morphological and Syntactic Processing for Text Retrieval. In *DEXA 2004*, Springer LNCS 3180.