Beyond N in N-gram Tagging

Robbert Prins
Alfa-Informatica
University of Groningen
P.O. Box 716, NL-9700 AS Groningen
The Netherlands
r.p.prins@let.rug.nl

Abstract

The Hidden Markov Model (HMM) for part-of-speech (POS) tagging is typically based on tag trigrams. As such it models local context but not global context, leaving long-distance syntactic relations unrepresented. Using n-gram models for $n > 3$ in order to incorporate global context is problematic as the tag sequences corresponding to higher order models will become increasingly rare in training data, leading to incorrect estimations of their probabilities.

The trigram HMM can be extended with global contextual information, without making the model infeasible, by incorporating the context separately from the POS tags. The new information incorporated in the model is acquired through the use of a wide-coverage parser. The model is trained and tested on Dutch text from two different sources, showing an increase in tagging accuracy compared to tagging using the standard model.

1 Introduction

The Hidden Markov Model (HMM) used for part-of-speech (POS) tagging is usually a second-order model, using tag trigrams, implementing the idea that a limited number of preceding tags provide a considerable amount of information on the identity of the current tag. This approach leads to good results. For example, the `TnT` trigram HMM tagger achieves state-of-the-art tagging accuracies on English and German (Brants, 2000). In general, however, as the model does not consider global context, mistakes are made that concern long-distance syntactic relations.

2 A restriction of HMM tagging

The simplifying assumption, which is the basis for HMM tagging, that the context of a given tag can be fully represented by just the previous two tags, leads to tagging errors where syntactic features that fall outside of this range, and that are needed for determining the identity of the tag at hand, are ignored.

One such error in tagging Dutch is related to finiteness of verbs. This is discussed in the next paragraph and will be used in explaining the proposed approach. Other possible applications of the technique include assignment of case in German, and assignment of chunk tags in addition to part-of-speech tags. These will be briefly discussed at the end of this paper.

2.1 An example from Dutch

In experiments on tagging Dutch text performed in the context of (Prins and van Noord, 2004), the most frequent type of error is a typical example of a mistake caused by a lack of access to global context. In Dutch, the plural finite form of a verb is similar in appearance to the infinitive form of the verb. In example (1-a) the second verb in the sentence, *vliegen*, is correctly tagged as an infinitive, but in example (1-b) the added adverb creates

a surrounding in which the tagger incorrectly labels the verb as the finite plural form.

(1) a. Jan zag–past_sg vogels vliegen–inf
Jan saw birds fly

b. *Jan zag–past_sg vogels vliegen–pl
Jan saw birds fly
gisteren
yesterday

Since a clause normally contains precisely one finite verb, this mistake could be avoided by remembering whether the finite verb for the current clause has already occurred, and using this information in classifying a newly observed verb as either finite or nonfinite. The trigram tagger has normally "forgotten" about any finite verb upon reaching a second verb, and is led into a mistake by other parts of the context even if the two verbs are close to each other.

Basing the model on n-grams bigger than trigrams is not a solution as the n-grams would often not occur in the training data, making the associated probabilities hard to estimate.

3 Extending the model

Instead of considering longer n-grams, the model can be extended with specific long-distance context information. Analogous to how sequences of tags can be modeled as a probabilistic network of events, modeling the probability of a tag given a number of preceding tags, in the same way we can model the syntactic context.

For the example problem presented in section 2.1, this network would consist of two states: pre and $post$. In state pre the finite verb for the current clause has not yet been seen, while in state $post$ is has. In general, the context feature C with values $C_{1...j}$ and its probability distribution is to be incorporated in the model.

In describing how the extra context information is added to the HMM, we will first look at how the standard model for POS tagging is constructed. Then the probability distribution on which the new model is based is introduced. A distinction is made between a naive approach where the extra context is added to the model by extending the tagset, and a method where the context is added separately from the tags which results in a much smaller increase in the number of probabilities to be estimated from the training data.

