

Sparse Data and Smoothing in Statistical Part-of-Speech Tagging

Joakim Nivre
Göteborg University
nivre@ling.gu.se

Abstract

This article reports on a series of experiments in statistical part-of-speech tagging of Swedish texts, with different probabilistic models and different smoothing schemes for both lexical and contextual probabilities. The most important conclusions are that lexical and contextual probabilities require different smoothing methods and that smoothing is only crucial for lexical probabilities. Of the particular smoothing methods tested in the experiments, Good-Turing estimation achieves the best results for the lexical model, while a simple additive smoothing scheme gives the best performance for the contextual model.

1 Introduction

Given that part-of-speech tagging is one of the most well-studied applications of statistical natural language processing and that sparse data is one of the most serious problems in any application of statistical natural language processing, it is somewhat surprising to find that the topic of sparse data in statistical part-of-speech tagging has hardly been treated at all in the literature. I think there are two explanations for this state of affairs. First, the problem of sparse data and its various remedies treated under the rubric of ‘smoothing’ have been studied extensively within the neighboring domain of language modeling, and many people have assumed that the results carry over directly to the domain of part-of-speech tagging. Secondly, the problem of sparse data in part-of-speech tagging mainly surfaces in the guise of the ‘unknown word problem’, which is most successfully handled by non-statistical methods even within the framework of a statistical tagger.

Nevertheless, I think there are important lessons to be learned if the problem of sparse data and smoothing in statistical part-of-speech tagging is studied in its own right. First, I do not think that the results from language modeling can be applied unproblematically within the context of part-of-speech tagging. Secondly, I think statistical methods have been underestimated for dealing with unknown words, even though it is clear that they need to be supplemented with

other means in order to achieve peak performance. Thirdly, I believe that a systematic evaluation of the methods inherited from language modeling within the context of part-of-speech tagging may increase our general understanding of the relationships between different statistical models of language. At least, that is my hope.

This article reports on a series of experiments in statistical part-of-speech tagging of Swedish texts, with different probabilistic models and with different smoothing schemes for both lexical and contextual probabilities. The aim of these experiments has not been to produce a maximally accurate part-of-speech tagger for Swedish texts, but rather to study the effect of different models and different smoothing methods in a systematic fashion, in order to gain a better theoretical understanding of the strength and weaknesses of different methods. Hopefully, this will eventually lead to better results also on the practical side.

2 Background

2.1 Statistical Part-of-Speech Tagging

Part-of-speech tagging refers to the problem of assigning lexical categories, or parts-of-speech, to words in a text, a problem for which there now exist a variety of methods. Sometimes, these methods are divided into two groups: statistical vs. rule-based. However, this terminology is a bit misleading since many ‘rule-based’ methods rely on statistics in the training phase (e. g., Brill 1995) and some of the ‘non-statistic’ methods are not really rule-based in the traditional sense (e. g., Daelemans *et al* 1996). In this article, I will use the term *statistical part-of-speech tagging* in a narrow sense, referring only to tagging methods that use a probabilistic model during the actual tagging phase and try to find the most probable part-of-speech sequence for a particular string of words.

Most statistical taggers are based on some variant of the n -class model (cf. Merialdo 1994), which can be seen as an instance of Shannon’s noisy channel model based on Bayesian inversion:

$$P(c_1, \dots, c_k | w_1, \dots, w_k) = \frac{P(w_1, \dots, w_k | c_1, \dots, c_k) P(c_1, \dots, c_k)}{P(w_1, \dots, w_k)}$$

In order to find the maximally probable part-of-speech sequence c_1, \dots, c_k for a given string of words w_1, \dots, w_k , we only need to find that sequence which maximizes the product in the numerator of the right hand side (since the denominator is constant for a given word string). The first factor of this product is given by the *lexical model*:

$$\hat{P}(w_1, \dots, w_k | c_1, \dots, c_k) = \prod_{i=1}^k P(w_i | c_i)$$

In this model, every word is conditioned only on its own part-of-speech, an independence assumption which may seem unrealistic but which is necessary in

order to get a tractable and trainable model. Some early systems (e. g., DeRose 1988) instead use the inverse probabilities, i. e., $P(c_i|w_i)$, which may be easier to estimate intuitively but which are not warranted by the noisy channel model and which appear to give worse performance (Charniak *et al* 1993).

The second factor is estimated by means of the *contextual model*:

$$\hat{P}(c_1, \dots, c_k) = \prod_{i=1}^k P(c_i | c_{i-(n-1)}, \dots, c_{i-1})$$

In this model, every part-of-speech is conditioned on the $n - 1$ previous parts of speech. Depending on the value of n , we get different varieties of the n -class model, known as uniclass, biclass, triclass, etc. The two most common values of n are 2 and 3, and in this article we will restrict ourselves mainly to the biclass and triclass models.

The n -class model can be implemented very efficiently as a Hidden Markov Model (HMM), where the contextual model is defined by the transition probabilities of the underlying Markov chain, while the lexical model is defined by the output probabilities. The task of finding the most probable part-of-speech sequence for a given string of words is then equivalent to finding the optimal path (state sequence) of the model for a particular output string, a problem which can be solved with reasonable efficiency using the Viterbi algorithm (Viterbi 1967).

Given enough training data, statistical taggers based on the n -class model typically achieve accuracy rates ranging from 95% (Charniak *et al* 1993) to 97% (Merialdo 1994), depending on the type of text and the tagset used. Although most of the studies still concern English text, there are now a fair amount of studies reporting similar results for other languages, such as French (Chanod and Tapanainen 1995) and Swedish (Brants and Samuelsson 1995). It should be noted, however, that most if not all statistical taggers that are used in practice are hybrid systems in the sense that they contain non-statistical components for handling such problems as unknown words.

2.2 Parameter Estimation

The major problem in constructing a statistical tagger — or any other probabilistic model for that matter — is to find good estimates for the model parameters. In the n -class model, there are two types of parameters that need to be estimated:

1. Lexical probabilities: $P(w|c)$
2. Contextual probabilities: $P(c_i | c_{i-(n-1)}, \dots, c_{i-1})$

There are basically two methods that are used to estimate these parameters empirically from corpus data, depending on what kind of data is available for training. Both methods are based on the notion of Maximum Likelihood Estimation (MLE), which means that we try to choose those estimates that maximize the probability of the observed training data. If we have access to tagged

training data, we can use relative frequencies to estimate probabilities:¹

$$\hat{P}(w|c) = \frac{f_N(w, c)}{f_N(c)}$$

$$\hat{P}(c_i|c_{i-(n-1)}, \dots, c_{i-1}) = \frac{f_N(c_{i-(n-1)}, \dots, c_i)}{f_N(c_{i-(n-1)}, \dots, c_{i-1})}$$

If we only have access to untagged data, the standard method is to start from some initial model and use the Baum-Welch algorithm for Hidden Markov Models (Baum 1972) to iteratively improve the estimates until we reach a local maximum.² Unfortunately, there is no guarantee that we ever reach a *global* maximum, and results are generally better if we can use tagged data for estimation (Merialdo 1994).

