Error Measures and Bayes Decision Rules Revisited with Applications to POS Tagging

Hermann Ney, Maja Popović, David Sündermann
Lehrstuhl für Informatik VI - Computer Science Department
RWTH Aachen University
Ahornstrasse 55
52056 Aachen, Germany
{popovic,ney}@informatik.rwth-aachen.de

Abstract

Starting from first principles, we re-visit the statistical approach and study two forms of the Bayes decision rule: the common rule for minimizing the number of string errors and a novel rule for minimizing the number of symbols errors. The Bayes decision rule for minimizing the number of string errors is widely used, e.g. in speech recognition, POS tagging and machine translation, but its justification is rarely questioned. To minimize the number of symbol errors as is more suitable for a task like POS tagging, we show that another form of the Bayes decision rule can be derived. The major purpose of this paper is to show that the form of the Bayes decision rule should not be taken for granted (as it is done in virtually all statistical NLP work), but should be adapted to the error measure being used. We present first experimental results for POS tagging tasks.

1 Introduction

Meanwhile, the statistical approach to natural language processing (NLP) tasks like speech recognition, POS tagging and machine translation has found widespread use. There are three ingredients to any statistical approach to NLP, namely the Bayes decision rule, the probability models (like trigram model, HMM, ...) and the training criterion (like maximum likelihood, mutual information, ...).

The topic of this paper is to re-consider the form of the Bayes decision rule. In virtually all NLP tasks, the specific form of the Bayes decision rule is never questioned, and the decision rule is adapted from speech recognition. In speech recognition, the typical decision rule is to maximize the *sentence* probability over all possible sentences. However, this decision rule is optimal for the *sentence* error rate and not for the *word* error rate. This difference is rarely studied in the literature.

As a specific NLP task, we will consider part-of-speech (POS) tagging. However, the problem addressed comes up in *any* NLP task which is tackled by the *statistical approach* and which makes use of a Bayes decision rule. Other prominent examples are speech recognition and machine translation. The advantage of the POS tagging task is that it will be easier to handle from the mathematical point of view and will result in closed-form solutions for the decision rules. From this point-of-view, the POS tagging task serves as a good opportunity to illustrate the key concepts of the statistical approach to NLP.

Related Work: For the task of POS tagging, statistical approaches were proposed already in the 60's and 70's (Stolz et al., 1965; Bahl and Mercer, 1976), before they started to find widespread use in the 80's (Beale, 1985; DeRose, 1989; Church, 1989).

To the best of our knowledge, the 'standard' version of the Bayes decision rule, which minimizes the number of string errors, is used in virtually all approaches to POS tagging and other NLP tasks. There are only two research groups that do not take this type of decision rule for granted:

(Merialdo, 1994): In the context of POS tagging, the author introduces a method that he calls maximum likelihood tagging. The spirit of this method is similar to that of this work. However, this method is mentioned as an aside and its implications for the Bayes decision rule and the statistical approach are not addressed. Part of this work goes back to (Bahl et al., 1974) who considered a problem in coding theory.

(Goel and Byrne, 2003): The error measure considered by the authors is the word error rate in speech recognition, i.e. the edit distance. Due to the mathematical complexity of this error measure, the authors resort to numeric approximations to compute the Bayes risk (see next section). Since this approach does not results in explicit closed-form equations and involves many numeric approximations, it is not easy to draw conclusions from this work.

2 Bayes Decision Rule for Minimum Error Rate

2.1 The Bayes Posterior Risk

Knowing that any task in NLP tasks is a difficult one, we want to keep the number of wrong decisions as small as possible. This point-of-view has been used already for more than 40 years in pattern classification as the starting point for many techniques in pattern classification. To classify an observation vector y into one out of several classes c, we resort to the so-called statistical decision theory and try to minimize the average *risk* or *loss* in taking a decision. The result is known as *Bayes decision rule* (Chapter 2 in (Duda and Hart, 1973)):

$$y \rightarrow \hat{c} \quad = \quad \arg\min_{c} \left\{ \sum_{\tilde{c}} Pr(c|y) \cdot L[c, \tilde{c}] \right\}$$

where $L[c, \tilde{c}]$ is the so-called loss function or error measure, i.e. the loss we incur in making decision c when the true class is $\tilde{c}$.

