Combining a Rule-based Tagger with a Statistical Tagger for Annotating German Texts

Stefan Klatt
Computer Science Department
University of Stuttgart
Breitwiesenstr. 20-22
70565 Stuttgart, Germany
klatt@informatik.uni-stuttgart.de

Abstract

This paper presents a rule-based tagger for unsupervised POS-Tagging of German sentences. The tagging process that uses lexical, syntactic and corpus-based information is organized in several stages by processing easier decisions before the harder ones. A few very difficult decisions are marked for later manual checking. Using the full sentential context, the tagger avoids certain errors made by statistical taggers and allows to formulate sophisticated rules. In an experiment, the tagger achieved a recall rate of 99,50% and a precision rate of 97,29%. Resolving the remaining ambiguities with the statistical *TreeTagger*, one of the best taggers for German to date, results in an accuracy rate of 99,36%. This is a 50% error reduction compared to the isolated use of the *TreeTagger*. Finally, checking the marked decisions manually leads to a further improved accurary rate of 99.51%.

1 Introduction

Due to the availability of the increasing amount of manual tagged training corpora, supervised statistical taggers have become more and more the standard in NLP tagging applications. They are very fast, they can be used language independently and they produce reasonable results. But there is still one big advantage left for using a rule-based tagger: the accuracy. Using the whole sentential context, it shouldn't be a problem for it to outperform a trigram tagger, which operates with a limited context window of three words. Regarding English, the ENGCG2-Tagger (Tapainainen, 1996) ist still the benchmark, as proved independently by (Samuelsson and Voutilainen, 1997). Surprisingly, the situation is different for German. In a comparison between the statistical *TreeTagger* (Schmid, 1999) and the rule-based Brill-Tagger (Brill, 1994) by (Volk and Schneider, 1998), the *TreeTagger* outperformed the Brill-Tagger. This is astonishing, since German has not such a rigid word order as English and seems therefore more difficult to tag with a statistical tagger as with a rule-based one. But, taking a closer look at the German tagging scene, the statistical victory seems to be more based on the lack of rule-based taggers. Apart from the *TreeTagger* and the *TnT-Tagger* (Brants, 2000), we find a lot of other statistical taggers (Lezius et al., 1996), but no *real* rule-based system. In fact, the Brill-Tagger is no such one in opposite to the English ENGCG2-Tagger, since its rules are the product of a data-driven learning approach. But, if we consider the limitations of statistical tagging and some typical errors the *TreeTagger* produces, it should't be a problem to create a *real* rule-based tagger with a higher accuracy rate – in isolation or in combination with the *TreeTagger*. In the latter case, this could be done with a small effort.

2 Motivation of Rule-based Tagging

2.1 Limitations of Statistical Tagging

German is a free word order language with left and right peripheral heads. It has therefore a few linguistic constructions that are a huge problem for a statistical trigram tagger. Looking for instance at ambiguous clause-initial elements in embedded verb-final clauses, in most of the cases we find the corresponding finite verb more than two words away. Regarding vice versa verbs in verb-final position, it is nearly impossible to decide with a three word context window, whether the verb has a finite reading as in an embedded clause (1). It could also have an infinitive reading (2) or a past participle reading (3) as part of a discontinuous verbal complex in a main clause, in which the finite verb occurs in second position of the clause.

(1) Ich glaube, dass sie es wieder einmal vergessen/VVFIN[1].
I believe that they forget it once again.

(2) Er könnte es wieder einmal vergessen/VVINF.
He could forget it once again.

(3) Er hat es wieder einmal vergessen/VVINF.
He has forgotten it once again.

Looking at the 36 million token *Stuttgarter-Zeitungs*-corpus (*STZ*-corpus) tagged with the *TreeTagger*, a lot of occurences of the above mentioned problems were wrongly tagged. But these were not the only errors the *TreeTagger* produced.

2.2 Avoidable Errors of the *TreeTagger*

We also found wrong tag assignments, where a three word window should be sufficient to assign the correct tag from a linguistic point of view. For instance, the infinitive marker *zu* can only be followed by a verb in the infinitive or an optional quotation mark between these two elements. But looking at the *STZ*-corpus, we find in nearly 1% of all occurences another right adjacent tag. Furthermore, looking at capitalized tagged verb imperatives, more than 50% of them occured in the middle of a clause with no left adjacent punctuation mark. In fact, these words can only have a noun reading. Or, if we consider words left adjacent to the indefinite pronoun *man*, we found in 2% of the cases an article assignment, what is an ungrammatical constellation in German.

