Lexicalized Hidden Markov Models for Part-of-Speech Tagging

Sang-Zoo Lee and **Jun-ichi Tsujii**
Department of Information Science
Graduate School of Science
University of Tokyo, Hongo 7-3-1
Bunkyo-ku, Tokyo 113, Japan
{lee,tsujii}@is.s.u-tokyo.ac.jp

Hae-Chang Rim
Department of Computer Science
Korea University
1 5-Ga Anam-Dong, Seongbuk-Gu
Seoul 136-701, Korea
rim@nlp.korea.ac.kr

Abstract

Since most previous works for HMM-based tagging consider only part-of-speech information in contexts, their models cannot utilize lexical information which is crucial for resolving some morphological ambiguity. In this paper we introduce uniformly lexicalized HMMs for part-of-speech tagging in both English and Korean. The lexicalized models use a simplified back-off smoothing technique to overcome data sparseness. In experiments, lexicalized models achieve higher accuracy than non-lexicalized models and the back-off smoothing method mitigates data sparseness better than simple smoothing methods.

1 Introduction

Part-of-speech(POS) tagging is a process in which a proper POS tag is assigned to each word in raw texts. Even though morphologically ambiguous words have more than one POS tag, they belong to just one tag in a context. To resolve such ambiguity, taggers have to consult various sources of information such as lexical preferences (e.g. without consulting context, *table* is more probably a noun than a verb or an adjective), tag n-gram contexts (e.g. after a non-possessive pronoun, *table* is more probably a verb than a noun or an adjective, as in *they table an amendment*), word n-gram contexts (e.g. before *lamp*, *table* is more probably an adjective than a noun or a verb, as in *I need a table lamp*), and so on(Lee et al., 1999).

However, most previous HMM-based taggers consider only POS information in contexts, and so they cannot capture lexical information which is necessary for resolving some morphological ambiguity. Some recent works have reported that tagging accuracy could be improved by using lexical information in their models such as the transformation-based patch rules(Brill, 1994), the maximum entropy model(Ratnaparkhi, 1996), the statistical lexical rules(Lee et al., 1999), the HMM considering multi-words(Kim, 1996), the selectively lexicalized HMM(Kim et al., 1999), and so on. In the previous works(Kim, 1996)(Kim et al., 1999), however, their HMMs were lexicalized selectively and restrictively.

In this paper we propose a method of uniformly lexicalizing the standard HMM for part-of-speech tagging in both English and Korean. Because the sparse-data problem is more serious in lexicalized models than in the standard model, a simplified version of the well-known back-off smoothing method is used to overcome the problem. For experiments, the Brown corpus(Francis, 1982) is used for English tagging and the KUNLP corpus(Lee et al., 1999) is used for Korean tagging. The experimental results show that lexicalized models perform better than non-lexicalized models and the simplified back-off smoothing technique can mitigate data sparseness better than simple smoothing techniques.

2 The "standard" HMM

We basically follow the notation of (Charniak et al., 1993) to describe Bayesian models. In this paper, we assume that $\{w^1, w^2, \ldots, w^\omega\}$ is a set of words, $\{t^1, t^2, \ldots, t^\tau\}$ is a set of POS tags, a sequence of random variables $W_{1,n} = W_1 W_2 \ldots W_n$ is a sentence of n words, and a sequence of random variables $T_{1,n} = T_1 T_2 \ldots T_n$ is a sequence of n POS tags. Because each of random variables W can take as its value any of the words in the vocabulary, we denote the value of W_i by w_i and a particular sequence of values for $W_{i,j}$ $(i \leq j)$ by $w_{i,j}$. In a similar way, we denote the value of T_i by t_i and a particular

sequence of values for $T_{i,j}$ $(i \leq j)$ by $t_{i,j}$. For generality, terms $w_{i,j}$ and $t_{i,j}$ $(i > j)$ are defined as being empty.