3.1 Standard model

In the standard second order HMM used for POS tagging (as described for example in chapter 10.2 of (Manning and Schütze, 1999)), a single state corresponds to two POS tags, and the observed symbols are words. The transitions between states are governed by probabilities that combine the probabilities for state transitions (tag sequences t_{i-2}, t_{i-1}, t_i) and output of observed symbols (words w_i):

$$P(t_i, w_i | t_{i-2}, t_{i-1})$$

This probability distribution over tags and words is factorized into two separate distributions, using the chain rule $P(A, B|C) = P(A|C) \cdot P(B|C, A)$:

$$P(t_i, w_i | t_{i-2}, t_{i-1}) = \\ P(t_i | t_{i-2}, t_{i-1}) \cdot P(w_i | t_{i-2}, t_{i-1}, t_i)$$

Finally, the POS tagging assumption that the word only depends on the current tag is applied:

$$P(t_i, w_i | t_{i-2}, t_{i-1}) \approx P(t_i | t_{i-2}, t_{i-1}) \cdot P(w_i | t_i)$$

If τ is the size of the tagset, ω the size of the vocabulary, and n the length of the tag n-grams used, then the number of parameters in this standard model is $\tau^n + \tau\omega$.

3.2 Extended model

As a starting point in adding the extra feature to the model, the same probability distribution used as a basis for the standard model is used:

$$P(t_i, w_i | t_{i-2}, t_{i-1})$$

Naive method: extending the tagset. The contextual information C with j possible values could be added to the model by extending the set of tags, so that every tag t in the tagset is replaced by a set of tags $\{t_{c_1}, t_{c_2}, \ldots, t_{c_j}\}$. If τ is the size of the original tagset, then the number of parameters in this extended model would be $\tau^n j^n + \tau j\omega$, the number of tag n-grams being multiplied by eight in our example. In experiments this increase in the number of parameters led to less accurate probability estimates.

Better method: adding context to states as a separate feature. In order to avoid the problem associated with the naive method, the context feature is added to the states of the model separately from the tags. This way it is possible to combine probabilities from the different distributions in an appropriate manner, restricting the increase in the number of parameters. For example, it is now stated that as far as the context feature is concerned, the model is first order. The probabilities associated with state transitions are defined as follows, where c_i is the value of the new context feature at position i:

$$P(t_i, w_i, c_i | t_{i-2}, t_{i-1}, c_{i-1})$$

As before, the probability distribution is factorized into separate distributions:

$$\begin{aligned} &P(t_i, w_i, c_i | t_{i-2}, t_{i-1}, c_{i-1}) = \\ &P(t_i | t_{i-2}, t_{i-1}, c_{i-1}) \cdot \\ &P(c_i | t_{i-2}, t_{i-1}, c_{i-1}, t_i) \cdot \\ &P(w_i | t_{i-2}, t_{i-1}, c_{i-1}, t_i, c_i) \end{aligned}$$

The assumption made in the standard POS tagging model that words only depend on the corresponding tag is applied, as well as the assumption that the current context value only depends on the current tag and the previous context value:

$$\begin{aligned} &P(t_i, w_i, c_i | t_{i-2}, t_{i-1}, c_{i-1}) \approx \\ &P(t_i | t_{i-2}, t_{i-1}, c_{i-1}) \cdot \\ &P(c_i | c_{i-1}, t_i) \cdot \\ &P(w_i | t_i) \end{aligned}$$

The total numbers of parameters for this model is $\tau^n j + \tau j^2 + \tau\omega$. In the case of the example problem this means the number of tag n-grams is multiplied by two. The experiments described in section 5 will make use of this model.

3.3 Training the model

The model's probabilities are estimated from annotated training data. Since the model is extended with global context, this has to be part of the annotation. The Alpino wide-coverage parser for Dutch (Bouma et al., 2001) was used to automatically add the extra information to the data. For the example concerning finite plural verbs and infinitives, this means the parser labels every word in the sentence with one of the two possible context values. When the parser encounters a root clause (including imperative clauses and questions) or a subordinate clause (including relative clauses), it assigns the context value *pre*. When a finite verb is encountered, the value *post* is assigned. Past the end of a root clause or subordinate clause the context is reset to the value used before the embedded clause began. In all other cases, the value assigned to the previous position is continued.