3 Features of our Rule-based Tagger

3.1 Easy-first in Several Stages

Like the ENGCG-2 tagger, we prefer an implicit disambiguation than an explicit one. We are looking out for ungrammatical constructions and eliminate the readings that are responsible for such a construction, step by step. In (4), we could eliminate at first all relative pronoun readings (PRELS) in sentence-initial position. Next, we eliminate the article reading (ART), since a disambiguous finite verb reading is following - an impossible construction in German. And that is all we have to do, since only the reading of a demonstrative pronoun (PDS) remains for the word *das*.

(4) Das/~~PRELS~~/~~ART~~/PDS ist/VAFIN richtig!
That's right!

3.2 Abandoning the Black Box Mentality

In some cases, the whole sentential context is not sufficient for a disambiguation as could be seen in (5) and (6). Whether the marked word is a predicative adjective (ADJD) or a past participle verb (VVPP) could only be resolved by the sentence-external context. Instead of assigning a tag in a black box manner, as a statistical tagger does, we mark this problem for later manual checking.

(5) Er war verrückt/ADJD.
He was crazy.

(6) Er war verrückt/VVPP.
It was moved.

3.3 Extracting Information Detected during the Tagging Process

Sometimes it would be fine to extract knowledge that was acquired during the tagging process. For example, if we find an unknown word right adjacent to a capitalized personal pronoun (PPER) that has only a nominative reading as in (7), we can be sure that it must be a finite verb. If we could extract this information, we can extend our lexicon with this verb and other flective forms of it manually or automatically by a corpus-based search.

(7) Er/PPER leaste/? ein Auto.
He leased a car.

3.4 Using Corpus Information

We use a raw corpus for checking the correct reading of separated verb particles (PTKVZ) in the way that we merge the verb particle with the next preceding finite verb and look up the composed form in our lexicon. If we don't find it, we compute how many times the composed word occurs in the *STZ*-corpus and store this information in an additional file. If the composed word doesn't occur in the corpus, we delete the PTKVZ-reading of this word. Otherwise, we could extend our lexicon after a manual check.

As mentioned in the last section, we could also make use of a raw corpus to acquire the correct reading of an unknown word and to extend our lexicon by this information. At the

[1] In the examples the relevant words marked by the corresponding tags of the STTS, the **S**tuttgart-**T**übingen-**T**ag**S**et (Schiller et al., 1999), which was used in the experiment in section 5. Each time a new STTS-tag is introduced, it is explained in its immediate context.

moment, we didn't integrate such a component into our tagger as yet, but we plan to do this in the near future.

3.5 Using the Topological Field Model

For the formulation of several disambiguation strategies, described in section 4.1, we make use of an extension of the topological field model (TFM) (Rehbein, 1992), which segments a sentence in smaller, well defined fields.

The extended TFM consists of seven fields. The fields LK (linke Klammer ≈ left bracket) and RK (rechte Klammer ≈ right bracket) divide the rest of the clause in three fields: VF (Vorfeld ≈ top field), MF (Mittelfeld ≈ middle field) and NF (Nachfeld ≈ bottom field). The sequence LK, MF and RK is also defined as SK (Satzklammer ≈ clause bracket). Conjunctions and punctuation marks can be positioned in the fields SAR (Satzanfangsrahmen ≈ left clause frame) and SER (Satzenderahmen ≈ right clause frame), which are the extension of the standard TFM.

SAR	VF	LK	MF	RK	SER	NF
,		daß	er sie	sah	.	
	Er	hat	sie	gesehen	.	
		Hat	er sie	gesehen	?	
Und	sie	hat	mehr	gesehen		als er

Figure 1: Extended topological field model

4 System Architecture

4.1 Stages of the Rule-based Tagger

The disambiguation task is divided into twelve stages, which are applied after the lexicon lookup. Our lexicon is a combination of a small full form lexicon and a morphology system. The full form lexicon has approximately 3000 entries. It contains all functional class words and approximately 1500 proper nouns. Furthermore, the lexicon consists of parts of multi word lexemes (MWLs) and a few other lexemes being involved in a lexical-driven disambiguation process. The lexicon lookup starts after the last word of the input sentence was read.

