Automatic Word Spacing Using Hidden Markov Model for Refining Korean Text Corpora

Do-Gil Lee and **Sang-Zoo Lee** and **Hae-Chang Rim**
NLP Lab., Dept. of Computer Science and Engineering, Korea University
1, 5-ka, Anam-dong, Seongbuk-ku, Seoul 136-701, Korea

Heui-Seok Lim
Dept. of Information and Communications, Chonan University
115 AnSeo-dong, CheonAn 330-704, Korea

Abstract

This paper proposes a word spacing model using a hidden Markov model (HMM) for refining Korean raw text corpora. Previous statistical approaches for automatic word spacing have used models that make use of inaccurate probabilities because they do not consider the previous spacing state. We consider word spacing problem as a classification problem such as Part-of-Speech (POS) tagging and have experimented with various models considering extended context. Experimental result shows that the performance of the model becomes better as the more context considered. In case of the same number of parameters are used with other method, it is proved that our model is more effective by showing the better results.

1 Introduction

Automatic word spacing is a process to decide correct boundaries between words in a sentence containing spacing errors. In Korean, word spacing is very important to increase the readability and to communicate the accurate meaning of a text. For example, if a sentence "아버지가 방에 들어가셨다(Father entered the room)" is written as "아버지 가방에 들어가셨다(Father entered the bag)", then its meaning is changed a lot.

There are many word spacing errors in documents on the Internet, which is the principal source of information. To deal with these documents properly, an automatic word spacing system is absolutely necessary. Besides, it plays an important role as a preprocessor of a morphological analyzer that is a fundamental tool for natural language processing applications, a postprocessor to restore line boundaries from an OCR, a postprocessor for continuous-syllable sentence from a speech recognition system, and one module for an orthographic error revision system.

In Korean, spacing unit is Eojeol. Each Eojeol consists of one or more words and a word consists of one or more morphemes. Figure 1 represents their relationships for a sentence "철수가 이야기책을 읽었다". According to the rules of Korean spelling, the main principle for word spacing is to split every word in a sentence. Because one morpheme may form a word and several morphemes too, there are confusing cases to distinguish among words. Even though postpositions belong to words, they should be concatenated with the preceding word. Besides, there are many conflicting (but can be permitted) cases with the principles. For example, spacing or concatenating individual nouns including a compound noun are both considered as right. As mentioned, word spacing is important for some reasons, but it is difficult for even man to space words correctly by spelling rules because of the characteristics of Korean and the inconsistent rules. Especially, it is much more confused in the case of having no influence on understanding the meaning of a sentence.

In this paper, we propose a word spacing model [1] using an HMM. HMM is a widely used statistical model to solve various NLP problems such as POS tagging(Charniak et al., 1993; Merialdo, 1994; Kim et al., 1998a; Lee, 1999). We regard the word spacing problem as a classification problem such as the POS tagging problem. When using an HMM for automatic word spacing task, raw texts can be used as training

[1] Strictly speaking, our model described here is an Eojeol spacing model rather than a word spacing model because spacing unit of Korean is Eojeol. But we in this paper do not distinguish between Eojeol and word for convenience. Therefore, we use the term "word" as word, spacing unit in English.