Regardless of which method we use to obtain a maximum likelihood estimation from our training data, we still have to face the ubiquitous problem of *sparse data*, which means that, for a lot of the events whose probability we want to estimate, we simply do not have enough data to get a reliable estimate. The most drastic case of this is events that do not occur at all in the training data, such as ‘unknown words’ in the context of part-of-speech tagging. If we assign these events zero probability (according to MLE), then any chain of independent events involving such an event will also be assigned probability zero, which is usually not very practical (unless we can be sure that the event in question is really impossible and not just infrequent). Therefore, we normally want to adjust our estimates in such a way that we can reserve some of the probability mass for events that we have not yet seen. This is what is known in the business as *smoothing*.

2.3 Smoothing

Before we turn to the various methods used for smoothing, let us note that the problem of sparse data affects the two models involved in statistical part-of-speech tagging rather differently. In the contextual model, we always know how many events we haven’t seen. For example, given a part-of-speech system with N_C tags, we know that there are N_C^n possible n -tuples. By contrast, the lexical model is open-ended, and it is usually very difficult to estimate how many words (or word-tag pairs) we haven’t seen — unless we use a lexicon to stipulatively limit the class of words allowable in texts, a move which is often made when evaluating taggers, but which is usually completely unrealistic from a practical application point of view. We will return to the problem of the open-ended lexical model in section 3.

Most work on smoothing of probabilistic models for natural language processing has been carried out within the context of language modeling, i. e., the task

¹The relative frequency $f_N(E)$ of an event E in a sample of N observations is always a maximum likelihood estimate of the probability $P(E)$; see, e. g., Lindgren (1993).

²The Baum-Welch algorithm can be seen as a special case of the general technique known as Expectation-Maximization (EM); cf. Dempster *et al* (1977).

of assigning probabilities to strings of words, which is a crucial problem in statistical approaches to speech recognition (cf. Jelinek 1997, Ney *et al* 1997). As stated in the introduction, I regard it as an open question to what extent the results from language modeling carry over directly to part-of-speech tagging, and it is part of the purpose of this article to throw some light upon this issue.

The methods used for smoothing can be divided into two broad categories. In the first category, which we may call *smoothing proper*, we find methods where the parameters of a single model are being adjusted to counter the effect of sparse data, usually by taking some probability mass from seen events and reserving it for unseen events. This category includes methods such as *additive smoothing* (Lidstone 1920, Gale and Church 1990), *Good-Turing estimation* (Good 1953, Gale and Sampson 1995), and various methods based on held-out data and cross-validation (Jelinek and Mercer 1985, Jelinek 1997).

In the second category, which we may call *combinatory smoothing*, we find methods for combining the estimates from several models. The most well-known methods in this category are probably *back-off smoothing* (Katz 1987) and *linear interpolation* (Brown *et al* 1992). In the following, I will restrict the discussion to those methods that will appear in the experiments later on.

2.3.1 Additive Smoothing

Perhaps the simplest of all smoothing methods is what is known as *additive smoothing*, and which consists in adding a constant k to all the frequencies (including the zero frequencies of unseen events) and then making a new maximum likelihood estimation. In other words, for each value x of a variable X , where x has the observed (absolute) frequency $f(x)$ in a sample of N observations, and X has a sample space of N_X possible values, we give the following estimate of $P(x)$:³

$$\hat{P}(x) = \frac{f(x) + k}{N + kN_X}$$

Depending on the value of k , this method has different names. For $k = 1$ it is known as Laplace’s Law; for $k = 0.5$ it is known as Lidstone’s Law or Expected Likelihood Estimation (ELE). The results from language modeling seem to indicate that additive smoothing is not a very good method, usually overestimating the probability of unseen events (Gale and Church 1994).

2.3.2 Good-Turing Estimation

Good-Turing estimation is a more sophisticated smoothing method, which uses expected frequencies of frequencies to reestimate the raw sample frequencies. More precisely, for any outcome with (absolute) frequency $f(x)$, we base our probability estimates on the reestimated frequency $f^*(x)$ derived by the following formula:

$$f^*(x) = (f + 1) \frac{E(N_{f(x)+1})}{E(N_{f(x)})}$$

³In these and following formulas, $P(x)$ is shorthand for $P(X = x)$ in the usual way.

where N_f is the number of outcomes with frequency f and $E(X)$ is the expectation value of the variable X . In practice, there is no way of precisely calculating expected frequencies of frequencies, and different versions of Good-Turing estimation differ mainly in the way they estimate these values from the observed frequencies of frequencies (see, e. g., Good 1953, Church and Gale 1991, Gale and Sampson 1995).

2.3.3 Back-off Smoothing

The basic idea in back-off smoothing is to use the basic MLE model for events which are frequent enough in the training data to have reliable estimates and to back off to a more general model for rare events, i. e., back off to a model where distinct outcomes in the first model are lumped together according to some equivalence relation. However, in order to get a correct probabilistic model, we must introduce a discounting factor for the first model probabilities (in order to reserve some probability mass for unseen or underestimated events) and a normalizing factor for the back-off probabilities (in order to make them comparable to the first model probabilities). For example, in language modeling, it is common practice to back off from a trigram to a bigram model (and from a bigram model to a unigram model if necessary):

$$\hat{P}(w_i|w_{i-2}, w_{i-1}) = \begin{cases} (1 - \delta_{f(w_{i-2}, w_{i-1}, w_i)}) \frac{f(w_{i-2}, w_{i-1}, w_i)}{f(w_{i-2}, w_{i-1})} & \text{if } f(w_{i-2}, w_{i-1}, w_i) > f' \\ \alpha_{f(w_{i-2}, w_{i-1})} \hat{P}(w_i|w_{i-1}) & \text{otherwise} \end{cases}$$

In this equation, f' is the frequency threshold above which we keep the estimates of the original model, $\delta_{f(w_{i-2}, w_{i-1}, w_i)}$ is the discounting factor for the frequency $f(w_{i-2}, w_{i-1}, w_i)$, and $\alpha_{f(w_{i-2}, w_{i-1})}$ is the normalization factor for the context w_{i-2}, w_{i-1} . Different ways of determining these factors give different versions of back-off smoothing. One common method is to use some version of Good-Turing estimation for this task (see, e. g., Katz 1987).