Next, we apply lexical-driven rules triggered by some of our lexical entries to find MWLs and to do some lexical-driven disambiguations. For instance, if the word *AG* (engl. *Inc.*) occurs in a sentence, we apply the following rules, which are linked with our corresponding lexical entry: We are going from the word *AG* to the left, until we reach the next definite article. If there is no preposition or another determiner in between, we consider the whole sequence without an optional adjective of origin in the front as a company name and tag all capitalized words in between as proper nouns (NE). In (8), this step overrules the previously unambiguous common noun reading of the word *Gebrüder*.

(8) der/ART Schweizer/ADJA Gebrüder/NE
Sulzer/NE AG/NN
The Swiss Brothers Sulzer Inc.

Next, we try to find possible readings for unknown words. If we have a capitalized unknown word that isn't in a clause-initial context, we assign a common noun (NN) and a proper noun (NE) reading to it. If it additionally ends with an attributive adjective suffix and it is left adjacent to a noun, we also assign an attributive adjective reading (ADJA) to it. At last, we check whether it is surrounded by quotation marks. If this is the case, we add a foreign material tag (FM) to it and all other words in between the quotation marks. For lowercase written unknown words, we only test an ADJA- or a FM-reading described in the way before. Next, we add to every capitalized word that follows a Christian name a proper noun reading, if this wasn't done during the lexicon lookup before.

In **stage 1**, we do some trivial eliminations. For example, we delete all verb imperative readings, if the candidates aren't in a clause-initial context. We also delete separated verb particle readings, if the candidates aren't in a clause-final context or preceded by a finite verb.

In **stage 2**, we consider verb readings and readings of clause-initial elements. Depending on how many finite verbs we have in the sentence we do the following: If there is only one finite verb that isn't in a sentence-final context, we eliminate all readings of clause-initial-elements of finite embedded clauses. If we have more than one verb in the sentence, we generate for all neighbouring verbs their continuous and/or discontinuous verbal complex and eliminate only those clause-initial element candidates that don't occur in a clause-initial context. In (9) and (10), we have more than one candidate for a finite verb reading. According to the above mentioned strategy, we build in both cases a continuous and a discontinuous verbal complex. The disambiguation of the correct verbal complex will be postponed till stage 7.

(9) Sie werden es [$_{VC}$ vergessen haben].
They will have forgotten it.

(10) Er sagte, dass sie es [$_{VC}$ vergessen haben].
He said that they have forgotten it.

In **stage 3**, we eliminate readings of verb infinitives (VVINF) having no right adjacent verb that governs a VVINF-form, as for instance a finite modal verb (VMFIN), or if it isn't preceded by such a verb. After that, we apply the same strategy on participle verbs (VVPP) and VVPP-governing verbs as for instance a finite auxiliary verb (VAFIN).

In **stage 4**, we try to benefit from the ranking order of pronouns in German clauses. If a pronominal verb argument isn't in a clause-initial position, it tends to occur immediately after the finite verb if it isn't governed by a preposition. We apply the whole rules of stage 4 only to reflexive pronouns and pronouns with an unambiguous nominative reading and the indefinite pronoun *man* as well as to other personal pronouns in a checked clause-initial context in the following way: If a pronoun in a clause-initial context is followed by a word with a finite verb reading as in (11), we eliminate all other readings of such a word. If the pronoun doesn't have a clause-initial context, we consider its left adjacent word. If this has a finite verb reading as in (12), we eliminate all other readings of it.

(11) Wir kommen/VVFIN/~~VVINF~~ morgen!
We are coming tomorrow!

(12) Heute gehen/VVFIN/~~VVINF~~ wir in die Stadt.
Today, we are going into town.

In **stage 5**, we eliminate spurious verb and adjective readings with different strategies. In the first strategy, we do for each word with only one lexical verb reading and one or more nonverbal readings the following: We go from such a word to the next left or right word with an unambiguous lexical verb reading, whereas it doesn't matter whether the verb was assigned different verbal tags. If there is no clausal border element as a comma or a conjunction between these two words, we have an ungrammatical constellation, since only one lexical verb can occur in a simple clause. All other verbs must be non-lexical verbs. That is, if we have two or more words with a lexical verb readings inside a clause and only one of them has no other reading, we eliminate the verb readings of the other words inside the clause. In (13), this leads to the elimination of the verb reading of *interessierten*, since *klatschen* (engl. *to applause*) only has a VVFIN- and a VVINF-reading, of which the VVINF-reading was eliminated in stage 3.