The purpose of Bayesian models for POS tagging is to find the most likely sequence of POS tags for a given sequence of words, as follows:

$$T(w_{1,n}) = \underset{t_{1,n}}{\operatorname{argmax}} \Pr(T_{1,n} = t_{1,n} \mid W_{1,n} = w_{1,n})$$

Because reference to the random variables themselves can be omitted, the above equation becomes:

$$T(w_{1,n}) = \underset{t_{1,n}}{\operatorname{argmax}} \Pr(t_{1,n} \mid w_{1,n}) \quad (1)$$

Now, Eqn. 1 is transformed into Eqn. 2 since $\Pr(w_{1,n})$ is constant for all $t_{1,n}$.

$$\begin{aligned} T(w_{1,n}) &= \underset{t_{1,n}}{\operatorname{argmax}} \frac{\Pr(t_{1,n}, w_{1,n})}{\Pr(w_{1,n})} \\ &= \underset{t_{1,n}}{\operatorname{argmax}} \Pr(t_{1,n}, w_{1,n}) \quad (2) \end{aligned}$$

Then, the probability $\Pr(t_{1,n}, w_{1,n})$ is broken down into Eqn. 3 by using the chain rule.

$$\Pr(t_{1,n}, w_{1,n}) = \prod_{i=1}^{n} \left(\begin{array}{l} \Pr(t_i \mid t_{1,i-1}, w_{1,i-1}) \\ \times \Pr(w_i \mid t_{1,i}, w_{1,i-1}) \end{array} \right) \quad (3)$$

Because it is difficult to compute Eqn. 3, the standard HMM simplified it by making a strict Markov assumption to get a more tractable form.

$$\Pr(t_{1,n}, w_{1,n}) \approx \prod_{i=1}^{n} \left(\begin{array}{l} \Pr(t_i \mid t_{i-K,i-1}) \\ \times \Pr(w_i \mid t_i) \end{array} \right) \quad (4)$$

In the standard HMM, the probability of the current tag t_i depends on only the previous K tags $t_{i-K,i-1}$ and the probability of the current word w_i depends on only the current tag[1]. Therefore, this model cannot consider lexical information in contexts.

3 Lexicalized HMMs

In English POS tagging, the tagging unit is a word. On the contrary, Korean POS tagging prefers a morpheme[2].

[1]Usually, K is determined as 1 (bigram as in (Charniak et al., 1993)) or 2 (trigram as in (Merialdo, 1991)).

[2]The main reason is that the number of word-unit tags is not finite because Korean words can be freely and newly formed by agglutinating morphemes(Lee et al., 1999).

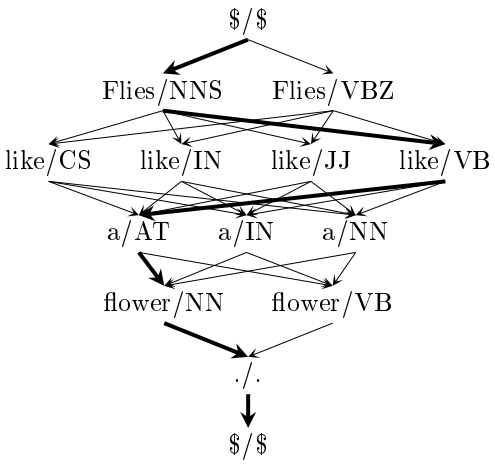


Figure 1: A word-unit lattice of "Flies like a flower ."

Figure 1 shows a word-unit lattice of an English sentence, "Flies like a flower.", where each node has a word and its word-unit tag. Figure 2 shows a morpheme-unit lattice of a Korean sentence, "*NeoNeun Hal Su issDa.*", where each node has a morpheme and its morpheme-unit tag. In case of Korean, transitions across a word boundary, which are depicted by a solid line, are distinguished from transitions within a word, which are depicted by a dotted line. In both cases, sequences connected by bold lines indicate the most likely sequences.

3.1 Word-unit models

Lexicalized HMMs for word-unit tagging are defined by making a less strict Markov assumption, as follows:

$$\begin{aligned} \Lambda(T_{(K,J)}, W_{(L,I)}) &\models \Pr(t_{1,n}, w_{1,n}) \\ &\approx \prod_{i=1}^{n} \left(\begin{array}{l} \Pr(t_i \mid t_{i-K,i-1}, w_{i-J,i-1}) \\ \times \Pr(w_i \mid t_{i-L,i}, w_{i-I,i-1}) \end{array} \right) \quad (5) \end{aligned}$$

In models $\Lambda(T_{(K,J)}, W_{(L,I)})$, the probability of the current tag t_i depends on both the previous K tags $t_{i-K,i-1}$ and the previous J words $w_{i-J,i-1}$ and the probability of the current word w_i depends on the current tag and the previous L tags $t_{i-L,i}$ and the previous I words $w_{i-I,i-1}$. So, they can consider lexical information. In experiments, we set K as 1 or 2, J as 0 or K, L as 1 or 2, and I as 0 or L. If J and I are zero, the above models are non-lexicalized models. Otherwise, they are lexicalized models.