In part-of-speech tagging, back-off smoothing can be used in the contextual model, which is structurally isomorphic to the n -gram model in language modeling. Thus, triclass probabilities may be smoothed as follows:

$$\hat{P}(c_i|c_{i-2}, c_{i-1}) = \begin{cases} (1 - \delta_{f(c_{i-2}, c_{i-1}, c_i)}) \frac{f(c_{i-2}, c_{i-1}, c_i)}{f(c_{i-2}, c_{i-1})} & \text{if } f(c_{i-2}, c_{i-1}, c_i) > f' \\ \alpha_{f(c_{i-2}, c_{i-1})} \hat{P}(c_i|c_{i-1}) & \text{otherwise} \end{cases}$$

For the lexical model, however, there is generally no sensible model to back off to. It is true that we could use the plain word probability $P(w)$ to estimate the lexical probability $P(w|c)$, but the problem is that in cases where w is an unknown word (or even a very rare one), estimates of the former probability are usually no better than estimates of the latter.

3 Method

3.1 Experimental Variables

In order to get as rich a picture as possible of the influence of different factors in statistical part-of-speech tagging, the following three variables have been varied systematically:

1. Tagging model
2. Lexical smoothing method
3. Contextual smoothing method

In addition, the experiments have been carried out with two different tagsets (cf. section 3.2.1).

3.1.1 Tagging Model

Two different versions of the n -class model have been used, the biclass ($n = 2$) and the triclass ($n = 3$) models:

$$\hat{P}_{Bi}(c_1, \dots, c_k, w_1, \dots, w_k) = \prod_{i=1}^k P(w_i|c_i)P(c_i|c_{i-1})$$

$$\hat{P}_{Tri}(c_1, \dots, c_k, w_1, \dots, w_k) = \prod_{i=1}^k P(w_i|c_i)P(c_i|c_{i-2}, c_{i-1})$$

In table headings, the abbreviations **Mod**, **Bi** and **Tri** will be used for ‘tagging model’, ‘biclass model’ and ‘triclass model’, respectively.

3.1.2 Lexical Smoothing

Two different methods for lexical smoothing have been used, the first being additive smoothing with $k = 0.5$:

$$\hat{P}_{Add}(w|c) = \frac{f(w, c) + 0.5}{f(c) + 0.5 + \sum_{w' \in W: f(w', c) > 0} 0.5}$$

In this equation, W refers to the set of all known words, i. e. the set of words occurring in the training data. This means that all unknown words will be treated as tokens of a single unknown word type w_U , which will be assigned the following lexical probability for a given part-of-speech c :

$$\hat{P}_{Add}(w_U|c) = \frac{0.5}{f(c) + 0.5 + \sum_{w \in W: f(w, c) > 0} 0.5}$$

This may seem like a radical way of dealing with the problem of knowing how many unseen words there are (cf. section 2.3), but it works well in practice.

The second method used for lexical smoothing is Good-Turing estimation:

$$\hat{P}_{GT}(w|c) = \frac{f_c^*(w, c)}{f(c)}$$

The reestimated frequencies have been calculated using the simple Good-Turing method (Gale and Sampson 1995) in Dan Melamed’s implementation.⁴ The frequency distribution has been reestimated separately for each part-of-speech, and the unknown word w_U is assigned the following lexical probability for a given part-of-speech c :

$$\hat{P}_{GT}(w_U|c) = \frac{1 - \sum_{w \in W} f_c^*(w, c)}{f(c)}$$

In table headings, the abbreviations **Lex**, **Add** and **GT** will be used to stand for ‘lexical smoothing method’, ‘additive smoothing’ and ‘Good-Turing estimation’, respectively.

3.1.3 Contextual Smoothing

For the contextual model, three different schemes have been used:

1. MLE (i. e., no smoothing at all)
2. Additive smoothing
3. Good-Turing estimation
4. Back-off smoothing

In Good-Turing estimation, the biclass and triclass distributions have been smoothed separately, but no further partitioning of these distributions according to parts-of-speech has been made. In fact, for the small tagset (cf. section 3.2.1), the biclass distribution required no reestimation at all (i. e., $f_{Bi}^*(c_1, c_2) = f(c_1, c_2)$ for all frequencies). In the back-off smoothing, the same Good-Turing estimation was used to determine the discounting and normalization factors (cf. section 2.3.3).

The following abbreviations will be used in table headings: **Con** = ‘contextual smoothing method’, **MLE** = ‘maximum likelihood estimation’, **Add** = ‘additive smoothing’, **GT** = ‘Good-Turing estimation’, **BO** = ‘back-off smoothing’.

⁴Available at: ftp://ftp.cis.upenn.edu/pub/melamed/tools/Good-Turing_smoothing/.

3.2 Data and Tagsets

The data used for the experiments come from the Stockholm-Umeå Corpus (Ejerhed *et al* 1992), a balanced corpus of written Swedish containing 1.2 million words. The corpus has been automatically tagged for parts-of-speech and manually corrected but is known to contain a small percentage of errors. Of this corpus, 99% was used for training and 1% for testing.

3.2.1 Tagsets

The tagset used to tag the Stockholm-Umeå Corpus consists of 23 basic parts-of-speech with morpho-syntactic features that bring the total number of distinct tags to 156. The experiments were run both with the small tagset, consisting only of the 23 basic parts-of-speech with no features, and with the large tagset containing all 156 tags. The two tagsets are listed in appendix A and B.

3.2.2 Training Data

After 11148 words had been sampled as test data (cf. section 3.2.3), the remaining corpus of 1,155,753 tokens (punctuation included) was used as training data to derive maximum likelihood estimates for lexical and contextual probabilities. The total number of word types in the training corpus was 96,672. No normalization of upper- and lowercase letters was performed, which means that words that differed only in terms of capitalization were treated as different word types using training.

3.2.3 Test Data

The test data consist of ten blocks of 1115 tokens each,⁵ randomly drawn from the entire corpus. These tokens include punctuation as well as ordinary words. In evaluating the performance of different methods (cf. section 3.5), the original tags were always assumed to be correct, even though the corpus is known to contain some errors.

Table 1 shows the proportion of unknown words (UW), known words with unknown tags (UT), and ambiguous words (AW) (i. e., words with more than one part-of-speech in the training corpus) in the different blocks for the small tagset. Table 2 gives the same information for the large tagset.