(13) Die interessierten/~~VVFIN~~ Zuhörer klatschen/VVFIN.
The interested listeners spend applause.

In the second strategy, we eliminate spurious finite verb readings in the same manner as before. In (14), this leads to the elimination of the verb reading of *interessiert*, since *hat* only has a finite verb reading as auxiliary verb (VAFIN) or as a lexical verb (VVFIN).

(14) Sie hat/VAFIN sich dafür interessiert/~~VVFIN~~.
She showed interest for that.

Finally, we investigate words that have a past participle reading (VVPP) and a predicative adjective reading (ADJD). If such a word is left adjacent to an auxiliary verb with the exception of a form of the verb *sein*, we only keep the VVPP-reading. In the latter case, we make no disambiguation and store this information for later manual checking, since we can't be sure to do the right decision (see section 3.2). If we don't have a right adjacent verb, we're going to the next left auxiliary verb in the sentence, if possible. If we passed now clausal border element, we act as mentioned before.

In **stage 6**, we do a partial syntactic analyses of elementary noun phrases. If a noun phrase contains attributive adjectives and determiners that have no additional relative pronoun reading, we disambiguate these elements. In (15), we disambiguate the article and the adjective in the recognized noun phrase, whereas in (16) we only disambiguate the adjective, because the article doesn't agree with the noun phrase in its relevant morphosyntactic features.

(15) dass ihr [$_{NP}$ die/ART roten/ADJA Schuhe] gefallen
that she likes the red shoes

(16) dass die [$_{NP}$ rote/ADJA Schuhe] trägt
that she wears red shoes

In **stage 7**, we segment the sentence in topological fields (see section 3.5). Now, we are able to disambiguate the ambiguous verbal complexes of the examples in (9) and (10), repeated here as (17) and (18).

(17) [$_{VF}$ Sie] [$_{LK}$ werden/VVFIN] es
[$_{RK}$ vergessen/VVPP haben/VAINF].

(18) [$_{VF}$ Er] [$_{LK}$ sagte/VVFIN], [$_{LK}$ dass/KOUS]
sie es [$_{RK}$ vergessen/VVPP haben/VAFIN].

In **stage 8**, we try to disambiguate ambiguous determiner readings. Here, we eliminate among other things the ART-reading of *die* in (16) as well as the ART-reading in (19), because of the fact that we have no corresponding noun in the relevant part of the right sentential context (MF), we can merge the definite articles with. The results of these eliminations is in both cases an unambiguous PDS-assignment.

(19) Er sagte, [$_{LK}$ dass] [$_{MF}$ **das**/~~ART~~ Geduld]
[$_{RK}$ erfordert].
He said that this needs patience.

In **stage 9**, we consider words with ambiguous preposition readings in more detail. For example, if we have a word with an ambiguous postposition reading (APPO) as in (20), we eliminate this reading if it isn't preceded by a noun phrase that has the same case feature as the postposition. And if we are able to build a prepositional phrase with a word with a preposition reading that also has an adverb (ADV) or ADJD-reading as in (21), we'll do that and eliminate the ambiguous readings.

(20) Er fuhr sie **nach**/~~APPO~~ Saarbrücken.
He drove her to Saarbrücken.

(21) Er wartete **innerhalb**/~~ADJD/~~APPR des Hauses.
He waited inside the house.

In **stage 10**, we investigate sentence-initial words with ambiguous readings. For instance, if we have a numeral word (CARD) as sentence-initial word, we always have a NN-reading because of its capitalization. If such a word is left adjacent to a noun or the preposition *von* followed by another numeral word as in (22), we assign a CARD-tag to it.

(22) Drei/CARD von sieben Zuhörern schliefen.
Three of seven listeners slept.

In **stage 11**, we take a closer look at the words we assigned a FM-tag after the lexicon lookup. For example, if more than half of the tokens inside the quotation marks only have a FM-reading, we assign all other tokens inside this sequence only a FM-tag.

In **stage 12**, we investigate noun ambiguities in more detail. If we detect a rather certain proper NE-context for a word, we delete its NN-reading. For instance, if such a word ends with a common town name suffix (e.g. *stadt* (engl. *town*)), we assign a NE-tag. We also assign a NE-Tag if the word is right adjacent to a preposition that commonly occurs before town names – but only, if this word was marked as unknown after the lexicon lookup.