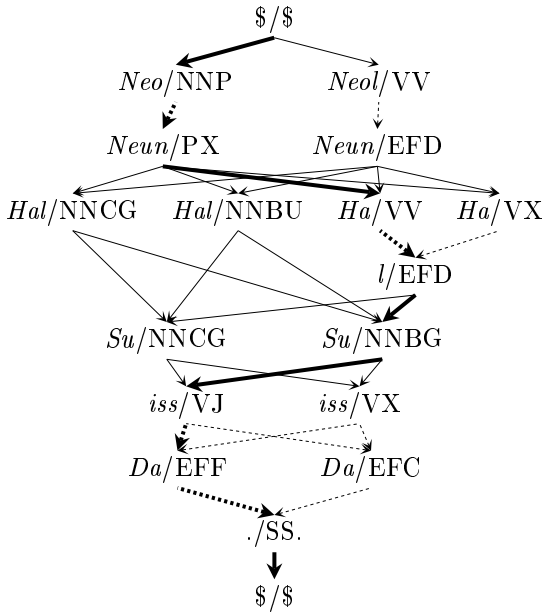


Figure 2: A morpheme-unit lattice of "*NeoNeun Hal Su issDa.*"(= You can do it.)

In a lexicalized model $\Lambda(T_{(2,2)}, W_{(2,2)})$, for example, the probability of a node "a/AT" of the most likely sequence in Figure 1 is calculated as follows:

$$\Pr(AT \mid NNS, VB, Flies, like) \times \Pr(a \mid AT, NNS, VB, Flies, like)$$

3.2 Morpheme-unit models

Bayesian models for morpheme-unit tagging find the most likely sequence of morphemes and corresponding tags for a given sequence of words, as follows:

$$T(w_{1,n}) = \underset{c_{1,u}, m_{1,u}}{\operatorname{argmax}} \Pr(c_{1,u}, m_{1,u} \mid w_{1,n}) \quad (6)$$

$$\approx \underset{c_{1,u}, m_{1,u}}{\operatorname{argmax}} \Pr(c_{1,u}, p_{2,u}, m_{1,u}) \quad (7)$$

In the above equations, $u(\geq n)$ denotes the number of morphemes in a sequence corresponding the given word sequence, c denotes a morpheme-unit tag, m denotes a morpheme, and p denotes a type of transition from the previous tag to the current tag. p can have one of two values, "#" denoting a transition across a word boundary and "+" denoting a transition within a word. Because it is difficult to calculate Eqn. 6, the word sequence term $w_{1,n}$ is usually ignored as in Eqn. 7. Instead, we introduce p in Eqn. 7 to consider word-spacing[3].

The probability $\Pr(c_{1,u}, p_{2,u}, m_{1,u})$ is also broken down into Eqn. 8 by using the chain rule.

$$\Pr(c_{1,u}, p_{2,u}, m_{1,u}) = \prod_{i=1}^{u} \left(\begin{array}{l} \Pr(c_i, p_i \mid c_{1,i-1}, p_{2,i-1}, m_{1,i-1}) \\ \times \Pr(m_i \mid c_{1,i}, p_{2,i}, m_{1,i-1}) \end{array} \right) \quad (8)$$

Because Eqn. 8 is not easy to compute, it is simplified by making a Markov assumption to get a more tractable form.

In a similar way to the case of word-unit tagging, lexicalized HMMs for morpheme-unit tagging are defined by making a less strict Markov assumption, as follows:

$$\Lambda(C_{[s](K,J)}, M_{[s](L,I)}) \models \Pr(c_{1,u}, p_{2,u}, m_{1,u}) \approx \prod_{i=1}^{u} \begin{array}{l} \Pr(c_i[, p_i] \mid \begin{array}{l} c_{i-K,i-1}[, p_{i-K+1,i-1}], \\ m_{i-J,i-1} \end{array}) \\ \times \Pr(m_i \mid c_{i-L,i}[, p_{i-L+1,i}], m_{i-I,i-1}) \end{array} \quad (9)$$

In models $\Lambda(C_{[s](K,J)}, M_{[s](L,I)})$, the probability of the current morpheme tag c_i depends on both the previous K tags $c_{i-K,i-1}$ (optionally, the types of their transition $p_{i-K+1,i-1}$) and the previous J morphemes $m_{i-J,i-1}$ and the probability of the current morpheme m_i depends on the current tag and the previous L tags $c_{i-L,i}$ (optionally, the types of their transition $p_{i-L+1,i}$) and the previous I morphemes $m_{i-I,i-1}$. So, they can also consider lexical information.