From the text annotated with POS tags and context labels the n-gram probabilities and lexical probabilities needed by the model are estimated based on the frequencies of the corresponding sequences.

4 The tagger

4.1 Tagging method

The trigram HMM tagger used in the experiments of section 5 computes the *a posteriori* probability for every tag. This value is composed of the forward and backward probability of the tag at hand as defined in the forward-backward algorithm for HMM-training. This idea is also described in (Jelinek, 1998) and (Charniak et al., 1996). The trigram data is combined with bigram and unigram data through linear interpolation to reduce the problem of sparse data.

4.1.1 Smoothing

Applying the method known as linear interpolation, probabilities of unigrams, bigrams and trigrams are combined in a weighted sum using weights λ_1, λ_2 and λ_3 respectively. The weights are computed for every individual case using the notion of n-gram *diversity* (Collins, 1999). The diversity of an n-gram is the number of different tags that appear in the position following this n-gram in the training data. The weight λ_3 assigned to the trigram $t_1 t_2 t_3$ is computed on the basis of the diversity and frequency of the prefix bigram $t_1 t_2$, using the following equation, where c regulates the importance of diversity ($c = 6$ was used in the experiments described below), and $C(x)$ and $D(x)$ are respectively the count and diversity of x:

$$\lambda_3 = \begin{cases} 0 & \text{if } C(t_1 t_2) = 0 \\ \frac{C(t_1 t_2)}{C(t_1 t_2) + c \times D(t_1 t_2)} & \text{if } C(t_1 t_2) > 0 \end{cases}$$

The bigram weight λ_2 is computed as a fraction of $1 - \lambda_3$ using the bigram version of the above equation. The remaining weight $1 - \lambda_3 - \lambda_2$ is used as the unigram weight λ_1.

4.1.2 Unknown words

The tagger uses a lexicon that has been created from the training data to assign an initial set of possible tags to every word. Words that were not seen during training are not in the lexicon, so that another method has to be used to assign initial tags to these words. A technique described and implemented by Jan Daciuk (Daciuk, 1999) was used to create automata for associating words with tags based on suffixes of those words.

5 Tagging experiment

5.1 Experiment setup

5.1.1 Method

An extended model was created featuring context information on the occurrence of the finite verb form. The tagger is used to tag a set of sentences, assigning one tag to each word, first using the standard model and then using the extended model. The results are compared in terms of tagging accuracy. The experiment is conducted twice with different data sets used for both training and testing.

5.1.2 Data

The first set consists of a large amount of Dutch newspaper text that was annotated with syntactical tags by the Alpino parser. This is referred to as the "Alpino" data. The second and much smaller set of data is the Eindhoven corpus tagged with the Wotan tagset (Berghmans, 1994). This data set was also used in (van Halteren et al., 2001), therefore the second experiment will allow for a comparison of the results with previous work on tagging Dutch. This data will be referred to as the "Wotan" data.

For both sets the contextual information concerning finite verbs is added to the training data by the Alpino parser as described in section 3.3. Due to memory restrictions, the parser was not able to parse 265 of the 36K sentences of Wotan training data. These sentences received no contextual labels and thus not all of the training data used in (van Halteren et al., 2001) could be used in the Wotan experiment.

Training data for the Alpino experiment is four years of daily newspaper text, amounting to about 2M sentences (25M words). Test data is a collection of 3686 sentences (59K words) from the *Parool* newspaper. The data is annotated with a tagset consisting of 2825 tags. (The large size of the Alpino tagset is mainly due to a large number of infrequent tags representing specific uses of prepositions.) In the Wotan experiment, 36K sentences (628K words) are used for training (compared to 640K words in (van Halteren et al., 2001)), and 4176 sentences (72K words) are used for testing. The Wotan data is annotated with a tagset consisting of 345 tags (although a number of 341 is reported in (van Halteren et al., 2001)).

