# POS Multi-tagging based on Combined Models

**Yan Zhao, Gertjan van Noord**

University of Groningen
Groningen, the Netherland
yan.zhao@rug.nl, vannoord@let.rug.nl

**Abstract**

In the POS tagging task, there are two kinds of statistical models: one is generative model, such as the HMM, the others are discriminative models, such as the Maximum Entropy Model (MEM). POS multi-tagging decoding method includes the N-best paths method and forward-backward method. In this paper, we use the forward-backward decoding method based on a combined model of HMM and MEM. If P(t) is the forward-backward probability of each possible tag t, we first calculate P(t) according HMM and MEM separately. For all tags options in a certain position in a sentence, we normalize P(t) in HMM and MEM separately. Probability of the combined model is the sum of normalized forward-backward probabilities P_norm(t) in HMM and MEM. For each word w, we select the best tag in which the probability of combined model is the highest. In the experiments, we use combined model and get higher accuracy than any single model on POS tagging tasks of three languages, which are Chinese, English and Dutch. The result indicates that our combined model is effective.

## 1. Motivation

Being different from POS single-tagging, POS multi-tagging can assign more than one single best POS tag to a word in a sentence, according to the rank of probability of each tag calculated by a certain statistical model. A common usage of POS multi-tagging is a pre-processing part for a parser to increase the accuracy in comparison with single-tagging.

Is single-tagging or multi-tagging suitable for parser? It depends on the kind of parser. In the experiments of PCFG parsing (Charniak and Carroll, 1996) and RASP parser(Watson, 2006), single-tagging is preferable to a multi-tagging, because multi-tagging provides only a minor improvement in accuracy, but with a significant loss in efficiency. On the contrary, for a parser based on highly lexicalized grammars, such as CCG parser and Alpino parser(Prins and van Noord, 2001), the accuracy of the single-tagging is only about 92% to 94% due to the large number of tags (hundreds of or thousands of tags), far below the current 97% accuracy in English POS tagging. Multi-tagger has been shown to be quite necessary in such two parsers. For other language, such as Chinese, the POS tagging is still not good enough due to the relatively small size training corpus and different annotation guidelines, so the multi-tagging is also promising for some further NLP applications.

POS tagging is one of the best-studied applications in the statistical NLP domain. There are two kinds of statistical models: one is generative model, such as HMM(Brants, 2000), and the other is discriminative model, such as Maximum Entropy(ME) model(Ratnaparkhi, 1996). In multi-tagging task, (Prins and van Noord, 2001) used forward-backward method based on HMM in Dutch corpus, and (Curran et al., 2006) used the same forward-backward method based on ME model. In this paper, for POS multi-tagging task, we test N-best paths and forward-backward method on three languages separately, and combined HMM and ME model based on forward-backward method.

In methodology, we firstly introduce HMM and MEM briefly; Then, describe the two decoding methods: N-best paths and forward-backward method; lastly, we give the detail about how to combine HMM and ME model based on forward-backward frame. In the experiment section, I compare four kinds of multi-tagging methods based on HMM and MEM. The last section is conclusion.

## 2. Methodology

### 2.1. HMM and MEM

POS tagging may be described as a decoding process of a noisy-channel. A sequence of POS tags , which is generated by a source with probability, is transmitted through a noisy channel. The output of the channel is a sequence of words with conditional probability . POS tagging need to covert output word sequence into the original input tag sequence . This task can be accomplished by finding that maximizes the probability .

$$\hat{T} = \underset{T}{\operatorname{argmax}} P(T|W) \tag{1}$$

Usually, There is not enough corpus in which we can estimate the probability directly, So Bayes theorem is applied to swap the order of dependence between the tag sequence $T$ and the word sequence $W$.

$$P(T|W) = \frac{P(T,W)}{P(W)} = \frac{P(W|T)P(T)}{P(W)} \tag{2}$$

Eliminating the normalizing constant$P(W)$ , the decoding is equivalent to

$$\hat{T} = \underset{T}{\operatorname{argmax}} P(W|T)P(T) \tag{3}$$

$P(W|T)$ can be calculated by the state-specific observation probability. $P(T)$ can be estimated as the product of transition probability, as defined in formula (4):

$$P(T) = P(t_1, \cdots, t_{i-1}) \prod_{i=n}^{N} P(t_i|t_{i-n+1}, \cdots, t_{i-1}) \tag{4}$$

When $n$ equals 2 or 3, we obtain bigram or trigram model. In HMM, we break up the tag sequence $T$ by multiplication rule, we can also break up the formula (2) and rewrite it to the formula (5) if we decode the sequence from left to right

$$\hat{T} = \underset{T}{\text{argmax}} P(T|W) \approx \prod_{i=2}^{N} P(t_i|t_{1,\cdots,i-1}, W) \qquad (5)$$