In a lexicalized model $\Lambda(C_{s(2,2)}, M_{(2,2)})$ where word-spacing is considered only in the tag probabilities, for example, the probability of a node "*Su*/NNBG" of the most likely sequence in Figure 2 is calculated as follows:

$$\Pr(NNBG, \# \mid VV, EFD, +, Ha, l) \times \Pr(Su \mid VV, EFD, NNBG, Ha, l)$$

3.3 Parameter estimation

In supervised learning, the simpliest parameter estimation is the maximum likelihood(ML) estimation(Duda et al., 1973) which maximizes the probability of a training set. The ML estimate of tag $(K+1)$-gram probability, $\Pr_{ML}(t_i \mid t_{i-K,i-1})$, is calculated as follows:

$$\underset{ML}{\Pr}(t_i \mid t_{i-K,i-1}) = \frac{\mathrm{Fq}(t_{i-K,i})}{\mathrm{Fq}(t_{i-K,i-1})} \quad (10)$$

[3]Most previous HMM-based Korean taggers except (Kim et al., 1998) did not consider word-spacing.

where the function $\mathrm{Fq}(x)$ returns the frequency of x in the training set. When using the maximum likelihood estimation, data sparseness is more serious in lexicalized models than in non-lexicalized models because the former has even more parameters than the latter.

In (Chen, 1996), where various smoothing techniques was tested for a language model by using the perplexity measure, a back-off smoothing(Katz, 1987) is said to perform better on a small traning set than other methods. In the back-off smoothing, the smoothed probability of tag $(K+1)$-gram $\Pr_{SBO}(t_i \mid t_{i-K,i-1})$ is calculated as follows:

$$\Pr_{SBO}(t_i \mid t_{i-K,i-1}) = \begin{cases} d_r \Pr_{ML}(t_i \mid t_{i-K,i-1}) & \text{if } r > 0 \\ \alpha(t_{i-K,i-1}) \Pr_{SBO}(t_i \mid t_{i-K+1,i-1}) & \text{if } r = 0 \end{cases} \quad (11)$$

$$\text{where } r = \mathrm{Fq}(t_{i-K,i}),\ r^* = (r+1)\frac{n_{r+1}}{n_r}$$

$$d_r = \frac{\frac{r^*}{r} - \frac{(r+1)\times n_{r+1}}{n_1}}{1 - \frac{(r+1)\times n_{r+1}}{n_1}}$$

n_r denotes the number of $(K+1)$-gram whose frequency is r, and the coefficient d_r is called the discount ratio, which reflects the Good-Turing estimate(Good, 1953)[4]. Eqn. 11 means that $\Pr_{SBO}(t_i \mid t_{i-K,i-1})$ is under-etimated by d_r than its maximum likelihood estimate, if $r > 0$, or is backed off by its smoothing term $\Pr_{SBO}(t_i \mid t_{i-K+1,i-1})$ in proportion to the value of the function $\alpha(t_{i-K,i-1})$ of its conditional term $t_{i-K,i-1}$, if $r = 0$.

However, because Eqn. 11 requires complicated computation in $\alpha(t_{i-K,i-1})$, we simplify it to get a function of the frequency of a conditional term, as follows:

$$\alpha(\mathrm{Fq}(t_{i-K,i-1}) = f) = \Delta \times \frac{\mathrm{E}[\mathrm{Fq}(t_{i-K,i-1}) = f]}{\sum_{f=0}^{\infty} \mathrm{E}[\mathrm{Fq}(t_{i-K,i-1}) = f]} \quad (12)$$

$$\text{where } \Delta = 1 - \sum_{t_{i-K,i}, r>0} \Pr_{SBO}(t_i | t_{i-K,i-1}),$$

$$\mathrm{E}[\mathrm{Fq}(t_{i-K,i-1}) = f] = \sum_{t_{i-K+1,i}, r=0, Fq(t_{i-K,i-1})=f} \Pr_{SBO}(t_i | t_{i-K+1,i-1})$$

In Eqn. 12, the range of f is bucketed into 7 regions such as $f = 0, 1, 2, 3, 4, 5$ and $f \geq 6$ since it is also difficult to compute this equation for all possible values of f.