In the following, we will consider two specific forms of the loss function or error measure $L[c, \tilde{c}]$. The first will be the measure for string errors, which is the typical loss function used in virtually all statistical approaches. The second is the measure for symbol errors, which is the more appropriate measure for POS tagging and also speech recognition with no insertion and deletion errors (such as isolated word recognition).

2.2 String Error

For POS tagging, the starting point is the observed sequence of words $y = w_1^N = w_1...w_N$, i.e. the sequence of words for which the POS tag sequence has $c = g_1^N = g_1...g_N$ has to be determined.

The first error measure we consider is the string error: the error is equal to zero only if the POS symbols of the two strings are identical at each position. In this case, the loss function is:

$$L[g_1^N, \tilde{g}_1^N] \quad = \quad 1 - \prod_{n=1}^{N} \delta(g_n, \tilde{g}_n)$$

with the Kronecker delta $\delta(c, \tilde{c})$. In other words, the errors are counted at the string level and not at the level of single symbols. Inserting this cost function into the Bayes risk (see Section 2.1), we immediately obtain the following form of *Bayes decision rule for minimum string error*:

$$\begin{aligned} w_1^N \rightarrow \hat{g}_1^N \quad &= \quad \arg\max_{g_1^N} \left\{ Pr(g_1^N | w_1^N) \right\} \\ &= \quad \arg\max_{g_1^N} \left\{ Pr(g_1^N, w_1^N) \right\} \end{aligned}$$

This is the starting point for virtually all statistical approaches in NLP like speech recognition and machine translation. However, this decision rule is only optimal when we consider *string* errors, e.g. sentence error rate in POS tagging and in speech recognition. In practice, however, the empirical errors are counted at the *symbol* level. Apart from (Goel and Byrne, 2003), this inconsistency of decision rule and error measure is never addressed in the literature.

2.3 Symbol Error

Instead of the *string* error rate, we can also consider the error rate of *single POS tag symbols* (Bahl et al., 1974; Merialdo, 1994).

This error measure is defined by the loss function:

$$L[g_1^N, \tilde{g}_1^N] \quad = \quad \sum_{n=1}^{N} [1 - \delta(g_n, \tilde{g}_n)]$$

This loss function has to be inserted into the Bayes decision rule in Section 2.1. The computation of the expected loss, i.e. the averaging over all classes $\tilde{c} = \tilde{g}_1^N$, can be performed in a closed form. We omit the details of the straightforward calculations and state only the result. It turns out that we will need the marginal (and posterior) probability distribution $Pr_m(g|w_1^N)$ at positions $m = 1, ..., N$:

$$Pr_m(g|w_1^N) \quad := \quad \sum_{g_1^N : \, g_m = g} Pr(g_1^N | w_1^N)$$

where the sum is carried out over all POS tag strings g_1^N with $g_m = g$, i.e. the tag g_m at position m is fixed at $g_m = g$. The question of how to perform this summation efficiently will be considered later after we have introduced the model distributions.

Thus we have obtained the *Bayes decision rule for minimum symbol error* at position $m = 1, ..., N$:

$$\begin{aligned} (w_1^N, m) \rightarrow \hat{g}_m \quad &= \quad \arg\max_{g} \left\{ Pr_m(g|w_1^N) \right\} \\ &= \quad \arg\max_{g} \left\{ Pr_m(g, w_1^N) \right\} \end{aligned}$$

By construction this decision rule has the special property that it does not put direct emphasis on local coherency of the POS tags produced. In other words, this decision rule may produce a POS tag *string* which is linguistically less likely.