4.2 Rule-based Tagger Implementation

To implement the above mentioned disambiguation strategies, we make use of deterministic transition networks, in which the arcs of the transitions are labelled with specific functions (see (Klatt, 1997)). These functions enable us to go from every position in the input to any other one. Furthermore, these functions allow us to formulate and check constraints for adjacent and non-adjacent word pairs and its immediate and inner context as well as to do a partial syntactic analysis.

4.3 Hybrid System Configuration

The easiest way to combine two taggers is to use them in sequence. Since our rule-based tagger operates in an easy-first manner, it makes more sense to use it before the statistical tagger. But it is also possible to start with the statistical tagger and use the rule-based one as a kind of error detection component. But therefore, we have to formulate other search strategies, what is a subject for future work. In our experiment, we make use of a simple two-pass architecture starting with the rule-based tagger.

5 Experiment

5.1 Preparing the Experiment

We used a test corpus of 10023 tokens from a small part of the *STZ*-corpus that was hand-tagged by the *The Institute for Natural Language Processing (IMS)* of the University of Stuttgart[2] with the STTS, consisting of 54 tags. After a transformation of the test corpus in a *sentence-per-line*-format, we implemented a conversion mechanism to assign our lexical entries their relevant STTS-tags. In some cases this was not trivial. For instance, in opposite to the STTS we also regard word forms of auxiliary and modal verbs as lexical verbs. This forced us to give different lexical entries the same STTS-tag in a few cases.

[2] Thanks to the IMS for making the corpus available to us.

5.2 Results and Discussion

In table 1, we show the results the *TreeTagger* produced in isolation. Here, the *TreeTagger* makes use of its own lexicon (approx. 700000 words). With the original version of the test corpus, the *TreeTagger* achieved an accuracy rate of 98,79% with its post-processing verb filter stage[3], and 98,69% without it.

After the first evaluation of the rule-based tagger results, we detected eleven wrong non-discussible tag assignments, which we corrected. Tagging the corrected test corpus with the *TreeTagger* resulted in a poorer accuracy rate (see last row of table 1).

Test corpus	Acc (-verb filter)	Acc (+verb filter)
Original	98,69%	98,79%
Corrected	98,61%	98,69%

Table 1: *TreeTagger* in isolation

Table 2 shows the result of our experiment. The first column contains the names of the rule-based tagger stages: LL (lexicon lookup), LD (lexical-driven disambiguation), UW (unknown words), CN (Christian name), S1-S12 for the twelve stages as described before and MC for the manual checking of 44 marked uncertain decisions. The next seven columns show the number of different readings the tokens have as well as whether the correct reading is among them. In the next column, we listed the number of readings considered not to be false (Pos.), followed by the recall and precision rate of the rule-based tagger. In the last two columns, we show the accuracy rate of the *TreeTagger*, using the output of the relevant stages of the rule-based tagger as input. First, without its immanent verb filter, then with it.

Wortwhile to mention before we look at the results in more detail are the following observations: It is astonishing that the *TreeTagger* performs better using our small lexicon instead of its own one. After the stage LD, our corpus consists only of 10001 tokens, since we detected several MWLs. Till stage 6, the recognition of elementary noun phrases, the recall rate doesn't change dramatically. The application of stage 7, the segmentation of the sentence into topological fields, produces a significiant increase in the error rate, but otherwise also significiantly increases the accuracy rate of the hybrid system, but only if we don't use the verb filter of the *TreeTagger*. Therefore it is highly recommendable to make no use of the verb filter from stage 7 on. Also the manual checking of the 44 marked uncertain decisions helped us to decrease the error rate significiantly.

After the lexicon lookup, we had 73 unknown words and 10 words with a wrong tag assignment. We didn't assign the word *nach* and adverb reading, since this word could never be an adverb. But it can be a part of the adverbial MWL *nach wie vor*. Since the STTS isn't able to handle MWLs[4] all parts of a MWL must be indicated in their *own* reading – what is another error source for a statistical tagger. In two cases, we didn't assign the correct FM-reading. In two cases we made a wrong tag conversion and in another case we assigned a NN- instead of a NE-reading. The other four errors are highly discussable. For the two FM-words *of* and *to*, we find in the test corpus a NE- and a APPR-assignment. The word *Inc* was hand-tagged as NE, for what we assume a NN-reading as for other company suffixes.