Next problem is how to calculate conditional probability. We can limit the scope because $t_i$ depends mainly on the words and tags around it. So we can simplify $P(t_i|t1, \cdots, i-1, W)$ to $P(t_i|c_i)$, where $c_i$ denote the context information around $t_i$. For example, $c_i$ can be a set $c_i = w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, t_{i-2}, t_{i-1}$. In this way, we change sequence decoding problem into a series of classification problem at which a discrimative model can be used . In this paper, is calculated in MEM by formula (6)

$$P(t_i|c_i) = \frac{1}{Z(c_i)} exp\left(\sum_j \lambda_j f_j(c_i, t_i)\right) \qquad (6)$$

Where $Z(c_i)$ is normalization constant. $f_j(c_i, t_i)$ represents the $jth$ feature function in a set of features. Feature function $f_j$ is a Boolean function, and each $f_j$ corresponds to exactly one parameter $\lambda_j$ which can be viewed as a weight of $f_j$. When feature function $f_j = 1$ , $\lambda_j$ is used to predict value of $P(t_i|c_i)$.

### 2.2. N-best paths and forward-backward decoding methods

There are two multi-tagging decoding methods. One is to find N-best paths in a trellis,all the POS tags which are not on the N-best paths will be removed. As shown in Figure 1.

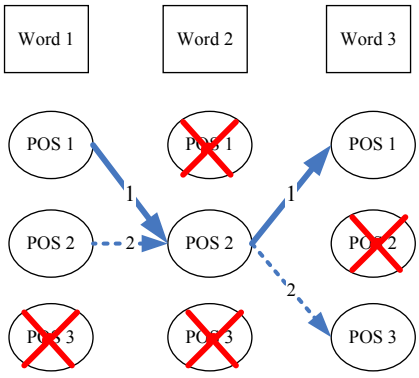


Figure 1: Best-path of method

The other is to use forward-backward method to rank all possible tags of the word in the certain position of a sentence and remove the unlikely ones according to a threshold value(Prins and van Noord, 2001). As shown in Figure 2. Despite the differences between HMM and ME model, in implementation, they all need to build a trellis which includes nodes, each node denotes a possible tag. Supposed there are $M$ tags in a POS tag set, and a sentence is comprised of $N$ words. Symbol $t_i^j$, where $1 \le i \le N, 1 \le j \le M$, denotes the $ith$ word in the sentence, $jth$ possible POS tag. As shown in Figure 3.

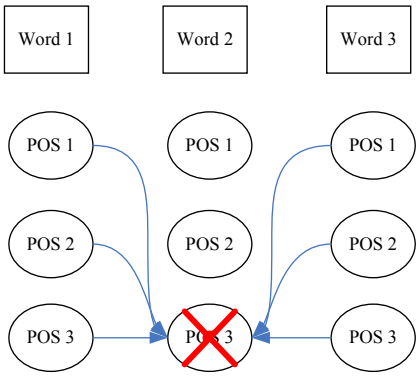


Figure 2: Forward-backward method

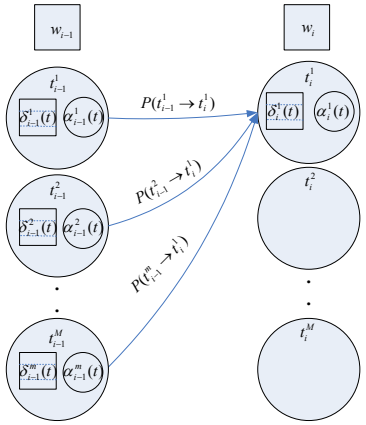

Figure 3: Implementation of a trellis

In Figure 3, $P(t_{i-1}^j \rightarrow t_i^j) = P(t_i^j|t_{i-1}^j)P(w_i|t_i^j)$ in HMM. In ME model, it can be defined by $P(t_{i-1}^j \rightarrow t_i^j) = P(t_i^j|c_i)$; And $P(t_i^j|c_i)$ can be calculated by $P(t_i^j|c_i) = 1/Z(c_i)\exp(\sum_k \lambda_k f_k(c_j, t_i^j))$. With the help of the unifying definition $P(t_{i-1}^j \rightarrow t_i^j)$, we illustrate two decoding methods in both HMM and ME model. The N-best paths can be defined by

$$\delta_i^1 = \max\_N_{1 \le j \le M}\left(\delta_{i-1}^j(t) P(t_{i-1}^j \rightarrow t_i^j)\right) \qquad (7)$$

Where $\delta$ is N-best values list. max_N means to get the N best values in a set. In each node, we need to keep N-best values and corresponding paths up to this node. If $\delta$ includes only one best value, it is viterbi algorithm, here our N-best paths can be thought as N-best viterbi algorithm.