3 The Modelling Approaches to POS Tagging

The derivation of the Bayes decision rule assumes that the probability distribution $Pr(g_1^N, w_1^N)$ (or $Pr(g_1^N | w_1^N)$) is known. Unfortunately, this is not the case in practice. Therefore, the usual approach is to approximate the true but unknown distribution by a *model distribution* $p(g_1^N, w_1^N)$ (or $p(g_1^N|w_1^N)$). We will review two popular modelling approaches, namely the generative model and the direct model, and consider the associated Bayes decision rules for both minimum string error and minimum symbol error.

3.1 Generative Model: Trigram Model

We replace the true but unknown *joint* distribution $Pr(g_1^N, w_1^N)$ by a model-based probability distribution $p(g_1^N, w_1^N)$:

$$Pr(g_1^N, w_1^N) \to p(g_1^N, w_1^N) = p(g_1^N) \cdot p(w_1^N|g_1^N)$$

We apply the so-called *chain rule* to factorize each of the distributions $p(g_1^N)$ and $p(w_1^N|g_1^N)$ into a product of *conditional probabilities* using specific dependence assumptions:

$$p(g_1^N, w_1^N) \;=\; \prod_{n=1}^{N} \Big[p(g_n|g_{n-2}^{n-1}) \cdot p(w_n|g_n) \Big]$$

with suitable definitions for the case $n = 1$. Here, the specific dependence assumptions are that the conditional probabilities can be represented by a POS trigram model $p(g_n|g_{n-2}^{n-1})$ and a word membership model $p(w_n|g_n)$. Thus we obtain a probability model whose structure fits into the mathematical framework of so-called *Hidden Markov Model (HMM)*. Therefore, this approach is often also referred to as HMM-based POS tagging. However, this terminology is misleading: The POS tag sequence is observable whereas in the Hidden Markov Model the state sequence is always hidden and cannot be observed. In the experiments, we will use a 7-gram POS model. It is clear how to extend the equations from the trigram case to the 7-gram case.

3.1.1 String Error

Using the above model distribution, we directly obtain the decision rule for minimum string error:

$$w_1^N \to \hat{g}_1^N \;=\; \arg\max_{g_1^N} \Big\{ p(g_1^N, w_1^N) \Big\}$$

Since the model distribution is a basically a second-order model (or trigram model), there is an efficient algorithm for finding the most probable POS tag string. This is achieved by a suitable dynamic programming algorithm, which is often referred to as Viterbi algorithm in the literature.

3.1.2 Symbol Error

To apply the Bayes decision rule for minimum symbol error rate, we first compute the marginal probability $p_m(g, w_1^N)$:

$$\begin{aligned} p_m(g, w_1^N) \;&=\; \sum_{g_1^N:\, g_m=g} p(g_1^N, w_1^N) \\ &=\; \sum_{g_1^N:\, g_m=g} \prod_n \Big[p(g_n|g_{n-2}^{n-1}) \cdot p(w_n|g_n) \Big] \end{aligned}$$

Again, since the model is a second-order model, the sum over all possible POS tag strings g_1^N (with $g_m = g$) can be computed efficiently using a suitable extension of the forward-backward algorithm (Bahl et al., 1974).

Thus we obtain the decision rule for minimum symbol error at positions $m = 1, ..., N$:

$$(w_1^N, m) \to \hat{g}_m \;=\; \arg\max_{g} \Big\{ p_m(g, w_1^N) \Big\}$$

Here, after the the marginal probability $p_m(g, w_1^N)$ has been computed, the task of finding the most probable POS tag at position m is computationally easy. Instead, the lion's share for the computational effort is required to compute the marginal probability $p_m(g, w_1^N)$.