Using the formalism of our simplified back-off smoothing, each of probabilities whose ML estimate is zero is backed off by its corresponding smoothing term. In experiments, the smoothing terms of $\Pr_{SBO}(t_i \mid t_{i-K,i-1}, w_{i-J,i-1})$ are determined as follows:

$$\begin{cases} \Pr_{SBO}(t_i \mid \begin{matrix} t_{i-K+1,i-1}, \\ w_{i-J+1,i-1} \end{matrix}) & \text{if } K \geq 1, J > 1 \\ \Pr_{SBO}(t_i \mid t_{i-K,i-1}) & \text{if } K \geq 1, J = 1 \\ \Pr_{SBO}(t_i \mid t_{i-K+1,i-1}) & \text{if } K \geq 1, J = 0 \\ \Pr_{AD}(t_i) & \text{if } K = 0, J = 0 \end{cases} \quad (13)$$

Also, the smoothing terms of $\Pr_{SBO}(w_i \mid t_{i-L,i}, w_{i-I,i-1})$ are determined as follows:

$$\begin{cases} \Pr_{SBO}(w_i \mid \begin{matrix} t_{i-L+1,i}, \\ w_{i-I+1,i-1} \end{matrix}) & \text{if } L \geq 1, I > 1 \\ \Pr_{SBO}(w_i \mid t_{i-L,i}) & \text{if } L \geq 1, I = 1 \\ \Pr_{SBO}(w_i \mid t_{i-L+1,i}) & \text{if } L \geq 1, I = 0 \\ \Pr_{SBO}(w_i) & \text{if } L = 0, I = 0 \\ \Pr_{AD}(w_i) & \text{if } L = -1, I = 0 \end{cases} \quad (14)$$

In Eqn. 13 and 14, the smoothing term of a unigram probability is calculated by using an additive smoothing with $\delta = 10^{-2}$ which is chosen through experiments. The equation for the additive smoothing(Chen, 1996) is as follows:

$$\Pr_{AD}(t_i \mid t_{i-K,i-1}) = \frac{\mathrm{Fq}(t_{i-K,i}) + \delta}{\sum_{t_i} (\mathrm{Fq}(t_{i-K,i}) + \delta)} \quad (15)$$

In a similar way, the smoothing terms of parameters in Eqn. 9 are determined.

3.4 Model decoding

From the viewpoint of the lattice structure, the problem of POS tagging can be regarded as the problem of finding the most likely path from the start node (\$/\$) to the end node (\$/\$). The Viterbi search algorithm(Forney, 1973), which has been used for HMM decoding, can be effectively applied to this task just with slight modification[5].

4 Experiments

4.1 Environment

In experiments, the Brown corpus is used for English POS tagging and the KUNLP corpus

[4] Katz said that $d_r = 1$ if $r > 5$.

[5] Such modification is explained in detail in (Lee, 1999).

	Brown	KUNLP
NW	1,113,189	167,115
NS	53,885	15,211
NT	82	65
DA	1.64	3.41
RUA	61.54%	26.72%

NW Number of words. **NS** Number of sentences. **NT** Number of tags (morpheme-unit tag for KUNLP). **DA** Degree of ambiguity (i.e. the number of tags per word). **RUA** Ratio of unambiguous words.

Table 1: Information about the Brown corpus and the KUNLP corpus

	Inside-test	Outside-test
ML	95.57	**94.97**
AD($\delta = 1$)	93.92	93.02
AD($\delta = 10^{-1}$)	95.02	94.79
AD($\delta = 10^{-2}$)	95.42	**95.08**
AD($\delta = 10^{-3}$)	95.55	95.05
AD($\delta = 10^{-4}$)	95.57	94.98
AD($\delta = 10^{-5}$)	95.57	94.94
AD($\delta = 10^{-6}$)	95.57	94.91
AD($\delta = 10^{-7}$)	95.57	94.89
AD($\delta = 10^{-8}$)	95.57	94.87
SBO	95.55	**95.25**

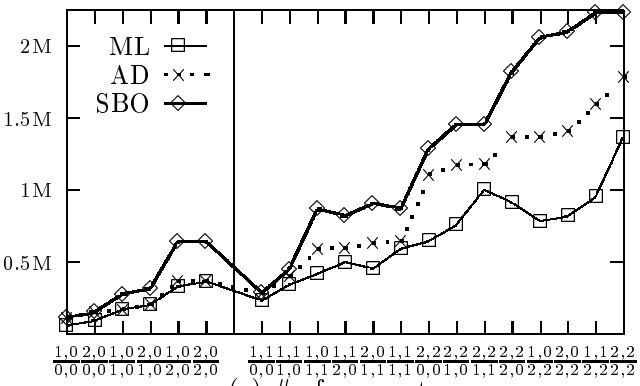
ML Maximum likelihood estimate (with simple smoothing). **AD** Additive smoothing. **SBO** Simplified back-off smoothing.