5.1.3 Baseline method

As a baseline method every word is assigned the tag it was most often seen with in the training data. Thus the baseline method is to tag each word w with a tag t such that $P(t|w)$ is maximized. Unknown words are represented by all words that occurred only once. The baseline accuracies are 85.9% on the Alpino data and 84.3% on the Wotan data.

5.2 Results

5.2.1 "Alpino" experiment

The results on the Alpino data are shown in table 1. Using the standard model, accuracy is 93.34% (3946 mistakes). Using the extended model, accuracy is 93.62% (3779 mistakes). This amounts to an overall error reduction of 4.23%. In table 2 and 3 the 6 most frequent tagging errors are listed for tagging using the standard and extended model respectively. Mistakes where `verb(pl)` is mixed up with `verb(inf)` sum up to 241 instances (6.11% of all mistakes) when using the standard model, as opposed to 82 cases (2.17%) using the extended model, an error reduction of 65.98%.

5.2.2 "Wotan" experiment

The results on the Wotan data can be seen in table 4. Using the standard model, accuracy is 92.05% (5715 mistakes). This result is very simi-

baseline accuracy	85.9%	
model	standard	extended
bigram accuracy	92.49%	92.94%
trigram accuracy	**93.34%**	**93.62%**
errors	3946	3779
error reduction	167 = **4.23%**	
pl/inf errors	241 (6.11%)	82 (2.17%)
pl/inf error red.	159 = **65.98%**	

Table 1: Tagging results on Alpino data

freq	assigned	correct
159	**verb(inf)**	**verb(pl)**
82	**verb(pl)**	**verb(inf)**
68	proper_name(both)	1-proper_name(both)
57	proper_name(both)	noun(de,sg)
53	verb(psp)	adjective(no_e,adv)
45	proper_name(both)	2-proper_name(both)

Table 2: Most frequent tagging mistakes on Alpino data, using standard model

lar to the 92.06% reported by Van Halteren, Zavrel and Daelemans in (van Halteren et al., 2001) who used the TnT trigram tagger (Brants, 2000) on the same training and testing data. Using the extended model, accuracy is 92.26% (5564 mistakes). This amounts to an overall error reduction of 2.64%. Mistakes where the plural verb is mixed up with the infinitive sum up to 316 instances (5.53% of all mistakes) when using the standard model, as opposed to 199 cases (3.58%) using the extended model, an error reduction of 37.03%.

5.3 Discussion of results

Extending the standard trigram tagging model with syntactical information aimed at resolving the most frequent type of tagging error led to a considerable reduction of this type of error in stand-alone POS tagging experiments on two different data sets. At the same time, other types of errors were also reduced.

freq	assigned	correct
69	proper_name(both)	1-proper_name(both)
57	proper_name(both)	noun(de,sg)
53	**verb(inf)**	**verb(pl)**
47	verb(psp)	adjective(no_e,adv)
45	proper_name(both)	2-proper_name(both)
42	punct(ligg_streep)	skip

Table 3: Most frequent tagging mistakes on Alpino data, using extended model

baseline accuracy	84.3%	
model	standard	extended
bigram accuracy	91.45%	91.73%
trigram accuracy	**92.05%**	**92.26%**
errors	5715	5564
error reduction	151 = **2.64%**	
pl/inf errors	316 (5.53%)	199 (3.58%)
pl/inf error red.	117 = **37.03%**	

Table 4: Tagging results on Wotan data

The relative error reduction for the specific type of error involving finite and infinite verb forms is almost twice as high in the case of the Alpino data as in the case of the Wotan data (respectively 65.98% and 37.03%). There are at least two possible explanations for this difference.

The first is a difference in tagsets. Although the Wotan tagset is much smaller than the Alpino tagset, the former features a more detailed treatment of verbs. In the Alpino data, the difference between plural finite verb forms and nonfinite verb forms is represented through just two tags. In the Wotan data, this difference is represented by 20 tags. An extended model that predicts which of the two forms should be used in a given situation is therefore more complex in the case of the Wotan data.