3.2 Direct Model: Maximum Entropy

We replace the true but unknown *posterior* distribution $Pr(g_1^N|w_1^N)$ by a model-based probability distribution $p(g_1^N|w_1^N)$:

$$Pr(g_1^N|w_1^N) \to p(g_1^N|w_1^N)$$

and apply the chain rule:

$$\begin{aligned} p(g_1^N|w_1^N) \;&=\; \prod_{n=1}^{N} p(g_n|g_1^{n-1}, w_1^N) \\ &=\; \prod_{n=1}^{N} p(g_n|g_{n-2}^{n-1}, w_{n-2}^{n+2}) \end{aligned}$$

As for the generative model, we have made specific assumptions: There is a second-order dependence for the tags g_1^n, and the dependence on the words w_1^N is limited to a window w_{n-2}^{n+2} around position n. The resulting model is still rather complex and requires further specifications. The typical procedure is to resort to *log-linear modelling*, which is also referred to as *maximum entropy modelling* (Ratnaparkhi, 1996; Berger et al., 1996).

3.2.1 String Error

For the minimum string error, we obtain the decision rule:

$$w_1^N \to \hat{g}_1^N \;=\; \arg\max_{g_1^N} \Big\{ p(g_1^N|w_1^N) \Big\}$$

Since this is still a second-order model, we can use dynamic programming to compute the most likely POS string.

3.2.2 Symbol Error

For the minimum symbol error, the marginal (and posterior) probability $p_m(g|w_1^N)$ has to be computed:

$$\begin{aligned} p_m(g|w_1^N) &= \sum_{g_1^N:\,g_m=g} Pr(g_1^N|w_1^N) \\ &= \sum_{g_1^N:\,g_m=g} \prod_n p(g_n|g_{n-2}^{n-1}, w_{n-2}^{n+2}) \end{aligned}$$

which, due to the specific structure of the model $p(g_n|g_{n-2}^{n-1}, w_{n-2}^{n+2})$, can be calculated efficiently using only a *forward* algorithm (without a 'backward' part).

Thus we obtain the decision rule for minimum symbol error at positions $m = 1, ..., N$:

$$(w_1^N, m) \rightarrow \hat{g}_m = \arg\max_g \left\{ p_m(g|w_1^N) \right\}$$

As in the case of the generative model, the computational effort is to compute the posterior probability $p_m(g|w_1^N)$ rather than to find the most probable tag at position m.

4 The Training Procedure

So far, we have said nothing about how we train the free parameters of the model distributions. We use fairly conventional training procedures that we mention only for the sake of completeness.

4.1 Generative Model

We consider the trigram-based model. The free parameters here are the entries of the POS trigram distribution $p(g|g'', g')$ and of the word membership distribution $p(w|g)$. These unknown parameters are computed from a *labelled training corpus,* i.e. a collection of sentences where for each word the associated POS tag is given.

In principle, the free parameters of the models are estimated as relative frequencies. For the test data, we have to allow for both POS trigrams (or n-grams) and (single) words that were not seen in the training data. This problem is tackled by applying *smoothing* methods that were originally designed for language modelling in speech recognition (Ney et al., 1997).

4.2 Direct Model

For the maximum entropy model, the free parameters are the so-called λ_i or feature parameters (Berger et al., 1996; Ratnaparkhi, 1996). The training criterion is to optimize the logarithm of the model probabilities $p(g_n|g_{n-1}^{n-2}, w_{n-2}^{n+2})$ over all positions n in the training corpus. The corresponding algorithm is referred to as GIS algorithm (Berger et al., 1996). As usual with maximum entropy models, the problem of smoothing does not seem to be critical and is not addressed explicitly.

5 Experimental Results

Of course, there have already been many papers about POS tagging using statistical methods. The goal of the experiments is to compare the two decision rules and to analyze the differences in performance. As the results for the WSJ corpus will show, both the trigram method and the maximum entropy method have an tagging error rate of 3.0% to 3.5% and are thus comparable to the best results reported in the literature, e.g. (Ratnaparkhi, 1996).

5.1 Task and Corpus

The experiments are performed on the Wall Street Journal (WSJ) English corpus and on the Münster Tagging Project (MTP) German corpus.

The POS tagging part of The WSJ corpus (Table 1) was compiled by the University of Pennsylvania and consists of about one million English words with manually annotated POS tags.