3.3 Tagger

The tagger used for the experiments was a standard HMM tagger using the Viterbi algorithm to derive the most probable part-of-speech sequence for a given string of words, and using the original tokenization of the Stockholm-Umeå Corpus (which was used also in training the tagger). The condition was one of forced choice, i. e., the tagger was forced to assign exactly one tag to every word.

⁵In fact, the tenth block only contains 1113 tokens.

Table 1: Test data statistics (small tagset)

Block	UW	UT	AW	Total
1	156	6	411	1115
2	122	4	401	1115
3	100	3	444	1115
4	88	1	446	1115
5	90	2	425	1115
6	88	2	420	1115
7	49	6	468	1115
8	47	2	467	1115
9	23	1	473	1115
10	49	2	472	1113
Total	812	29	4427	11148

Table 2: Test data statistics (large tagset)

Block	UW	UT	AW	Total
1	156	8	454	1115
2	122	12	454	1115
3	100	7	520	1115
4	88	5	505	1115
5	90	4	504	1115
6	88	8	492	1115
7	49	10	550	1115
8	47	7	513	1115
9	23	4	529	1115
10	49	4	530	1113
Total	812	69	5051	11148

In order to avoid underflow problems,⁶ input strings were segmented into minimal strings with an unambiguous suffix of two words, i. e., where the last two words have only one possible part-of-speech each. By supplying these parts-of-speech as context for the next minimal string, we guarantee that no information is lost, since dependencies spanning more than three words are beyond the scope of the triclass model.

3.4 Unknown Words

As indicated earlier, most statistical taggers use non-statistical rules and heuristics in dealing with unknown words in order to improve performance. However, in this study, we are not interested in performance *per se*, but in the relative performance of different smoothing methods. Therefore, no attempt has been made to optimize the treatment of unknown words, which are simply treated as tokens of the single unknown word type w_U (cf. section 3.1.2). The only exception concerns capitalized words, which are handled by the following algorithm during testing:

- Let w_{Cap} be a capitalized word form occurring in the test data and let w_{LC} be that form which differs from w_{Cap} only in being all lowercase.
 1. If w_{Cap} occurs in the training data, the lexical probabilities $P(w_{Cap}|c_i)$ are used (for all c_i used to tag w_{Cap} in the training corpus).
 2. If w_{Cap} does not occur in the training but w_{LC} does, the lexical probabilities $P(w_{LC}|c_i)$ are used (for all c_i used to tag w_{LC} in the training corpus).
 3. Otherwise the probabilities $P(w_U|c_i)$ are used (and the fact that w_{Cap} is capitalized is not taken into account in tagging).

Another issue that needs to be decided in the treatment of unknown words is which parts-of-speech should be considered *open*, i. e., should be treated as possible tags for unknown words. Rather than rely on intuitive judgements, it was decided to use statistical criteria to determine which parts-of-speech to treat as open:

A part-of-speech c is *open* iff (i) c has at least x tokens in the training corpus, and (ii) the reestimated frequency of zero frequency items in c (according to Good-Turing estimation) is at least y .

Different values for x and y were tested, but in the final experiments x was set to 100 and y to 1/1000.

Comparing the set of open classes thus defined to the parts-of-speech actually represented by unknown words in the test corpus showed that 9 out of 10

⁶Multiplying long chains of probabilities may yield numbers that are so small they are effectively rounded off to zero; this is known in the literature as the underflow problem (see, e. g., Cutting *et al* 1992).

(small tagset) and 40 out of 48 (large tagset) of the parts-of-speech that actually occurred in the test data were defined as open according to the definition above. Conversely, 9 out of 10 (small tagset) and 40 out of 42 (large tagset) of the classes defined as open actually occurred with unknown words in the test data. However, if we look at the number of word tokens in different classes, we find that the classes missed were actually very rare, and that 99.8% (810/812) for the small tagset and 98.0/(795/812) for the large tagset of the unknown word tokens in fact had a part-of-speech that was defined as open according to the given criteria, which means that it was at least possible for the tagger to find the correct tag. This seems to indicate that the definition chosen was at least reasonable for the corpus and tagsets used in this study.⁷

Finally, it should be pointed out that the notion of open parts-of-speech was only used for unknown words, i. e., word types that had not occurred in the training data. By contrast, known words were only ever allowed to have parts-of-speech with which they had been seen in the training corpus, which means that some words in the test data could never be tagged correctly, since they were known words with new parts-of-speech (cf. Tables 1 and 2).

3.5 Evaluation

The standard way of evaluating part-of-speech taggers (assuming a condition of forced choice) is by computing its accuracy rate, i. e., the proportion of correct tags out of the total number of tagged tokens. In the present study, accuracy rates were calculated for four different categories of words:

1. Total (T) = All tokens
2. Known (K) = Known words (incl. known words with unknown tags)
3. Ambiguous (A) = Ambiguous words
4. Unknown (U) = Unknown words

Significance was determined using a paired *t*-test on the number of errors per text block ($df = 9$). Significant differences will be indicated by * (.95 level) and ** (.99 level). The abbreviations **Acc** and **Sign** in table headings stand for ‘accuracy rate’ and ‘significance’ respectively.

4 Results

4.1 Tagging Model

Table 3 shows a paired comparison of the biclass and triclass model under different experimental conditions with the small tagset. As can be seen from the last column, there are very few significant differences between the two models, although the triclass model seems to get slightly higher accuracy rates for the

⁷The open parts-of-speech are marked with an asterisk in appendix A and B.

given test data, the top result being 94.82% overall, with 96.92% for known words, 92.84% for ambiguous words, and 69.45% for unknown words. These results were obtained with Good-Turing lexical smoothing and additive contextual smoothing. Table 4 gives the results for the large tagset. As expected, variation is greater and accuracy rates lower than for the small tagset, and the number of statistically significant differences is slightly larger than for the small tagset. In this case, all the significant comparisons except one favor the biclass model, although the single best result is still obtained with the triclass model, Good-Turing lexical smoothing and additive contextual smoothing (T = 91.45%, K = 95.52%, A = 90.94%, U = 39.66%). The most striking difference with respect to the small tagset is the dramatic drop in performance for unknown words, from close to 70% to less than 40% correct.

4.2 Lexical Smoothing

The comparison of Good-Turing and additive lexical smoothing is presented in Table 5 (small tagset) and Table 6 (large tagset). The pattern emerging jointly from these two tables is very clear. First of all, Good-Turing is significantly better than additive smoothing for unknown words under all conditions. With the small tagset, the difference is dramatic, resulting in an error reduction of 33% on average. With the large tagset, the difference is not as large numerically but nevertheless very significant. Secondly, Good-Turing is significantly better than additive smoothing overall (except in two cases), but this difference can be attributed in its entirety to the difference for unknown words. For known and ambiguous words, there are hardly any differences at all.