A further important difference between the two data sets is the available amount of training data (25 million words for the Alpino experiment compared to 628 thousand words for the Wotan experiment). In general a stochastic model such as the HMM will become more accurate when more training data is available. The Wotan experiment was repeated with increasing amounts of training data, and the results indicated that using more data would improve the results of both the standard and the extended model. The advantage of the extended model over the standard model increases slightly as more data is available, suggesting that the extended model would benefit more from extra data than the standard model.

6 Conclusion and future work

This work has presented how the HMM for POS tagging was extended with global contextual information without increasing the number of parameters beyond practical limits. Two tagging experiments, using a model extended with a binary feature concerning the occurrence of finite verb forms, resulted in improved accuracies compared to using the standard model. The annotation of the training data with context labels was acquired automatically through the use of a wide-coverage parser.

The tagger described here is used as a POS tag filter in wide-coverage parsing of Dutch (Prins and van Noord, 2004), increasing parsing efficiency as fewer POS tags have to be considered. In addition to reducing lexical ambiguity, it would be interesting to see if structural ambiguity can be reduced. In the approach under consideration, the tagger supplies the parser with an initial syntactic structure in the form of a partial bracketing of the input, based on the recognition of larger syntactic units or 'chunks'. Typically chunk tags will be assigned on the basis of words and their POS tags. An alternative approach is to use an extended model that assigns chunk tags and POS tags simultaneously, as was done for finite verb occurrence and POS tags in the current work. In this way, relations between POS tags and chunk tags can be modeled in both directions.

Another possible application is tagging of German. German features different cases, which can lead to problems for statistical taggers. This is illustrated in (Hinrichs and Trushkina, 2003) who point out that the `TnT` tagger wrongly assigns nominative case instead of accusative in a given sentence, resulting in the unlikely combination of two nominatives. The preference for just one assignment of the nominative case might be learned by including case information in the model.

Acknowledgements. This research was carried out as part of the PIONIER Project *Algorithms for Linguistic Processing*, funded by NWO (Dutch Organization for Scientific Research) and the University of Groningen. I would like to thank Hans van Halteren for supplying the Eindhoven corpus data set as used in (van Halteren et al., 2001).

References

J. Berghmans. 1994. Wotan, een automatische grammatikale tagger voor het Nederlands. Master's thesis, Dept. of Language and Speech, University of Nijmegen.

Gosse Bouma, Gertjan van Noord, and Robert Malouf. 2001. Wide coverage computational analysis of Dutch. In Walter Daelemans, Khalil Sima'an, Jorn Veenstra, and Jakub Zavrel, editors, *Computational Linguistics in the Netherlands, CLIN 2000*, pages 45–59, Amsterdam. Rodopi.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, WA.

E. Charniak, G. Carroll, J. Adcock, A. Cassandra, Y. Gotoh, J. Katz, M. Littman, and J. McCann. 1996. Taggers for parsers. *Artificial Intelligence*, 85(1-2):45–57.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, Pennsylvania.

Jan Daciuk. 1999. Treatment of unknown words. In *Proceedings of the Workshop on Implementing Automata WIA'99*, pages IX–1 – IX–9, Potsdam, Germany, July.

Erhard W. Hinrichs and Julia Trushkina. 2003. N-gram and PCFG models for morpho-syntactic tagging of German. In *Proceedings of The 2nd Workshop on Treebanks and Linguistic Theories (TLT 2003)*, pages 81–92, Växjö, Sweden, November.

Frederick Jelinek. 1998. *Statistical Methods for Speech Recognition*. MIT Press.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge Mass.

Robbert Prins and Gertjan van Noord. 2004. Reinforcing parser preferences through tagging. *Traitement Automatique des Langues (TAL), special issue on Evolutions in Parsing*. Accepted for publication, 2004.

H. van Halteren, J. Zavrel, and W. Daelemans. 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27(2):199–230.