After the stage LD, we solved the problem of the ADV-reading of *nach* by handling it as part of the above mentioned MWL. But unfortunately, we produced five more errors in this stage. In three cases, we cancelled the PRELS-reading of the word *was* (engl. *what*). But this is no real problem, since we stored these cases for later manual checking because of the uncertainty of this decision. For a certain disambiguation we must have the information, whether the interrogative sentence can function as a verb argument as in (24) or not as in (23) – an information we don't have.

(23) Was/PRELS ich denke, ist disskussionswürdig.
What I think is discussable.

(24) Was/PWS ich denke, fragte er mich.
What I think, he asked me.

In two other cases, we assigned the capitalized parts of a company name a NE-tag instead of a given NN-tag (Flughafen/NN Frankfurt AG)

[3] The task of this filter is to overcome some of the problems mentioned in section 2.1.

[4] This is astonishing for a *theory-neutral* tagset, the STTS is specified. As well as the fact that auxiliary verbs are not considered as lexical verbs as mentioned before.

Rule-based Tagger											+ *TreeTagger*	
Stage	Hits	#0	#1	#2	#3	#4	#5	Pos.	Rec.	Prec.	-vfilter	+vfilter
LL	pos.	-	6427	1844	1423	245	1	15458	99.17%	64.30%	98.80%	98.88 %
	neg.	73	7	-	3	-	-					
+LD	pos.	-	6786	1709	1288	134	-	14693	99.16%	67.49%	98.87%	98.95%
	neg.	71	9	3	1	-	-					
+UW	pos.	-	6800	1764	1291	135	-	14757	99.89%	67.70%	98.96%	99.04%
	neg.	-	7	3	1	-	-					
+CN	pos.	-	6799	1766	1291	135	-	14759	99.90%	67.69%	98.96%	99.04%
	neg.	-	6	3	1	-	-					
+S1	pos.	-	7489	1312	1167	22	-	13716	99.89%	72.83%	98.96%	99.04%
	neg.	-	8	3	-	-	-					
+S2	pos.	-	7610	2092	264	21	-	12689	99.86%	78.71%	98.99%	99.07%
	neg.	-	10	3	1	-	-					
+S3	pos.	-	7720	2015	242	10	-	12535	99.86%	79.67%	99.09%	99.12%
	neg.	-	10	3	1	-	-					
+S4	pos.	-	7731	2005	241	10	-	12523	99.86%	79.75%	99.09%	99.12%
	neg.	-	10	3	1	-	-					
+S5	pos.	-	8043	1791	148	5	-	12108	99.86%	82.48%	99.15%	99.15%
	neg.	-	10	3	1	-	-					
+S6	pos.	-	9225	634	122	4	-	10896	99.84%	91.64%	99.14%	99.14%
	neg.	-	12	3	1	-	-					
+S7	pos.	-	9519	401	31	4	-	10479	99.54%	95.00%	99.26%	99.15%
	neg.	-	43	3	-	-	-					
+S8	pos.	-	9608	318	25	4	-	10383	99.54%	95.88%	99.32%	99.21%
	neg.	-	43	3	-	-	-					
+S9	pos.	-	9663	272	15	4	-	10317	99.53%	96.48%	99.32%	99.21%
	neg.	-	45	2	-	-	-					
+S10	pos.	-	9694	245	12	3	-	10281	99.53%	96.82%	99.34%	99.23%
	neg.	-	45	2	-	-	-					
+S11	pos.	-	9696	245	10	3	-	10277	99.53%	96.86%	99.36%	99.25%
	neg.	-	45	2	-	-	-					
+S12	pos.	-	9742	196	10	3	-	10228	99.50%	97.29%	99.36%	99.25%
	neg.	-	48	2	-	-	-					
+MC	pos.	-	9760	194	10	2	-	10222	99.65%	97.50%	99.51%	99.40%
	neg.	-	34	1	-	-	-					

Table 2: Rule-based tagger in combination with the *TreeTagger*

and ADJA-tag (Deutschen/ADJA Lufthansa AG). Maybe, it would be better to store them as MWL units to avoid confusion.

In the stage UW, we were able to assign all unknown words their correct tag. Furthermore, we add to two previously NN-tagged words (*Rate* and *Primus*) their correct FM-tag since they occur in contexts surrounded by a quotation mark that contain other FM-words (*,,Prime Rate"* and *,,Primus inter pares"*).