4.3 Contextual Smoothing

The effects of the four different contextual smoothing schemes (MLE, Add, GT, BO) together with the small tagset are shown in Table 7. Perhaps the most striking result here is the total lack of difference for the biclass model, which seems to indicate that with a small tagset and one million words of tagged data for training, no smoothing at all is required for the contextual part of the biclass model. If we turn to the triclass model, we begin to find some small differences, especially together with the better lexical model (Good-Turing), where additive smoothing gives significantly better results than the other methods (although the numerical difference is small). It is also worth noticing that the pure MLE model still works as well as back-off smoothing and actually better than Good-Turing estimation.

The results for the large tagset can be found in Table 8. Although we find more significant differences here, they are still rather small for the biclass model, and the pure MLE model still gives a reasonable performance although it is now consistently the worst contextual model. It is only when we consider the triclass model that the MLE model begins to drop noticeably, followed by the GT model. Under this condition, additive smoothing and back-off smoothing give considerably better results than the other models.

Table 3: Tagging model (small tagset)

Lex	Con	Acc	Bi	Tri	Sign
Add	MLE	T	93.14	93.29	
		K	96.57	96.80	
		A	92.03	92.57	
		U	51.59	50.76	
	Add	T	93.14	93.29	
		K	96.57	96.80	
		A	92.03	92.55	
		U	51.59	50.88	
	GT	T	93.14	93.13	
		K	96.57	96.74	
		A	92.03	92.41	
		U	51.59	49.47	
BO	T	93.14	93.32		
	K	96.57	96.83	*	
	A	92.03	92.64		
	U	51.59	50.76		
GT	MLE	T	94.38	94.68	*
		K	96.61	96.84	*
		A	92.12	92.66	
		U	67.45	68.51	
	Add	T	94.38	94.82	**
		K	96.61	96.92	*
		A	92.12	92.84	
		U	67.45	69.45	*
	GT	T	94.38	94.37	
		K	96.61	96.69	
		A	92.12	92.30	
		U	67.45	66.27	
BO	T	94.38	94.71	*	
	K	96.61	96.87	*	
	A	92.12	92.73		
	U	67.45	68.51		

Table 4: Tagging model (large tagset)

Lex	Con	Acc	Bi	Tri	Sign
Add	MLE	T	90.37	89.27	**
		K	95.08	93.83	**
		A	90.04	87.52	
		U	30.54	31.28	
	Add	T	90.52	90.13	
		K	95.22	94.67	*
		A	90.33	89.22	
		U	30.67	32.39	
	GT	T	90.46	89.76	*
		K	95.15	94.67	*
		A	90.20	89.22	
		U	30.67	27.34	
	BO	T	90.47	90.93	**
		K	95.17	95.46	
		A	90.24	90.82	
		U	30.67	33.25	
GT	MLE	T	91.20	89.11	**
		K	95.32	93.21	**
		A	90.53	86.26	
		U	38.79	36.95	
	Add	T	91.40	91.45	
		K	95.51	95.52	
		A	90.92	90.94	
		U	39.04	39.66	
	GT	T	91.33	89.53	**
		K	95.45	94.15	**
		A	90.80	88.16	
		U	38.91	30.79	**
	BO	T	91.31	91.46	
		K	95.42	95.66	
		A	90.75	91.22	
		U	38.92	38.05	

Table 5: Lexical smoothing (small tagset)

Mod	Con	Acc	Add	GT	Sign
Bi	MLE	T	93.14	94.38	**
		K	96.57	96.61	
		A	92.03	92.12	
		U	51.59	67.45	**
	Add	T	93.14	94.38	**
		K	96.57	96.61	
		A	92.03	92.12	
		U	51.59	67.45	**
	GT	T	93.14	94.38	**
		K	96.57	96.61	
		A	92.03	92.12	
		U	51.59	67.45	**
	BO	T	93.14	94.38	**
		K	96.57	96.61	
		A	92.03	92.12	
		U	51.59	67.45	**
Tri	MLE	T	93.29	94.68	**
		K	96.80	96.84	
		A	92.57	92.66	
		U	50.76	68.51	**
	Add	T	93.29	94.82	**
		K	96.80	96.92	
		A	92.55	92.84	
		U	50.88	69.45	**
	GT	T	93.13	94.37	**
		K	96.74	96.69	
		A	92.41	92.30	
		U	49.47	66.27	**
	BO	T	93.32	94.71	**
		K	96.83	96.87	
		A	92.64	92.73	
		U	50.76	68.51	**

Table 6: Lexical smoothing (large tagset)

Mod	Con	Acc	Add	GT	Sign
Bi	MLE	T	90.37	91.20	**
		K	95.08	95.32	
		A	90.04	90.53	
		U	30.54	38.79	**
	Add	T	90.52	91.40	**
		K	95.22	95.51	
		A	90.33	90.92	
		U	30.67	39.04	**
	GT	T	90.46	91.33	**
		K	95.15	95.45	
		A	90.20	90.80	
		U	30.67	38.91	**
	BO	T	90.47	91.31	**
		K	95.17	95.42	
		A	90.24	90.75	
		U	30.67	38.92	**
Tri	MLE	T	89.27	89.11	
		K	93.83	93.21	**
		A	87.52	86.26	
		U	31.28	36.95	**
	Add	T	90.13	91.45	**
		K	94.67	95.52	**
		A	89.22	90.94	
		U	32.39	39.66	**
	GT	T	89.76	89.53	
		K	94.67	94.15	*
		A	89.22	88.16	
		U	27.34	30.79	**
	BO	T	90.93	91.46	**
		K	95.46	95.66	
		A	90.82	91.22	
		U	33.25	38.05	**

Table 7: Contextual smoothing (small tagset)

Mod	Lex	Acc	MLE	Add	GT	BO	Sign
Bi	Add	T	93.14	93.14	93.14	93.14	
		K	96.57	96.57	96.57	96.57	
		A	92.03	92.03	92.03	92.03	
		U	51.59	51.59	51.59	51.59	
	GT	T	94.38	94.38	94.38	94.38	
		K	96.61	96.61	96.61	96.61	
		A	92.12	92.12	92.12	92.12	
		U	67.45	67.45	67.45	67.45	
Tri	Add	T	93.29	93.29	93.13	93.32	^a *
		K	96.80	96.80	96.74	96.83	^a **
		A	92.57	92.57	92.41	92.64	
		U	50.76	50.88	49.47	50.76	
	GT	T	94.68	94.82	94.37	94.71	^a ^b ^c ^d ^e **
		K	96.84	96.92	96.69	96.87	^a ^b ^d **
		A	92.66	92.84	92.30	92.73	
		U	68.51	69.45	66.27	68.51	^a *

