Learn - Filter - Apply - Forget.
Mixed Approaches to Named Entity Recognition.

Martin Volk, Simon Clematide

University of Zurich
Department of Computer Science
Computational Linguistics Group
Winterthurerstr. 190
CH-8057 Zurich
{volk, siclemat}@ifi.unizh.ch

Abstract: We have explored and implemented different approaches to named entity recognition in German, a difficult task in this language since both regular nouns and proper names are capitalized. Our goal is to identify and recognise person names, geographical names and company names in a computer magazine corpus. Our geographical name classifier works with precompiled lists but our company name classifier learns the names from the corpus. For the recognition of person names we work with a precompiled list of first names and the program learns the last names. For this classifier we suggest setting an activation value for the last name and subsequently depriming the value until "forgetting" the name. Our evaluation results show that our mixed approaches are as good as the recall and precision values reported for English. It is shown that a carefully tuned cascade of name classifiers can even distinguish between different interpretations of a name token within the same document.

1 Introduction

We are working on a method for the disambiguation of PP-attachment ambiguities in German. It relies on competing cooccurrence strengths between the noun and the preposition (N+P) and the verb and the preposition (V−P). Therefore we have computed these cooccurrence strengths from a corpus. We chose to work on a computer magazine corpus since it is a semi-technical text which displays features from newspapers (some articles are very short) and from technical texts (such as many abbreviations, company and product names). We selected the ComputerZeitung [Ko98], a weekly computer magazine. The raw texts contain around 1.4 million tokens per year. One important step in the corpus preparation is the recognition of named entities.

1.1 General corpus preparation

At the start all corpus texts are in pure text format. There is no formatting information except for a special string that marks the beginning of an article. In order to enrich the corpus with semantic information the texts had to be processed in various steps. All programming was done in Perl.

1. **Clean up.** The texts have been dehyphenated by the publisher before they were distributed (with few exceptions). There are blanks that are not token delimiters. Our corpus contains blanks within long sequences of digits such as numbers over 10 000 and telephone numbers. We substitute these blanks with auxiliary symbols (e.g. a dash) to facilitate tokenization.

2. **Recognition of text structure.** Headlines are often elliptical sentences (*Mehrwert gefunden, Arbeitsplätze gesucht*). They cause many tagging errors since the tagger has been trained over complete sentences. Therefore we have to recognize and mark headers, regular paragraphs, and list items in order to treat them specifically. A header is a line that ends without a sentence final punctuation marker. We use SGML tags to mark these items and all other meta-information. We identify newspaper specific article starters. Most articles begin with a city name and an abbreviation symbol for the author.

(1) "*Bonn (pg)* - Bundesregierung und SPD kamen sich ...".

3. **Recognition of sentence boundaries.** Sentences end at the end of a paragraph or with a sentence finishing punctuation symbol (a full stop, an exclamation mark, a question mark). Unfortunately a full stop symbol is identical to a dot that ends an abbreviation (*mit Dr. Neuhaus*) or an ordinal number (*Auf dem 5. Deutschen Softwaretag*). We use an abbreviation list with 1200 German abbreviations to distinguish the dot from a full stop.

4. **Verticalization of the text.** The text is then verticalized (one word per line). Punctuation marks are considered separate tokens and thus each occupies a separate line.

2 Recognition and Classification of Named Entities

At some point in the processing we have to recognize and classify proper names for e.g. persons, geographical locations, companies, and products. One could argue that the recognition of proper names is a task for a part-of-speech tagger and therefore classification should be done after tagging. But [VS98] have shown that a tagger's distinction between proper nouns and regular nouns is not reliable in German. This distinction is the main source of tagging errors. In German both proper nouns and regular nouns are spelled with an initial capital letter and their distributional properties are not distinct enough to warrant a clear tagger judgement.

Therefore we decided to recognize and classify named entities before tagging, in this way helping the tagger with the difficult task of noun classification. Later, classified proper names will help us build semantic clusters (all person names, all geographical entities, ...). These clusters will be used to reduce the sparse data problem in the frequency counts over our corpus.

2.1 Previous approaches to named entity recognition

Named entity recognition is a topic of active research especially in the context of message understanding and classification. The approaches use internal evidence (keywords, gazetteers) and external evidence (the context). Of course, most problematic is the classification of unknown names. Different algorithms have been used to learn names and their classification from annotated texts and from raw corpora. Evaluation figures are difficult to compare.