		Text	POS
Train	Sentences	43508	
	Words+PMs	1061772	
	Singletons	21522	0
	Word Vocabulary	46806	45
	PM Vocabulary	25	9
Test	Sentences	4478	
	Words+PMs	111220	
	OOVs	2879	0

Table 1: WSJ corpus statistics.

The MTP corpus (Table 2) was compiled at the University of Münster and contains tagged German words from articles of the newspapers *Die Zeit* and *Frankfurter Allgemeine Zeitung* (Kinscher and Steiner, 1995).

For the corpus statistics, it is helpful to distinguish between the true words and the punctuation marks (see Table 1 and Table 2). This distinction is made for both the text and the POS corpus. In addition, the tables show the vocabulary size (number of different tokens) for the words and for the punctuation marks.

Punctuation marks (PMs) are all tokens which do not contain letters or digits. The total number of running tokens is indicated as Words+PMs. Singletons are the tokens which occur only once in

		Text	POS
Train	Sentences	19845	
	Words+PMs	349699	
	Singletons	32678	11
	Word Vocabulary	51491	68
	PM Vocabulary	27	5
Test	Sentences	2206	
	Words+PMs	39052	
	OOVs	3584	2

Table 2: MTP corpus statistics.

the training data. Out-of-Vocabulary words (OOVs) are the words in the test data that did not not occur in the training corpus.

5.2 POS Tagging Results

The tagging experiments were performed for both types of models, each of them with both types of the decision rules. The generative model is based on the approach described in (Sündermann and Ney, 2003). Here the optimal value of the n-gram order is determined from the corpus statistics and has a maximum of $n = 7$. The experiments for the direct model were performed using the maximum entropy tagger described in (Ratnaparkhi, 1996).

The tagging error rates are showed in Table 3 and Table 4. In addition to the overall tagging error rate (Overall), the tables show the tagging error rates for the Out-of-Vocabulary words (OOVs) and for the punctuation marks (PMs).

For the generative model, both decision rules yield similar results. For the direct model, the overall tagging error rate increases on each of the two tasks (from 3.0 % to 3.3 % on WSJ and from 5.4 % to 5.6 % on MTP) when we use the symbol decision rule instead of the string decision rule. In particular, for OOVs, the error rate goes up clearly. Right now, we do not have a clear explanation for this difference between the generative model and the direct model. It might be related to the 'forward' structure of the direct model as opposed to the 'forward-backward' structure of the generative model. Anyway, the refined bootstrap method (Bisani and Ney, 2004) has shown that differences in the overall tagging error rate are statistically not significant.

5.3 Examples

A detailed analysis of the tagging results showed that for both models there are sentences where the one decision rule is more efficient and sentences where the other decision rule is better.

For the generative model, these differences seem to occur at random, but for the direct model, some distinct tendencies can be observed. For example, for the WSJ corpus, the string decision rule is significantly better for the present and past tense of verbs (VBP, VBN), and the symbol decision rule is better for adverb (RB) and verb past participle (VBN). Typical errors generated by the symbol decision rule are tagging present tense as infinitive (VB) and past tense as past participle (VBN), and for string decision rule, adverbs are often tagged as preposition (IN) or adjective (JJ) and past participle as past tense (VBD).

WSJ Task	Decision Rule	Overall	OOVs	PMs
Generative Model	string	3.5	16.9	0
	symbol	3.5	16.7	0
Direct Model	string	3.0	15.4	0.08
	symbol	3.3	16.6	0.1

Table 3: POS tagging error rates [%] for WSJ task.

MTP Task	Decision Rule	Overall	OOVs	PMs
Generative Model	string	5.4	13.4	3.6
	symbol	5.4	13.4	3.6
Direct Model	string	5.4	12.7	3.8
	symbol	5.6	13.4	3.7

Table 4: POS tagging error rates [%] for MTP task.