In forward-backward method, we need to keep value of forward-backward probability in each node, In Figure 3, $\alpha_i^1(t)$ is computed by summing over all the probabilities of every path that could lead us to this node from left to right, it is defined as below.

$$\alpha_i^1(t) = \sum_{j=1}^{M} \alpha_{i-1}^j(t) P(t_{i-1}^j \rightarrow t_i^1) \qquad (8)$$

When we calculate from the right to left, we can get backward probability.

$$\beta_i^1(t) = \sum_{j=1}^{M} P(t_i^1 \rightarrow t_{i+1}^j)\beta_{i+1}^j(t) \quad (9)$$

For $w_i$, we can calculate each possible tag $t_i^j$ by $P(t_i^j) = \alpha_i^j(t)\beta_i^j(t)$, if $P(t_i^{max})$ is the maximum probability, and $P(t_i^j)/P(t_i^{max}) < \tau$, where $1 \leq j \leq M$, $t_i^j$ will be deleted and $\tau$ is a threshold value.Practically, we use log to avoid underflow of calculation. The more detail about viterbi and forward-backward algorithm can be found in the book(Jurafsky and Martin, 2008).

### 2.3. Combined Models

With the trellis and the unifying definition $P(t_i^j \rightarrow t_{i+1}^j)$, we can implement the forward-backward method based on other statistical models. In this paper, we get the HMM and MEM together based on forward-backward method. $P_{HMM}(t_i^j)$ is forward-backward probability of node $t_i^j$ calculated by HMM and $P_{ME}(t_i^j)$ is forward-backward probabilities of node $t_i^j$ calculated by MEM, we need normalize these probability before we combine them.

$$P_{NOR\_HMM}(t_i^j) = \frac{P_{HMM}(t_i^j)}{\sum_{j=1}^{M} P_{HMM}(t_i^j)} \quad (10)$$

$$P_{COMBINED}(t_i^j) = P_{NOR\_HMM}(t_i^j) + P_{NOR\_MEM}(t_i^j) \quad (11)$$

After we get $P_{COMBINED}(t_i^j)$, we can use the threshold value to delete the unlikely tags, as described previously.

## 3. Experiment

### 3.1. Corpora

In the experiments, we test POS multi-tagging on three kinds of languages: Chinese, English and Dutch. Table 1 gives general information about three Corpora.

| Lang. | name | Num. |
|---|---|---|
| **Chinese** | People's Daily | 43 |
| **English** | Brown Corpus | 165 |
| **Dutch** | News papers | 2316 |

Table 1: Training corpora

For Chinese and English, we divided the corpus with proportion 8:2 roughly from beginning to end to create training and testing corpus. The number of tags comes from training corpus. Considering that we are only interesting in the result of comparison of different methods, not the specific accuracy, we didn't consider unknown word problem. That is to say, if a word in test didn't appear in training corpus, I will give it right tag directly. For unknown word problem, MEM will be better than HMM because it is able to integrate more lexical features.

### 3.2. Implementation

In HMM, we use trigram and linear interpolation smoothing methods. In N-best paths method, we can keep N-best paths for each sentence. If the sentence includes $M$ words, the last result will be $M + N$ tags for the sentece. Other way is that for each path, we can compare it with the best path value, if comparison is smaller than a given threshold value $\tau$, we will add the path into the last result. In our experiment, the second way is better than the first one. Our last result was obtained by the second way. In MEM, The value of $\lambda_j$ is trained by L-BFGS method(Malouf, 2002) and Gaussian prior(Chen and Rosenfeld, 1999) to fight against overfitting problem. We just use forward-backward method decoding method because we found that forward-backward method is better than N-best paths in HMM. And we tried the different Gaussian prior and iteration time, the results in Table 2, 3 and 4 is the best result we acquired . For English and Chinese, we use the $c_i = w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, t_{i-2}, t_{i-1}$ as a template; for Dutch corpus, we use $c_i = w_{i-1}, w_i, w_{i+1}, w_{i,i+1}, w_{i+2}, t_{i-2}, t_{i-1}$ as a template.