^aBO>GT
^bAdd>GT
^cAdd>MLE
^dMLE>GT
^eAdd>BO

Table 8: Contextual smoothing (large tagset)

Mod	Lex	Acc	MLE	Add	GT	BO	Sign
Bi	Add	T	90.37	90.52	90.46	90.47	** <i>ab</i> * <i>c</i>
		K	95.08	95.22	95.15	95.17	** <i>ab</i> * <i>c</i>
		A	90.04	90.33	90.20	90.24	
		U	30.54	30.67	30.67	30.67	
	GT	T	91.20	91.40	91.33	91.31	* <i>abc</i>
		K	95.32	95.51	95.45	95.42	* <i>bc</i>
		A	90.53	90.92	90.80	90.75	
		U	38.79	39.04	38.91	38.92	
Tri	Add	T	89.27	90.13	89.76	90.93	** <i>bd</i> <i>ef</i>
		K	93.83	94.67	94.67	95.46	** <i>defg</i> * <i>b</i>
		A	87.52	89.22	89.22	90.82	
		U	31.28	32.39	27.34	33.25	** <i>f</i> * <i>ah</i>
	GT	T	89.11	91.45	89.53	91.46	** <i>abd</i> <i>f</i>
		K	93.21	95.52	94.15	95.66	** <i>abdf</i> * <i>g</i>
		A	86.26	90.94	88.16	91.22	
		U	36.95	39.66	30.79	38.05	** <i>af</i> * <i>h</i>

^aAdd>GT^bAdd>MLE^cAdd>BO^dBO>MLE^eBO>Add^fBO>GT^gGT>MLE^hMLE>GT

5 Discussion

5.1 Tagging Model

The main result concerning the choice of tagging model is that the biclass and triclass models are more or less equivalent with respect to the data at hand. However, it should also be noted that the few differences that can be found seem to indicate that the triclass model works better with the small tagset, while the biclass model gives higher accuracy for the large tagset. The obvious conclusion to draw from this is that, in principle, the triclass model is a better model, since it takes a larger amount of context into account, but in practice, the biclass model is more robust and less sensitive to the problem of sparse data, which becomes worse with a larger tagset. Therefore, if the available training corpus is considerably smaller than the one used in these experiments, it is likely that the biclass model will be better even for a small tagset. Conversely, if only we have enough training data, the triclass model should be able to outperform the biclass model also for a larger tagset.

One way of simulating the effect of more training data is to test the tagger with a complete lexicon, i. e., to add all the unknown words in the test data with their correct part-of-speech. Under this condition, the triclass tagger does in fact outperform the biclass tagger with the large tagset and contextual back-off smoothing (96.03% vs 95.86%) but the difference is not significant. With additive contextual smoothing, we get exactly the same result for biclass and triclass (95.93%).⁴¹ So, the general conclusion to draw seems to be that lack of data and/or a large tagset may be reason to use the biclass model, even though the triclass model is theoretically superior.

5.2 Lexical Smoothing

Concerning lexical smoothing, we have seen that the more sophisticated Good-Turing estimation outperforms simple additive smoothing, a result that has been reported before in the literature (Church and Gale 1991, Gale and Sampson 1995). It is important to note, however, that the difference seems to be wholly restricted to unknown words. This calls for two comments.

First, one should not forget that tagging errors due to unknown words normally represent less than half of the total amount of errors, which means that it is equally or more important to find ways to deal with known word errors. Secondly, even though the Good-Turing method seems to give reasonable estimates of the probability that a given part-of-speech is realized as the unknown word (i. e., estimates of the probability $P(w_U|c)$ for a given part-of-speech c), we can usually do much better by combining different heuristics based on capitalization, suffixes, etc.

⁴¹With the small tagset and a complete lexicon, the triclass tagger does significantly better than the biclass tagger with additive contextual smoothing (96.97% vs 96.76%) and still better, though not significantly so, with back-off smoothing (96.91% vs 96.76%).

Still, even with such hybrid methods, there will be a need to estimate statistical parameters (such as the probability that a given part-of-speech is realized by a word ending in a certain suffix), and the experimental results reported in this article can probably be taken to show that, when dealing with lexical models, it is worth the time and effort to use a more sophisticated smoothing method such as Good-Turing estimation over simple additive smoothing.

5.3 Contextual Smoothing

Although the differences found are much smaller for contextual smoothing than for lexical smoothing, there are at least two aspects of the results that are somewhat unexpected. The first is the fact that smoothing seems not to be required at all — certainly not for the biclass model with the small tagset, and hardly for the triclass model with the small tagset or the biclass model for the large tagset. It is only for the triclass model with the large tagset that the sparse data problem really becomes noticeable. It thus seems that for the biclass model with a tagset of only 23 different tags, a training corpus of 1 million words is really sufficient to eliminate the sparse data problem as far as the contextual model is concerned.

The second surprising result is the fact that the simplest of all smoothing methods, the additive method, actually gives the best result under almost all conditions, despite previous studies which have shown additive smoothing to be inferior to, for example, Good-Turing estimation (see, e. g., Church and Gale 1991, Gale and Sampson 1995). For the contextual model, Good-Turing estimation is in fact the worst method, in some cases even giving results that are worse than those obtained with the pure MLE model. The explanation for this paradoxical state of affairs must lie in the fact that different smoothing methods are appropriate for different kinds of distributions. For instance, it seems that Good-Turing estimation gives good results for distributions where most of the outcomes have very low frequency (0 or 1), which is true for the lexical model in part-of-speech tagging as well as most statistical language models. However, the contextual model in part-of-speech tagging has a totally different distribution, especially with a small tagset and the biclass model, where most of the events under consideration have frequencies much higher than 0 or 1. Apparently, the additive method is better suited for this kind of distribution. On a more general note, this result should serve as a warning against too readily assuming that results established within one kind of statistical language application can automatically be transferred to other applications.

Finally, it should be pointed out that back-off smoothing gives results that are comparable to those obtained with additive smoothing. But since back-off smoothing is generally more complicated (since it requires the calculation of discounting factors and normalization factors), it seems that the simple method of additive smoothing can be recommended for the contextual model in statistical part-of-speech tagging.