For the German corpus, the string decision rule better handles demonstrative determiners (Rr) and subordinate conjunctions (Cs) whereas symbol decision rule is better for definite articles (Db). The symbol decision rule typically tags the demonstrative determiner as definite article (Db) and subordinate conjunctions as interrogative adverbs (Bi), and the string decision rule tends to assign the demonstrative determiner tag to definite articles.

These typical errors for the symbol decision rule are shown in Table 5, and for the string decision rule in Table 6.

6 Conclusion

So far, the experimental tests have shown no improvement when we use the Bayes decision rule for minimizing the number of symbol errors rather than the number of string errors. However, the important result is that the new approach results in comparable performance. More work is needed to contrast the two approaches.

The main purpose of this paper has been to show that, in addition to the widely used decision rule for minimizing the string errors, it is possible to derive a decision rule for minimizing the number of symbol errors and to build up the associated mathematical framework.

There are a number of open questions for future work:

1) The error rates for the two decision rules are comparable. Is that an experimental coincidence? Are there situations for which we must expect a significance difference between the two decision rules? We speculate that the two decision rules could *always* have similar performance if the error rates are small.

2) Ideally, the training criterion should be closely related to the error measure used in the decision rule. Right now, we have used the training criteria that had been developed in the past and that had been (more or less) designed for the string error rate as error measure. Can we come up with a training criterion tailored to the symbol error rate?

3) In speech recognition and machine translation, more complicated error measures such as the edit distance and the BLEU measure are used. Is it possible to derive closed-form Bayes decision rules (or suitable analytic approximations) for these error measures? What are the implications?

References

L. Bahl, J. Cocke, F. Jelinek and J. Raviv. 1974. Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. *IEEE Trans. on Information Theory*, No. 20, pages 284–287

L. Bahl and L. R. Mercer. 1976. Part of Speech Assignment by a Statistical Decision Algorithm. In *IEEE Symposium on Information Theory*, abstract, pages 88–89, Ronneby, Sweden.

A. D. Beale. 1985. A Probabilistic Approach to Grammatical Analysis of Written English by Computer. In *2nd Conf. of the European Chapter of the ACL*, pages 159–169, Geneva, Switzerland.

A. L. Berger, S. Della Pietra and V. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, No. 22, Vol. 1, pages 39–71.

M. Bisani and H. Ney. 2004. Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 409–412, Montreal, Canada.

K. W. Church. 1989. A Stochastic Parts Program Noun Phrase Parser for Unrestricted Text. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 695–698, Glasgow, Scotland.

S. DeRose. 1989. Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, No. 14, Vol. 1, pages 31–39

R. O. Duda and P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.

V. Goel and W. Byrne. 2003. Minimum Bayes-risk Automatic Speech Recognition. In W. Chou and B. H. Juang (editors): *Pattern Recognition in Speech and Language Processing*. CRC Press, Boca Rota, Florida.

J. Kinscher and P. Steiner. 1995. Münster Tagging Project (MTP). *Handout for the 4th Northern German Linguistic Colloquium*, University of Münster, Internal report.

B. Merialdo. 1994. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, No. 20, Vol. 2, pages 155–168.

H. Ney, S. Martin and F. Wessel. 1997. Statistical Language Modelling by Leaving-One-Out. In G. Bloothooft and S. Young (editors): *Corpus-Based Methods in Speech and Language*, pages 174–207. Kluwer Academic Publishers, Dordrecht.

A. Ratnaparkhi. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora* , pages 133–142, Sommerset, NJ.

W. S. Stolz, P. H. Tannenbaum and F. V. Carstensen. 1965. Stochastic Approach to the Grammatical Coding of English. *Communications of the ACM*, No. 8, pages 399–405.

D. Sündermann and H. Ney. 2003. SYNTHER - a New m-gram POS Tagger. In *Proc. of the Int. Conf. on Natural Language Processing and Knowledge Engineering*, pages 628–633, Beijing, China.