Table 2: Tagging accuracy of $\Lambda(C_{(1:0)}, M_{(0:0)})$

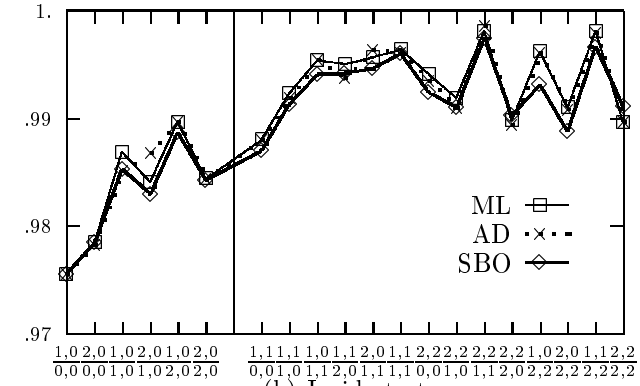
for Korean POS tagging. Table 1 shows some information about both corpora[6]. Each of them was segmented into two parts, the training set of 90% and the test set of 10%, in the way that each sentence in the test set was extracted from every 10 sentence. According to Table 1, Korean is said to be more difficult to disambiguate than English.

We assume "closed" vocabulary for English and "open" vocabulary for Korean since we do not have any English morphological analyzer consistent with the Brown corpus. Therefore, for morphological analysis of English, we just

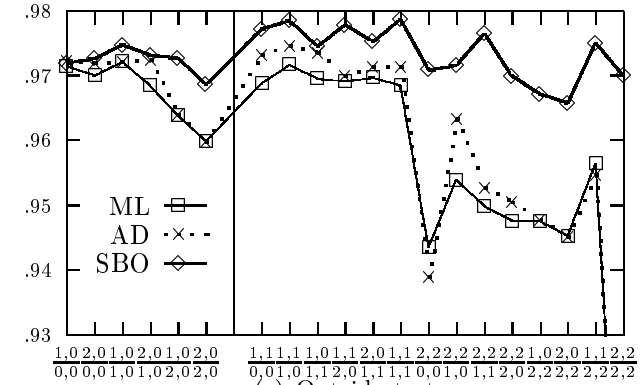
[6]Note that some sentences, which have composite tags(such as "HV+TO" in "*hafta*"), "ILLEGAL" tag, or "NIL" tag, were removed from the Brown corpus and tags with "*"(not) such as "BEZ*" were replaced by corresponding tags without "*" such as "BEZ".



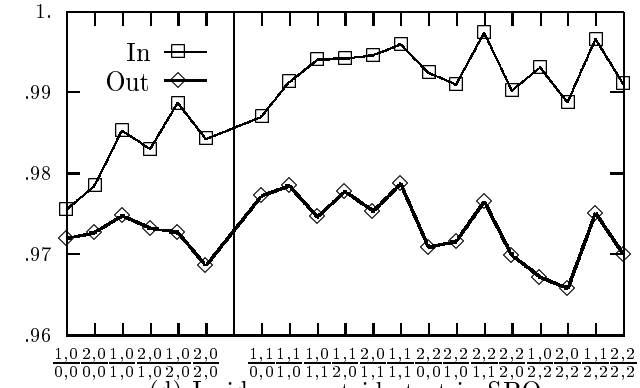
(a) # of parameters



(b) Inside-test



(c) Outside-test



(d) Inside vs. outside-test in SBO

Figure 3: Results of English tagging

looked up the dictionary tailored to the Brown corpus. In case of Korean, we have used a Korean morphological analyzer(Lee, 1999) which is consistent with the KUNLP corpus.