### 3.3. result

We test four methods. In Table 2, 3 and 4, FB-HMM is abbreviation of forward-backward method in HMM. NB-HMM is abbreviation of N-Best paths method in HMM, and FB-MEM is abbreviation of forward-backward method in MEM, and the last, we gave the result of forward-backward method based on combined models

The ratio of tags to words is listed on the first row of each table. We can see that forward-backward method is better than N-best paths in English and Chinese task. A little surprisingly, when tags/words equals 1, that is a single-tagging task, forward-backward method surpasses the viterbi method in precision in Chinese and English language too. In Dutch task, there are some exceptions that are indicated by italic format in Table 4, because Dutch language contains some multi-word-units. An example of multi-word-unit is listed in Table 5. The tag *2/3-Noun(both,pl,[])* is the only one answer if the previous tag is *1/3-Noun(both,pl,[])*, the N-best paths method can recognize the multi-word-units better than forward-backward method under this circumstance.

| Noun(both,pl,[]) | Example |
|---|---|
| 1/3-Noun(both,pl,[]) | Van |
| 2/3-Noun(both,pl,[]) | der |
| 3/3-Noun(both,pl,[]) | Valk-hotels |

Table 5: An example of Multi-Word-Unit

In all three languages, the best result comes from forward-backward method based on combined models (ME and HMM).

## 4. Conclusion

As we expected, MEM is better than HMM in accuracy, this has been approved in single-tagging problem, and we get the same conclusion in multi-tagging problem. As a basic decoding method, for multi-tagging task, forward-backward method is better in precision than N-best paths method. Another advantage of forwad-backward method lies on that it is more convenient to combine many models.

| tags/words | 1 | 1.03345 | 1.07577 | 1.14673 | 1.22969 | 1.92443 |
|---|---|---|---|---|---|---|
| **FB_HMM** | 96.9523 | 98.1036 | 98.7321 | 99.1329 | 99.3547 | 100 |
| **NB_HMM** | 96.9496 | 98.0904 | 98.7231 | 99.1263 | 99.3469 | 100 |
| **FB_MEM** | 97.0431 | 98.2003 | 98.814 | 99.2054 | 99.3754 | 100 |
| **Combined** | 97.2587 | 98.3602 | 98.9345 | 99.2577 | 99.3953 | 100 |

Table 2: Multi-tagging result of English

| tags/words | 1 | 1.04787 | 1.07358 | 1.14887 | 1.32054 | 1.66062 |
|---|---|---|---|---|---|---|
| **FB_HMM** | 95.8076 | 97.5626 | 98.169 | 99.0516 | 99.4395 | 100 |
| **NB_HMM** | 95.7743 | 97.5297 | 98.1365 | 98.9984 | 99.2653 | 100 |
| **FB_MEM** | 95.4741 | 98.102 | 98.574 | 99.1568 | 99.4427 | 100 |
| **Combined** | 96.6788 | 98.2198 | 98.6674 | 99.2311 | 99.457 | 100 |

Table 3: Multi-tagging result of Chinese

| tags/words | 1 | 1.03448 | 1.07449 | 1.11595 | 1.38231 | 2.20224 |
|---|---|---|---|---|---|---|
| **FB_HMM** | 93.5223 | 94.7586 | 95.6587 | 96.2628 | 97.8698 | 100 |
| **NB_HMM** | *93.5347* | 94.7337 | *95.6618* | *96.2691* | 97.8107 | 100 |
| **FB_MEM** | 93.5378 | 94.8427 | 95.914 | 96.5836 | 98.3557 | 100 |
| **Combined** | 93.8617 | 95.1168 | 96.1383 | 96.8328 | 98.3993 | 100 |

Table 4: Multi-tagging result of Dutch

In this paper, we introduce how to get HMM and MEM together. In fact, you can combine multiple models. If you need higher speed and more storage efficiency, you can use HMM model as a primary one, for the ambitious words that HMM can not handle properly, build some light-weight discriminate models to deal with and get them together.

## 5. Acknowledgements

People's Daily Newspaper of January 1998, which we used as Chinese POS corpora in this paper, came from Institute of Computational Linguistic of Peking of University.

## 6. References

Thorsten Brants. 2000. Tnt– a statistical part-of-speech tagger. In *Proceeding of the Sixth Applied Natural Language Processing Conference*.

Eugene Charniak and Clenn Carroll. 1996. Taggers for parsers. *Artificial Intelligence*.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University.

James R. Curran, Stephen Clark, and David Vadas. 2006. Multi-tagging for lexicalized-grammar parsing. In *ACL-44:The 44th annual meeting of the Association for Computational Linguistics*.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Conference on Computational Natural Language Learning (CoNLL)*.

Robbert Prins and Gertjan van Noord. 2001. Unsupervised pos-tagging improves parsing accuracy and parsing efficiency. In *Proceedings of IWPT*.

Adwait Ratnaparkhi. 1996. A maximum entropy model of part-of-speech tagging. In *Proceeding EMNLP*.

Rebecca Watson. 2006. Part-of-speech tagging models for parsing. In *Proceeding of the Computational Linguistics in the UK Conference*.