VBP → VB	
reference	... investors/NNS already/RB *have/VBP* sharply/RB scaled/VBN ...
string	... investors/NNS already/RB *have/VBP* sharply/RB scaled/VBN ...
symbol	... investors/NNS already/RB **have/VB** sharply/RB scaled/VBN ...
reference	We/PRP basically/RB *think/VBP* that/IN ...
string	We/PRP basically/RB *think/VBP* that/IN ...
symbol	We/PRP basically/RB **think/VB** that/IN ...
VBD → VBN	
reference	... plant-expansion/JJ program/NN *started/VBD* this/DT year/NN ...
string	... plant-expansion/NN program/NN *started/VBD* this/DT year/NN ...
symbol	... plant-expansion/NN program/NN **started/VBN** this/DT year/NN ...
reference	... countries/NNS have/VBP in/IN recent/JJ years/NNS *made/VBD* agreements/NNS ...
string	... countries/NNS have/VBP in/IN recent/JJ years/NNS *made/VBD* agreements/NNS ...
symbol	... countries/NNS have/VBP in/IN recent/JJ years/NNS **made/VBN** agreements/NNS ...
Rr → Db	
reference	Das/Db Sandmännchen/Ne ,/Fi *das/Rr* uns/Rp der/Db NDR/Ab präsentiert/Vf ...
string	Das/Db Sandmännchen/Ng ,/Fi *das/Rr* uns/Rp der/Db NDR/Ab präsentiert/Vf ...
symbol	Das/Db Sandmännchen/Ng ,/Fi **das/Db** uns/Rp der/Db NDR/Ab präsentiert/Vf ...
reference	... für/Po Leute/Ng ,/Fi *die/Rr* glauben/Vf ...
string	... für/Po Leute/Ng ,/Fi *die/Rr* glauben/Vf ...
symbol	... für/Po Leute/Ng ,/Fi **die/Db** glauben/Vf ...
Cs → Bi	
reference	Denke/Vf ich/Rp nach/Qv ,/Fi *warum/Cs* mir/Rp die/Db Geschichte/Ng gefällt/Vf ...
string	Denke/Vf ich/Rp nach/Qv ,/Fi *warum/Cs* mir/Rp die/Db Geschichte/Ng gefällt/Vf ...
symbol	Denke/Vf ich/Rp nach/Qv ,/Fi **warum/Bi** mir/Rp die/Db Geschichte/Ng gefällt/Vf ...

Table 5: Examples of tagging errors for the symbol decision rule (direct model)

RB → IN, JJ	
reference	The/DT negotiations/NNS allocate/VBP *about/RB* 15/CD %/NN ...
string	The/DT negotiations/NNS allocate/VBP **about/IN** 15/CD %/NN ...
symbol	The/DT negotiations/NNS allocate/VBP *about/RB* 15/CD %/NN ...
reference	... will/MD lead/VB to/TO a/DT *much/RB* stronger/JJR performance/NN ...
string	... will/MD lead/VB to/TO a/DT **much/JJ** stronger/JJR performance/NN ...
symbol	... will/MD lead/VB to/TO a/DT *much/RB* stronger/JJR performance/NN ...
VBN → VBD	
reference	... by/IN a/DT police/NN officer/NN *named/VBN* John/NNP Klute/NNP ...
string	... by/IN a/DT police/NN officer/NN **named/VBD** John/NNP Klute/NNP ...
symbol	... by/IN a/DT police/NN officer/NN *named/VBN* John/NNP Klute/NNP ...
Db → Rr	
reference	er/Rp kam/Vf auf/Po die/Db Idee/Ng ,/Fi *die/Db* Emotionen/Ng zu/Qi kanalisieren/Vi ...
string	er/Rp kam/Vf auf/Po die/Db Idee/Ng ,/Fi **die/Rr** Emotionen/Ng zu/Qi kanalisieren/Vi ...
symbol	er/Rp kam/Vf auf/Po die/Db Idee/Ng ,/Fi *die/Db* Emotionen/Ng zu/Qi kanalisieren/Vi ...

Table 6: Examples of tagging errors for the string decision rule (direct model)