6 Conclusion

The most important conclusion from this study is the simple observation that different models and distributions require different smoothing methods, and that we should therefore be suspicious about the applicability of results across models and application. In the specific application studied here, statistical part-of-speech tagging, we have seen that the lexical model and the contextual model require different techniques. For the lexical model, we have obtained the best results using Good-Turing estimation, a technique that only seems to make things worse in the contextual model. The contextual model, on the other hand, seems to favor additive smoothing and back-off smoothing, and for the biclass model with small tagsets the pure MLE model does quite well on its own. Let me close by saying that, although the aim of this study has been mainly theoretical, I hope that the results presented can contribute also to the construction of more accurate taggers for practical applications in the future.

References

- Baum, L. E. (1972). An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process. *Inequalities* 3, 1–8.
- Brants, T. & Samuelsson, C. (1995). Tagging the Teleman Corpus. In *Proceedings of the 10th Nordic Conference of Computational Linguistics, NODALIDA-95*, Helsinki, 7–20.
- Brill, E. (1992). A Simple Rule-based Part of Speech Tagger. In *Third Conference of Applied Natural Language Processing*, ACL.
- Brown, P. F., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L. and Roossin, P. S. (1992). Class-based n -gram models of natural language. *Computational Linguistics* 18, 467–479.
- Chanod, J.-P. & Tapanainen, P. (1995). Tagging French — Comparing a Statistical and a Constraint-based Method. In *Seventh Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, 149–156.
- Charniak, E. (1993). *Statistical Language Learning*. Cambridge, MA: MIT Press.
- Charniak, E., Hendrickson, C., Jacobson, N. and Perkowitz, M. (1993). Equations for Part-of-Speech Tagging. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, AAAI Press/MIT Press.
- Church, K. W. and Gale, W. A. (1991) A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language* 5, 19–54.
- Cutting, D., Kupiec, J., Pedersen, J. and Sibun, P. (1992). A Practical Part-of-

- speech Tagger. In *Third Conference on Applied Natural Language Processing*, ACL, 133–140.
- Daelemans, W., Zavrel, J., Berck, P. and Gillis, S. (1996). MBT: A Memory-Based Part of Speech Tagger-Generator. In Ejerhed, E. and Dagan, I. (eds) *Proceedings of the Fourth Workshop on Very Large Corpora*, Copenhagen, Denmark, 14-27.
- Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39, 1–38.
- DeRose, S. J. (1988). Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics* 14, 31–39.
- Ejerhed, E., Källgren, G., Wennstedt, O. and Åström, M. (1992). The Linguistic Annotation System of the Stockholm-Umeå Corpus Project. Report 33. University of Umeå: Department of Linguistics.
- Gale, W. A. and Church, K. W. (1990). Poor Estimates of Context Are Worse Than None. In *Proceedings of the Speech and Natural Language Workshop*, 283–287. Morgan Kaufmann.
- Gale, W. A. and Church, K. W. (1994). What is wrong with adding one? In Oostdijk, N. and de Haan, P. (eds) *Corpus-Based Research into Language*, 189–198. Amsterdam: Rodopi.
- Gale, W. A. and Sampson, G. (1995). Good-Turing Frequency Estimation Without Tears. *Journal of Quantitative Linguistics* 2, 217–237.
- Good, I. J. (1953). The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika* 43, 45–63.
- Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. MIT Press.
- Jelinek, F. and Mercer, R. (1985). Probability Distribution Estimation from Sparse Data. *IBM Technical Disclosure Bulletin* 28, 2591–2594.
- Katz, S. (1987). Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing* 35, 400–401.
- Lidstone, G. J. (1920). Note on the General Case of the Bayes-Laplace Formula for Inductive or *A Posteriori* Probabilities. *Transactions of the Faculty of Actuaries* 8, 182–192.
- Lindgren, B. W. (1993). *Statistical Theory*. Chapman-Hall.
- Merialdo, B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics* 20, 155-171.
- Ney, H., Martin, S. and Wessel, F. (1997). Statistical Language Modeling Using Leaving-One-Out. In Young, S. and Bloothoof, G. (eds) *Corpus-Based Methods*

in Language and Speech Processing. Kluwer Academic Publishers.

Viterbi, A. J. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory* 13, 260–269.

A The Small Tagset

(* = open part-of-speech)

Tag	Part-of-speech
ab*	adverb
dl	delimiter
dt	determiner
ha	wh adverb
hd	wh determiner
hp	wh pronoun
hs	wh possessive
ie	infinitive marker
in*	interjection
jj*	adjective
kn	conjunction (coordinating)
nn*	noun
pc*	participle
pl	(verb) particle
pm*	proper name
pn	pronoun
pp	preposition
ps	possessive
rg*	cardinal numeral
ro	ordinal numeral
sn	subjunction
uo*	foreign word
vb*	verb

B The Large Tagset

(* = open part-of-speech)