[MM96] stress the importance of representing uncertainty about name hypotheses. Their system exploits the textual structure of documents to classify names and to tackle coreference. In particular it exploits appositives to determine name categories (e.g. *X, a small Bay Area town* → *X* = city name). A newly introduced name leads to the generation of a normalized name, name elements and abbreviations so that these forms are available for coreference matching. The system works in two passes. It first builds hypotheses on name chunks (sequences of capitalized words). Second, it groups these name chunks into longer names if there are intervening prepositions or conjunctions. They report on 85% precision and 67% recall on 42 hand-tagged Wall Street Journal articles with 2075 names.

[SG00] describe the LaSIE system which combines list lookup, part-of-speech tagging, name parsing, and name matching. They have experimented with learning name lists from annotated corpora. They show that carefully compiled lists cleaned through dictionary filtering and probability filtering lead to the best results. Dictionary filtering means removing list items which also occur as entries in the dictionary. But this should only be done if a word occurs more frequently as non-name than as name in the annotated data (probability filtering). This approach is similar to ours.

Most of the research on the classification of named entities is for English. In particular, there are very few publications on German name recognition. One is [La99] describing briefly the PRONTO system for person name recognition. He uses a combination of first name lists, last name lists, heuristics ("a capitalized word following a first name is a last name"), context information ("a determiner in front of a hypothetical person name cancels this hypothesis") and typical letter trigrams over last names. He reports precision and recall figures of 80%. Another is [PN00] which deals with organization, person and location name recognition. They report a precision of 95% and a recall of 85% but it is not clear whether these figures relate to the recognition task (telling apart a proper name from a regular noun) or the classification task.

2.2 Recognition of person names (learn - apply - forget)

Our approach to person name recognition relies on the observation that there is a rather stable set of personal first names. Additionally, we find that a person's last name is usually introduced in a text with either his/her first name, with a title (*Dr.*, *Prof.*), or a word describing his/her profession or function (manager, director, developer, ...).

Therefore we use a list of 16 000 first names and another list of a dozen titles as keywords to find such name pairs (keyword followed by a capitalized word). The name list contains mostly German and English first names with many different spelling variations (e.g. *Jörg, Joerg, Jürg, Jürgen*). It is derived from searching through machine readable telephone books. Our recognition program "**learns**" the last name, a capitalized word that follows the first name. The last name will then be recognized (**applied**) if it occurs standing alone in subsequent sentences.

(2) Beim ersten Internet-Chat-in von EU-Kulturkommissar **Marcelino Oreja** mußten die Griechen "leider draußen bleiben". **Oreja**, ..., beantwortete unter Zuhilfenahme von elf ...

This approach, however, leads to two problems. First, the program may incorrectly learn a last name if e.g. it misinterprets a company name (*Harris Computer Systems*) or if there is a first name preceding a regular noun (... *weil Martin Software entwickelt*). Second, a last name correctly learned in the given context might not be a last name in all subsequent cases (consider the person name *Michael Dell* and the company name *Dell*). Applying an incorrectly learned last name in all subsequent occurrences in the corpus might lead to hundreds of erroneously recognized names.

Therefore we use the observation that a person name is usually introduced in a document in either full form (i.e. first name and last name) or with a title or job function word. The last name is thereafter primed for a certain number of sentences in which it can be used standing alone. If it is used again later in the text it needs to be reintroduced. So, the question is, for how many sentences does the priming hold. We use an initial value of 15 and a refresh value of 5. This means that a full name being introduced is activated for 15 subsequent sentences. In fact, its activation level is reduced by 1 in every following sentence. After 15 sentences the program "**forgets**" the name. If, within these 15 sentences, the last name occurs standing alone the activation level increases by 5 and thus keeps that name active for 5 more sentences.

```
foreach sentence {
  if full_name(first_name|title, last_name) {
    activation_level[last_name] +  15;
  }
  elsif match(last_name) && (activation_level[last_name] > 0) {
    activation_level[last_name] +  5;
  }
  elsif end_of_document {
    foreach last_name {
      activation_level[last_name]   0;
  } }
  else {  ## sentence without last_name
    foreach last_name {
      if activation_level[last_name] > 0 {
        activation_level[last_name]--;
} } } }
```

We found the initial activation value by counting the number of sentences between the introduction of a full name and the subsequent reuse of the last name standing alone. In an annual volume of our corpus we found 2160 full names with a reused last name in the same document. In around 50% of the cases the reuse happens within the following two sentences. But the reuse span may stretch up to 30 sentences. With an initial activation value of 10 we miss 7%, but with a value of 15 only 3% of reused names. We therefore decided to set this level to 15. We also experimented with a lower refresh value of 2. Against our test set we found that we are loosing about 10% recall and therefore kept the refresh value at 5.