4.2 Results and evaluation

Table 2 shows the tagging accuracy of the simplest HMM, $\Lambda(C_{(1:0)}, M_{(0:0)})$, for Korean tagging, according to various smoothing methods[7]. Note that *ML* denotes a simple smoothing method where ML estimates with probability less than 10^{-9} are smoothed and replaced by 10^{-9}. Because, in the outside-test, $AD(\delta = 10^{-2})$ performs better than *ML* and $AD(\delta \neq 10^{-2})$, we use $\delta = 10^{-2}$ in our additive smoothing. According to Table 2, *SBO* performs well even in the simplest HMM.

Figure 3 illustrates 4 graphs about the results of English tagging: (a) the number of parameters in each model, (b) the accuracy of each model for the training set, (c) the accuracy of each model for the test set, and (d) the accuracy of each model with *SBO* for both training and test set. Here, labels in x-axis specify models in the way that $\frac{K,J}{L,I}$ denotes $\Lambda(T_{(K,J)}, W_{(L,I)})$. Therefore, the first 6 models are non-lexicalized models and the others are lexicalized models.

Actually, *SBO* uses more parameters than others. The three smoothing methods, *ML, AD, SBO*, perform well for the training set since the inside-tests usually have little data sparseness. On the other hand, for the unseen test set, the simple methods, *ML* and *AD*, cannot mitigate the data sparseness problem, especially in sophisticated models. However, our method *SBO* can overcome the problem, as shown in Figure 3(c). Also, we can see in Figure 3(d) that some lexicalized models achieve higher accuracy than non-lexicalized models. We can say that the best lexicalized model, $\Lambda(T_{(1,1)}, W_{(1,1)})$ using *SBO*, improved the simple bigram model, $\Lambda(T_{(1,0)}, W_{(0,0)})$ using *SBO*, from 97.19% to 97.87% (the error reduction ratio of 24.20%). Interestingly, some lexicalized models (such as $\Lambda(T_{(1,1)}, W_{(0,0)})$ and $\Lambda(T_{(1,1)}, W_{(1,0)})$), which have a relatively small number of parameters, perform better than non-lexicalized models in the case of outside-tests using *SBO*. Unfortunately, we cannot expect the result of outside-tests from that of inside-tests because there is no direct relation between them.

[7]Inside-test means an experiment on the training set itself and outside-test an experiment on the test set.

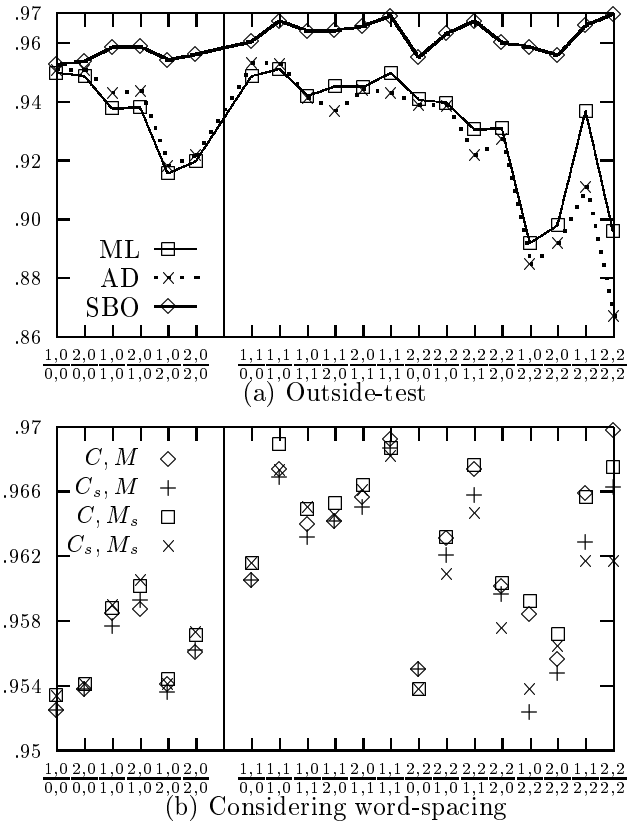


Figure 4: Results of Korean tagging

Figure 4 includes 2 graphs about the results of Korean tagging: (a) the outside accuracy of each model $\Lambda(C_{(K,J)}, M_{(L,I)})$ and (b) the outside accuracy of each model $\Lambda(C_{[s](K,J)}, M_{[s](L,I)})$ with/without considering word-spacing when using *SBO*. Here, labels in x-axis specify models in the way that $\frac{K,J}{L,I}$ denotes $\Lambda(C_{[s](K,J)}, M_{[s](L,I)})$ and, for example, C_s, M in (b) denotes $\Lambda(C_{s(K,J)}, M_{(L,I)})$.