After the stage CN, we had the highest recall rate, since we additionally assigned one word its correct NE-tag, which was tagged before as NN (*Stolzenburg*).

The next six errors till stage 6 were produced by some constellations we didn't take into account and some tokenizing problems. In one case, we wrongly tagged a headline as a sentence.

In stage 7, we had the most dramatic error increase. From the new 30 errors, 15 were produced by a wrong TFM segmentation, three occured because of ungrammatical sentences. The other twelve errors were made by a wrong VVPP-assignment instead of an ADJD-assignment, of which we stored eleven of them as uncertain decisions for later manual checking.

In stage 9, we tagged *ausschließlich* (engl. *solely*) as a preposition instead of an adverb, since it occurs in a typical prepositional context.

In stage 12, we made in three cases a wrong noun distinction. In two cases we assigned a wrong NN-tag, in one case a wrong NE-tag.

If we look at the remaining ambiguities after stage 12, we find in most of the cases NN-NE-ambiguities (69x), followed by ADJD-ADV-ambiguities (20x) and NN-ADJA-ambiguities (18x). Some of the NN-NE-ambiguities could be resolved by enlarging our lexicon by communication verbs like *sagen* (engl. *to say*), which often occur adjacent to proper nouns.

If the information of article borders would be available to us, the application of Yarowskys *one sense per discourse*-constraint (Yarowsky, 1995) would be another worthwile strategy. That is, if we are able to disambiguate one NE-reading in a certain context, we can be sure that all other occurences of this word would have the same reading in the article.

Finally, the comparison of our tagger output with the output of other taggers could also lead to a higher accuracy rate, if the taggers has produced different results. But instead of trying to find the correct tag automatically as done in (Zavrel and Daelemans, 2000), we prefer at the moment a manual check of the different tag assignments - but what is of course only applicable to small corpora.

6 Summary

We presented an unsupervised rule-based tagger for German that can be used in isolation or in combination with a statistical tagger to produce highly reliable accuracy results. It also leaves the black box mentality of statistical taggers and marks specific uncertain tag assignments for later manual checking. Unknown words in particular contexts can be extracted to extend the lexicon. And finally, our tagger helps to improve supposedly correct handtagged corpora.

References

T. Brants. 2000. TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP conference, ANLP-2000. Seattle, WA.*

E. Brill. 1994. Some advances in transformation-based part of speech tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence. Volume 1*, Menlo Park, CA, USA, July31 August–4 . AAAI Press.

S. Klatt. 1997. Pattern-matching Easy-first Planning. In A. Drewery, G. Kruijff, and R. Zuber, editors, *The Proceedings of the Second ESSLLI Student Session*, Aix-en-Provence. 9th European Summer School in Logic, Language and Information.

W. Lezius, R. Rapp, and M. Wettler. 1996. A Morphology-System and Part-of-Speech Tagger for German. In D. Gibbon, editor, *Natural Language Processing and Speech Technology. Results of the 3rd KONVENS Conference.* Mouton de Gruyter.

J. Rehbein. 1992. Zur Wortstellung in komplexen deutschen Sätzen. In L. Hoffmann, editor, *Deutsche Syntax: Ansichten und Aussichten*, Institut für deutsche Sprache (Jahrbuch 1991). Walter de Gruyter, Berlin, New York.

C. Samuelsson and A. Voutilainen. 1997. Comparing a linguistic and a stochastic tagger. In *Proceedings of the Joint 35th Annual Meeting of the Association for Computational Linguistics.*

A. Schiller, C. Stöckert S. Teufel, and C. Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universität Stuttgart, Universität Tübingen.

H. Schmid. 1999. Improvements in part-of-speech tagging with an application to german. In S. Armstrong, K.W. Church, P. Isabelle, S. Manzi, E. Tzoukermann, and D. Yarowsky, editors, *Natural Language Processing Using Very Large Corpora.* Kluwer, Dordrecht.

P. Tapainainen. 1996. The Constraint Grammar Parser CG-2. Technical report, Dept. General Linguistics, University of Helsinki.

M. Volk and G. Schneider. 1998. Comparing a statistical and a rule-based tagger for german. In *Proceedings of KONVENS-98.*

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics.*

J. Zavrel and W. Daelemans. 2000. Bootstrapping a tagged corpus through combination of existing heterogeneous taggers. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000).*