In another experiment we checked all documents of an annual volume of our corpus for recognized last names that reoccur lateron in the document without being recognized as last names. For an initial activation value of 10 we found 209 such last name tokens in 6027 documents. The initial value of 15 only resulted in 98 unrecognized last name tokens (about 1% improved recall) with only 6 erroneously recognized items (a negligible loss in precision).

With this priming activation algorithm we delimit the effect of erroneously learned last names to the priming area of the last name. The priming area ends in any case at the end of the document. Note that this algorithm allows a name to belong to different classes within the same document. We have observed this in our corpus especially when a company name is derived from its founder's name and both are mentioned in the same document.

(3) Der SAP-Konkurrent **Baan** verfolgt eine aggressive Wachstumsstrategie. ... Das Konzept des Firmengründers **Jan Baan** hat Erfolg.

These findings contradict the one-sense-per-document hypothesis brought forward by [GCY92]. They had claimed that it is possible to combine all contextual evidence of all occurrences of a proper noun from one document to strengthen the evidence for the classification. But in our corpus we find dozens of documents in every annual volume where their hypothesis does not hold.

Included in our algorithm is the use of the genitive form of every last name (ending in the suffix *-s*). Whenever the program learns a last name it treats the genitive as a parallel form with the same activation level. Thus the program will also recognize *Kanthers* after having learned the last name *Kanther*.

(4) Wie es heißt, gewinnen derzeit die Hardliner um Bundesinnenminister **Manfred Kanther** die Oberhand. ... **Kanthers** Interesse gilt der inneren Sicherheit.

If a learned last name is also a first name our system regards it as last name for the priming span. If it occurs standing alone it is recognized as last name if it is not followed by a capitalized word. An immediate capitalized successor will trigger the learning of a new last name. This strategy is succesful in most cases (cp. 5) but leads to rare errors as exemplified in 6. The trigger is not applied for the genitive form since this form as such is not in the list of first names.

(5) "Im Juli werden die ersten Ergebnisse des San-Francisco-Projekts ausgeliefert", veranschaulicht **Julius Peter**....
... ergänzt Lawsons Cheftechnologie **Peter Patton.**

(6) Am Anfang war die Zukunftsvision von einem künftigen Operationssaals, die der Neurochirurg **Volker Urban** von der Dr.-Horst-Schmidt-Klinik ...
... räumt **Urban *Akzeptanzprobleme** ein.

In an evaluation of 990 sentences from our computer magazine corpus we manually detected 116 person names. 73 of these names are full names and 43 are stand alone last names. Our algorithm achieves a recall of 93% for the full names (68 found) and of 74% for the stand alone names (32 found). The overall precision is 92%.

The algorithm relies on last names being introduced by first names or titles. It will miss a last name that occurs without this introduction. In our corpus this (rarely) happens for last names that are very prominent in the domain of discourse (*Gates*) and in cataphoric uses, mostly in headlines where the full name is given shortly after in the text.

(7) <h2>**McNealy** präzisiert Vorwürfe gegen **Gates**1</h2>
... Suns Präsident **Scott McNealy** hat auf der IT Expo ...

2.3 Recognition of geographical names (list - learn - apply)

Names of geographical entities (cities, countries, states and provinces, mountains and rivers) are relatively stable over time. Therefore it is easy to compile such lists from resources in the WWW. In addition we exploit the structure of our newspaper texts that are often introduced with a city name (cf. step 2). We collected (**learned**) all city names used in our computer magazine corpus as introductory words as well as (German) city names from the WWW into a gazetteer of around 1000 city names. We also use a list of 250 country names (including abbreviations like *USA*) and (mostly German) state names. When matching these geographical names in our corpus we have to also include the genitive forms of these names (*Hamburgs, Deutschlands, Bad Sulzas*). Fortunately, the genitive is always formed with the suffix *-s*.

A more challenging aspect of geographical name recognition is their adjectival use, frequently as modifiers to company names or other organizations.

(8) Das gleiche gilt für die zur **Londoner** Colt Telecom Group gehörende **Frankfurter** Colt Telecom GmbH, ...