Tag	Part-of-speech
ab	adverb
ab an	adverb abbreviation
ab kom	adverb comparative
ab pos*	adverb positive
ab sms	adverb compound (element)
ab suv	adverb superlative
d1 mad	delimiter major
d1 mid	delimiter minor
d1 pad	delimiter paired
dt an	delimiter abbreviation
dt mas sin def	determiner masculine singular definite
dt mas sin ind	determiner masculine singular indefinite
dt neu sin def	determiner neuter singular definite
dt neu sin ind	determiner neuter singular indefinite
dt neu sin ind/def	determiner neuter singular
dt utr sin def	determiner uter singular definite
dt utr sin ind	determiner uter singular indefinite
dt utr sin ind/def	determiner uter singular
dt utr/neu plu def	determiner plural definite
dt utr/neu plu ind	determiner plural indefinite
dt utr/neu plu ind/def	determiner plural
dt utr/neu sin def	determiner singular definite
dt utr/neu sin ind	determiner singular indefinite
dt utr/neu sin/plu ind	determiner indefinite
ha	wh adverb
hd neu sin ind	wh determiner neuter singular indefinite
hd utr sin ind	wh determiner uter singular indefinite
hd utr/neu plu ind	wh determiner plural indefinite
hp	wh pronoun
hp neu sin ind	wh pronoun neuter singular indefinite
hp neu sin ind sms	wh pronoun neuter singular indefinite compound (element)
hp utr sin ind	wh pronoun uter singular indefinite
hp utr/neu plu ind	wh pronoun uter plural
hs def	wh possessive definite
ie	infinitive marker
ij*	interjection
jj an	adjective abbreviation
jj kom utr/neu sin/plu ind/def gen	adjective comparative genitive
jj kom utr/neu sin/plu ind/def nom*	adjective comparative nominative
jj kom utr/neu sin/plu ind/def sms	adjective comparative compound (element)
jj pos mas sin def gen	adjective positive masculine singular definite genitive
jj pos mas sin def nom*	adjective positive masculine singular definite nominative
jj pos neu sin ind gen	adjective positive masculine singular indefinite genitive
jj pos neu sin ind nom*	adjective positive masculine singular indefinite nominative
jj pos neu sin ind/def nom	adjective positive neuter singular nominative
jj pos utr sms	adjective positive uter compound (element)
jj pos utr sin ind gen	adjective positive uter singular indefinite genitive
jj pos utr sin ind nom*	adjective positive uter singular indefinite nominative
jj pos utr sin ind/def nom	adjective positive uter singular nominative
jj pos utr/neu sms	adjective positive compound (element)
jj pos utr/neu plu ind nom	adjective positive plural indefinite nominative
jj pos utr/neu plu ind/def gen	adjective positive plural genitive
jj pos utr/neu plu ind/def nom*	adjective positive plural nominative
jj pos utr/neu sin def gen	adjective positive singular definite genitive
jj pos utr/neu sin def nom*	adjective positive singular definite nominative
jj pos utr/neu sin/plu ind nom	adjective positive indefinite nominative
jj pos utr/neu sin/plu ind/def nom*	adjective positive
jj suv mas sin def gen	adjective superlative masculine singular definite genitive
jj suv mas sin def nom	adjective superlative masculine singular definite nominative
jj suv utr/neu plu def nom	adjective superlative plural definite nominative
jj suv utr/neu plu ind nom	adjective superlative plural indefinite nominative
jj suv utr/neu sin/plu def nom	adjective superlative definite nominative
jj suv utr/neu sin/plu ind nom	adjective superlative indefinite nominative
kn	conjunction (coordinating)
kn an	conjunction abbreviation
nn	noun
nn sms	noun compound (element)
nn an	noun abbreviation
nn neu	noun neuter
nn neu sms	noun neuter compound (element)
nn neu plu def gen*	noun neuter plural definite genitive
nn neu plu def nom*	noun neuter plural definite nominative
nn neu plu ind gen	noun neuter plural indefinite genitive
nn neu plu ind nom*	noun neuter plural indefinite nominative
nn neu sin def gen*	noun neuter singular definite genitive
nn neu sin def nom*	noun neuter singular definite nominative
nn neu sin ind gen	noun neuter singular indefinite genitive
nn neu sin ind nom*	noun neuter singular indefinite nominative

nn utr	noun uter
nn utr sms*	noun uter compound (element)
nn utr plu def gen*	noun uter plural definite genitive
nn utr plu def nom*	noun uter plural definite nominative
nn utr plu ind gen*	noun uter plural indefinite genitive
nn utr plu ind nom*	noun uter plural indefinite nominative
nn utr sin def gen*	noun uter singular definite genitive
nn utr sin def nom*	noun uter singular definite nominative
nn utr sin ind gen*	noun uter singular indefinite genitive
nn utr sin ind nom*	noun uter singular indefinite nominative
pc an	participle abbreviation
pc prf mas sin def gen	participle past masculine singular definite genitive
pc prf mas sin def nom	participle past masculine singular definite nominative
pc prf neu sin ind nom*	participle past neuter singular indefinite nominative
pc prf utr sin ind gen	participle past uter singular indefinite genitive
pc prf utr sin ind nom*	participle past uter singular indefinite nominative
pc prf utr/neu plu ind/def gen	participle past plural genitive
pc prf utr/neu plu ind/def nom*	participle past plural nominative
pc prf utr/neu sin def gen	participle past singular definite genitive
pc prf utr/neu sin def nom*	participle past singular definite nominative
pc prs utr/neu sin/plu ind/def gen	participle present genitive
pc prs utr/neu sin/plu ind/def nom*	participle present nominative
pl	(verb) particle
pl sms	(verb) particle compound (element)
pm gen*	proper name genitive
pm nom*	proper name nominative
pm sms	proper name compound (element)
pn mas sin def sub/obj	pronoun masculine singular definite
pn neu sin def sub/obj	pronoun neuter singular definite
pn neu sin ind sub/obj	pronoun neuter singular indefinite
pn utr plu def obj	pronoun uter plural definite accusative
pn utr plu def sub	pronoun uter plural definite nominative
pn utr sin def obj	pronoun uter singular definite accusative
pn utr sin def sub	pronoun uter singular definite nominative
pn utr sin def sub/obj	pronoun uter singular definite
pn utr sin ind sub	pronoun uter singular indefinite nominative
pn utr sin ind sub/obj	pronoun uter singular indefinite
pn utr/neu plu def obj	pronoun plural definite accusative
pn utr/neu plu def sub	pronoun plural definite nominative
pn utr/neu plu def sub/obj	pronoun plural definite
pn utr/neu plu ind sub/obj	pronoun plural indefinite
pn utr/neu sin/plu def obj	pronoun definite accusative
pp	preposition
pp an	preposition abbreviation
ps an	possessive abbreviation
ps neu sin def	possessive neuter singular definite
ps utr sin def	possessive uter singular definite
ps utr/neu plu def	possessive plural definite
ps utr/neu sin/plu def	possessive definite
rg gen	cardinal numeral genitive
rg mas sin def nom	cardinal numeral masculine singular definite nominative
rg neu sin ind nom	cardinal numeral neuter singular indefinite nominative
rg nom*	cardinal numeral nominative
rg sms	cardinal numeral compound (element)
rg utr sin ind nom	cardinal numeral uter singular indefinite nominative
rg utr/neu sin def nom	cardinal numeral singular definite nominative
ro gen	ordinal numeral genitive
ro mas sin ind/def gen	ordinal numeral masculine singular genitive
ro mas sin ind/def nom	ordinal numeral masculine singular nominative
ro nom	ordinal numeral nominative
ro utr/neu sin/plu ind/def sms	ordinal numeral compound (element)
sn	subjunction
uo*	foreign word
vb an	verb abbreviation
vb imp akt*	verb imperative active
vb imp sfo	verb imperative passive (form)
vb inf akt*	verb infinitive active
vb inf sfo*	verb infinitive passive (form)
vb kon prs akt	verb subjunctive present active
vb kon prt akt	verb subjunctive preterite active
vb kon prt sfo	verb subjunctive preterite passive (form)
vb prs akt*	verb present active
vb prs sfo*	verb present passive (form)
vb prt akt*	verb preterite active
vb prt sfo*	verb preterite active (form)
vb sms	verb compound (element)
vb sup akt*	verb supine active
vb sup sfo*	verb supine passive (form)