As shown in Figure 4, the simple methods, *ML* and *AD*, cannot mitigate that sparse-data problem, but our method *SBO* can overcome it. Also, some lexicalized models perform better than non-lexicalized models. On the other hand, considering word-spacing gives good clues to the models sometimes, but yet we cannot say what is the best way. From the experimental results, we can say that the best model, $\Lambda(C_{(2,2)}, M_{(2,2)})$ using *SBO*, improved the previous models, $\Lambda(C_{(1,0)}, M_{(0,0)})$ us-

ing ML(Lee, 1995), and $\Lambda(C_{s(1,0)}, M_{(0,0)})$ using ML(Kim et al., 1998), from 94.97% and 95.05% to 96.98% (the error reduction ratio of 39.95% and 38.99%) respectively.

5 Conclusion

We have presented uniformly lexicalized HMMs for POS tagging of English and Korean. In the models, data sparseness was effectively mitigated by using our simplified back-off smoothing. From the experiments, we have observed that lexical information is useful for POS tagging in HMMs, as is in other models, and our lexicalized models improved non-lexicalized models by the error reduction ratio of 24.20% (in English tagging) and 39.95% (in Korean tagging).

Generally, the uniform extension of models requires rapid increase of parameters, and hence suffers from large storage and sparse data. Recently in many areas where HMMs are used, many efforts to extend models non-uniformly have been made, sometimes resulting in noticeable improvement. For this reason, we are trying to transform our uniform models into non-uniform models, which may be more effective in terms of both space complexity and reliable estimation of paremeters, without loss of accuracy.

References

E. Brill. 1994. Some Advances in Transformation-Based Part of Speech Tagging. In *Proc. of the 12th Nat'l Conf. on Artificial Intelligence(AAAI-94)*, 722-727.

E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz. 1993. Equations for Part-of-Speech Tagging. In *Proc. of the 11th Nat'l Conf. on Artificial Intelligence(AAAI-93)*, 784-789.

S. F. Chen. 1996. *Building Probabilistic Models for Natural Language*. Doctoral Dissertation, Harvard University, USA.

R. O. Duda and R. E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley.

G. D. Forney. 1973. The Viterbi Algorithm. In *Proc. of the IEEE*, 61:268-278.

W. N. Francis and H. Kučera. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin Company, Boston, Massachusetts.

I. J. Good. 1953. "The Population Frequencies of Species and the Estimation of Population Parameters," In *Biometrika*, 40(3-4):237-264.

S. M. Katz. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. In *IEEE Transactions on Acoustics, Speech and Signal Processing(ASSP)*, 35(3):400-401.

J.-D. Kim, S.-Z. Lee, and H.-C. Rim. 1998. A Morpheme-Unit POS Tagging Model Considering Word-Spacing. In *Proc. of the 10th National Conference on Korean Information Processing*, 3-8.

J.-D. Kim, S.-Z. Lee, and H.-C. Rim. 1999. HMM Specialization with Selective Lexicalization. In *Proc. of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora(EMNLP-VLC-99)*, 144-148.

J.-H. Kim. 1996. *Lexical Disambiguation with Error-Driven Learning*. Doctoral Dissertation, Korea Advanced Institute of Science and Technology(KAIST), Korea.

S.-H. Lee. 1995. *Korean POS Tagging System Considering Unknown Words*. Master Thesis, Korea Advanced Institute of Science and Technology(KAIST), Korea.

S.-Z. Lee, J.-D. Kim, W.-H. Ryu, and H.-C. Rim. 1999. A Part-of-Speech Tagging Model Using Lexical Rules Based on Corpus Statistics. In *Proc. of the International Conference on Computer Processing of Oriental Languages(ICCPOL-99)*, 385-390.

S.-Z. Lee. 1999. *New Statistical Models for Automatic POS Tagging*. Doctoral Dissertation, Korea University, Korea.

B. Merialdo. 1991. Tagging Text with a Probabilistic Model. In *Proc. of the International Conference on Acoustic, Speech and Signal Processing(ICASSP-91)*, 809-812.

A. Ratnaparkhi. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proc. of the Empirical Methods in Natural Language Processing Conference(EMNLP-96)*, 133-142.