(9) Die **amerikanische** Engineering Information und das **Karlsruher** Fachinformationszentrum wollen gemeinsam ...

We decided to also mark these adjectives as geographical names since they determine the location of the company or organization. The difficulty lies in building a gazetteer for these words. Obviously, we are unlikely to find a gazetteer of derived forms in the WWW. But it is also difficult to derive these forms systematically from the base forms due to phonological deviations.

(10) London > Londoner; Karlsruhe > Karlsruher; München > Münchner

(11) England → englische/r; Finnland → finnische/r

As these examples show, both *-er* and *-isch* can be used as derivational suffixes to turn a geographical name into an adjective. *-isch* is the older form but it has been pushed back by *-er* since the 15th century (cf. [FB95] p.240). While *-isch* is used to build a fully inflectional lower case adjective, *-er* is used to form an invariant adjective that keeps the capitalized spelling of the underlying noun. There is currently a strong tendency to use the *-isch* form for country names and the *-er* form for city names. Rarely, both forms are used side by side (*schweizerisch, Schweizer*). For all country names we manually compiled the list of the *-isch* base form of the adjectives. For the city names we are faced with a much larger set. We therefore used the morphological analyzer Gertwol to identify such words. According to [HM94] Gertwol comprises around 12 000 proper names out of which 2600 are geographical names. For every geographical name Gertwol derives a masculine and a feminine form for the inhabitants (*Bremer, Bremerin*) as well as the form for the adjective.

The capitalized geographical adjectives are therefore homographic to nouns denoting a masculine inhabitant of that city or state and also to the plural form of the inhabitants (*die Bremer sind* ...). We use this ambiguity to identify geographical adjectives in the Gertwol output: If a capitalized word ending in *-er* is analyzed as both a proper noun and an invariant adjective then this word will be a geographical adjective and we can list it in a special gazetteer.

In our corpus we mark all forms of the lower case geographical adjectives. For the capitalized adjectives we mark all occurrences that are followed by a capitalized noun. Occurrences followed by a lower case word are likely to stand for the inhabitant reading (as in 12).

(12) Vor fünf Jahren hatten sich die **Redmonder** bei der Forschung ...

In our evaluation of 990 sentences we manually found 166 geographical names. Out of these 151 were automatically marked (a recall of 91%). The algorithm incorrectly marked 36 geographical names (a precision of 81%). It should be noted, however, that there are rare cases of ambiguities between geographical names and regular nouns (e.g. *Essen, Halle, Hof* are names of German cities as well as regular German nouns). There are also ambiguities between geographical names and person names (e.g. the first name *Hagen* is also the name of a German city). Many city names can also be used as personal last names. But these ambiguities hardly ever occur in our corpus.

2.3 Recognition of company names (learn - filter - apply)

Company names are very frequent in our computer magazine corpus since most articles deal with news about hardware and software products and companies. Our algorithm for company name recognition is based on keywords that indicate the occurrence of a company name. Based on this we have identified the following patterns:

1. A sequence of capitalized words after strong keywords such as *Firma*. The sequence can consist of only one such capitalized word and ends with the first lower case word. The keyword is not part of the company name.

(13) ... das Software-System "DynaText" der Firma **Electronic Book Technologies.**

2. A sequence of capitalized words preceding keywords such as *GmbH, Ltd., Inc., Oy.* The sequence can consist of only one such capitalized word and ends to the left with the first lower case word or with a geographical adjective or with a feminine determiner. The keyword is considered to be part of the company name.

(14) In Richtung Multimedia marschiert **J. D. Edwards & Co.** (JDE) mit ihrem kommerziellen Informationssystem ...

3. According to German orthographical standards a compound consisting of a proper name and a regular noun is spelled with a hyphen. We can exploit this fact and find company names in hyphenated compounds ending in a keyword such as *Chef, Tochter*.[1]

(15) ... ist die Zukunft der deutschen **France-Télécom**-Tochter geklärt.

4. Combining evidence from two or more weaker sources suffices to identify candidates for company names. We have found two useful patterns involving geographical adjectives.

4.a. A sequence of capitalized words after a feminine article followed by a geographical adjective.

(16) Für Ethernet- und Token-Ring-Netze hat die Münchner **Ornetix** einen Medienserver entwickelt.

4.b. A sequence of capitalized words after a geographical adjective and a weak keyword (like *Agentur, Unternehmen*)[2] Neither the adjective nor the keyword is part of the company name.

(17) Das Münchner Unternehmen **Stahlgruber** zählt zu den wenigen Anwendern, die ...

Using these patterns our program "**learns**" simple and complex company names and saves them in a list. All learned company names constitute a gazetteer for a second pass of name application over the corpus. The learning of company names will thus profit from enlarging the corpus while our recognition of person and geographical names is independent of corpus size.

Complex company names consist of two or more words. The complex names found with the above patterns are relatively reliable. Most problems arise with pattern 2 because it is difficult to find all possible front boundaries (cf. *das Automobilkonsortium Micro Compact Car AG*). Our algorithm sometimes includes unwanted front boundaries into the name.

Often acronyms refer to company names (*IBM* is probably the best known example). These acronyms are frequently introduced as part of a complex name. We therefore search complex names for such acronyms (all upper case words) and add them to the list of found names.

(18) ... die **CCS Chipcard & Communications GmbH**. Tätigkeitsschwerpunkt der **CCS** sind programmierbare Chipkarten.

Learning single word company names is much more error prone. It can happen that a capitalized word following the keyword *Firma* is not a company name but a regular noun (... *weil die Firma Software verkauft*) or that the first part of a hyphenated compound with *Chef* is a country name (*Abschied von Deutschland-Chef Zimmer*). Therefore

[1] We owe this observation to our student Jeannette Roth.

[2] We distinguish between strong keywords that always trigger company name recognition and weak keywords that are less reliable cues and need to cooccur with a geographical adjective.

we need to **filter** these one-word company names before applying them to our corpus. We use Gertwol to analyse all one-word names. We accept as company names all words

- that are unknown to Gertwol (e.g. *Acotec, Belgacom*), or
- that are unknown to Gertwol (e.g. *Acotec, Belgacom*), or
- that are known to Gertwol as proper names (e.g. *Alcatel, or Apple*),
- that are recognized by Gertwol as abbreviations (e.g. *AMD, AT&T, Be*), and
- that are not in an English dictionary (with some exceptions like *Apple, Bull, Sharp, Sun*).

In this way we exclude all regular (lexical) nouns from the list of simple company names. In a separate pass over the corpus we then **apply** all company names collected in the learning phase and cleared in the filter phase. In the application process we also accept genitive forms of the company names (*IBMs, Microsofts*).

Note that the order of name recognition combined with the rather cautious application of person names leads to the desired effect that a word can be both person name and company name in the same corpus. With the word *Dell* we get:

sentence	example	type
6917	*Auch IBM und* **Dell** ...	company
11991	*Michael* **Dell**	person
11994	... *warnte* **Dell**	person
12549	*Siemens Nixdorf,* **Dell** *und Amdahl ...*	company

In our evaluation of 990 sentences the program found 283 out of 348 company name occurrences (a recall of 81%). It incorrectly recognized 89 items as company names that were not companies (a precision of 76%). These values are based on completely recognized names. Many company names, however, consist of more than one token. In our evaluation text 50 company names consist of two tokens, 13 of three tokens, 3 of four tokens and 1 of five tokens (*Challenger Gray & Christmas Corp.*). We therefore performed a second evaluation of company names checking only the correct recognition of the first token. We then get a recall of 86% and a precision of 80%.

With these patterns we look for sequences of capitalized words. That means we miss company names that are spelled all lower case (against traditions in German). We also have problems with names that contain function words such as conjunctions or prepositions. We will only partially match these names.

3 Part-of-Speech Tagging the Corpus

In order to extract nouns, verbs and prepositions we need to identify these parts-of-speech in the texts. Before we decided on a part-of-speech tagger we made a detailed comparative evaluation of the Brill-Tagger (a rule-based tagger) and the Tree-Tagger (a statistics-based tagger) for German. We showed that the Tree-Tagger was slightly better [VS98]. Therefore we use the Tree-Tagger [SK] in this research.

The Tree-Tagger uses the STTS (Stuttgart-Tübingen Tag Set; [TS96]), a tag-set for German with around 50 tags for parts-of-speech and 3 tags for punctuation marks. The STTS distinguishes between proper nouns and regular nouns, between full verbs, modal verbs and auxiliary verbs, and between prepositions, contracted prepositions and postpositions.

The tagger works on the vertical text (each word and each punctuation mark in a separate line). In addition, in our corpus the tagger input already contains the proper name tag (NE) for all previously recognized names in noun function and the adjective tag (ADJA) for all recognized names in attributive function. The tagger assigns one part-of-speech tag to every word in a sentence. It does not change any tag provided in the input text. Thus the prior recognition of proper names ensures the correct tags for these names and improves the overall tagging quality. After tagging some missed sentence boundaries can be inserted. If, for instance, a number plus dot (suspected to be an ordinal number) is followed by a capitalized article or pronoun, there must be a sentence boundary after the number (*... freuen sich über die Werbekampagne für Windows 95. Sie steigert ihre Umsätze*). We find between 75 and 130 such sentence boundaries per annual volume.

4 Conclusion

We have shown that named entity recognition in German will profit from precompiled lists as well as from learning and filtering. We employ carefully tuned algorithms for person, geographical, and company name recognition. Person name recognition is most reliable with 86% recall and 92% precision. The most difficult is company name recognition (81% recall and 76% precision). This is due to the fact that company names vary greatly in length and structure. Lately some company names are even spelled with lower case letters.

Proper name recognition helps to improve tagging performance. The problem of distinguishing between regular nouns and proper names is greatly reduced (cf. [CV01]).

Of course, there are names beyond persons, companies and geographical locations. In a next step we will work on product name recognition. Moreover our corpus contains names of exhibitions like *Orbit, Cebit, Comdex* and organisations like *Gesellschaft für Informatik*.

Bibliography

[CV01] Clematide,S.; Volk, M.: Linguistische und semantische Annotation eines Zeitungskorpus. In: Proc. of GLDV-Jahrestagung, Giessen, 2001.

[FB95] Fleischer,W.; Barz, I.: Wortbildung der deutschen Gegenwartssprache. Niemeyer, Tübingen, 1995.

[GCY92] Gale, W.A.; Church, K.W.; Yarowsky, D.: One sense per discourse. In: Proc. of DARPA speech and Natural Language Workshop, Harriman, NY, 1992.

[HM94] Haapalainen, M.; Majorin, A.: Gertwol. Ein System zur automatischen Wortformerkennung deutscher Wörter. Lingsoft Oy, Helsinki, 1994.

[Ko98] Konradin-Verlag: Computer Zeitung auf CD-ROM. Volltextrecherche aller Artikel der Jahrgänge 1993 bis 1998. Konradin Verlag; Leinfelden-Echterdingen. 1998.

[La99] Langer. H.: Parsing-Experimente. Habilitationsschrift, Universität Osnabrück, 1999.

[MM96] Mani, I.; MacMillan, T. R.: Identifying unknown proper names in newswire text. In: Boguraev, B.;Pustejovsky, J. (editors): Corpus Processing for Lexical Acquisition. MIT Press, Cambridge,MA, 1996: pp. 41-59.

[PN00] Piskorski, J.; Neumann, G.: An intelligent text extraction and navigation system. In: Proc. of 6th International Conference on Computer-Assisted Information Retrieval (RIAO-2000), Paris, April, 2000.

[SK96] Schmid, H.; Kempe, A.: Tagging von Korpora mit HMM, Entscheidungsbäumen und Neuronalen Netzen. In: Feldweg, H.; Hinrichs, E.W. (editors): Wiederverwendbare Methoden und Ressourcen zur linguistischen Erschliessung des Deutschen, volume 73 of Lexicographica. Series Maior, Niemeyer Verlag, Tübingen, 1996: pp. 231-244.

[SG00] Stevenson, M.; Gaizauskas, R.: Using corpus-derived name lists for named entity recognition. In: Proc. of ANLP, Seattle, 2000.

[TS96] Thielen, C.; Schiller, A.: Ein kleines und erweitertes Tagset fürs Deutsche. In: Feldweg, H.; Hinrichs, E.W. (editors): Wiederverwendbare Methoden und Ressourcen zur linguistischen Erschliessung des Deutschen, volume 73 of Lexicographica. Series Maior. Niemeyer Verlag, Tübingen, 1996: pp. 193-203.

[VS98] Volk. M.; Schneider, G.: Comparing a statistical and a rule-based tagger for German. In: Schröder, B.; Lenders,W.; Hess, W.; Portele, T. (editors): Computers, Linguistics, and Phonetics between Language and Speech. Proc. of the 4th Conference on Natural Language Processing - KONVENS-98. Bonn, 1998: pp. 125-137.