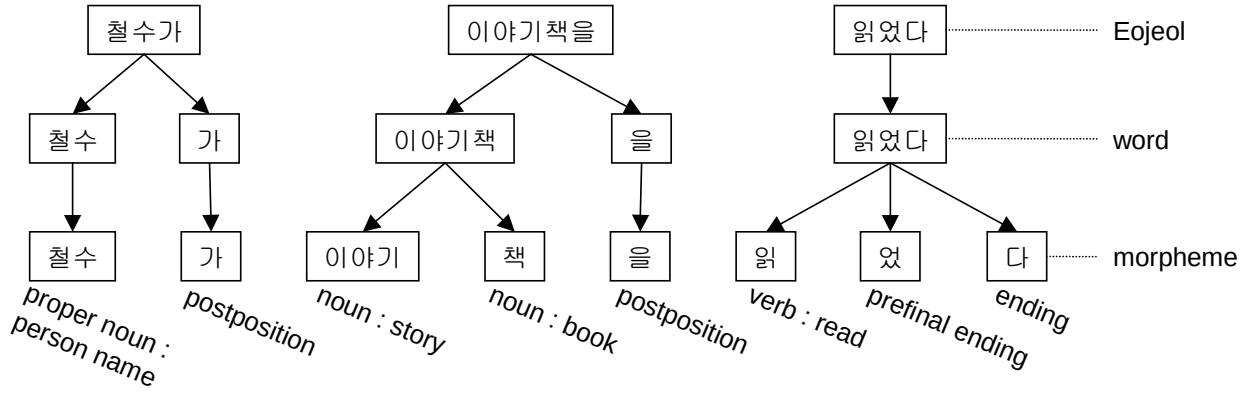

Figure 1: Constitution of the sentence "철수가 이야기책을 읽었다"

data. Therefore, we expect that HMM can be applied to the task effectively without bothering to construct training data.

2 Related Works

Previous approaches for automatic word spacing can be classified into two groups: rule based approach and statistical approach. The rule-based approach uses lexical information and heuristic rules(Choi, 1997; Kim et al., 1998b; Kang, 1998; Kang, 2000). Lexical information consists of postposition and Eomi[2] information, a list of spaced word examples, etc. Heuristic rules are composed of longest match or shortest match rule, morphological rules, and error patterns. This approach has disadvantage requiring higher computational complexity than the statistical approach. It also costs too much in constructing and maintaining lexical information. Most of rule-based systems use a morphological analyzer to recognize word boundaries. Another disadvantages of rule-based approach are resulted from using morphological analyzer. First, if ambiguous analyses are possible, frequent backtracking may be caused and many errors are propagated by an erroneous analysis. Second, results of automatic word spacing are highly dependent on the morphological analyzer; false word boundary recognition occurs if morphological analysis fails due to unknown words. In addition, if an erroneous word is successfully analyzed through overgeneration, the error cannot even be detected. Finally, if a word spacing system is used as a preprocessor of a morphological analyzer, the same morphological analyzing process should be repeated twice.

The statistical approach uses syllable statistics extracted from large amount of corpora to decide whether two adjacent syllables should be spaced or not(Shim, 1996; Shin and Park, 1997; Chung and Lee, 1999; Jeon and Park, 2000; Kang and Woo, 2001). In contrast to the rule-based approach, it does not require many costs to construct and to maintain statistics because they can be acquired automatically. It is more robust against unknown words than rule-based approach that uses a morphological analyzer.

A statistical method proposed in Kang and Woo (2001) has shown the best performance so far. In this method, word spacing probability $P(x_i, x_{i+1})$, between two adjacent syllables x_i and x_{i+1}, is in Equation 1. If the probability is greater than 0.375, a space is inserted between x_i and x_{i+1}.

$$\begin{aligned} P(x_i, x_{i+1}) = {} & 0.25 \times P_R(x_{i-1}, x_i) + \\ & 0.5 \times P_M(x_i, x_{i+1}) + \\ & 0.25 \times P_L(x_{i+1}, x_{i+2}) \end{aligned} \quad (1)$$

In Equation 1, P_R, P_M, and P_L denote the probability of a space being inserted in the right, middle, and left of the two syllables, respectively. They are calculated as follows:

$$P_R(x_{i-1}, x_i) = \frac{freq(x_{i-1}, x_i, SPACE)}{freq(x_{i-1}, x_i)}$$

$$P_M(x_i, x_{i+1}) = \frac{freq(x_i, SPACE, x_{i+1})}{freq(x_i, x_{i+1})}$$

[2]Eomi is a grammatical morpheme of Korean which is attached to verbal root

$$P_L(x_{i+1}, x_{i+2}) = \frac{freq(SPACE, x_{i+1}, x_{i+2})}{freq(x_{i+1}, x_{i+2})}$$

In the above equations, $freq(x)$ denotes a frequency of a string x from training data, and $SPACE$ denotes a white space.

Similar to this method, other statistical systems usually use the word spacing probability estimated from every syllable bigram[3] in the corpora. They calculate the probability by combining P_R, P_M, and P_L and compare it with a certain threshold. If the probability is higher than the threshold, then a space is inserted between two syllables.

It is reported that the performance is so sensitive to training data: it shows somewhat different performance according to similarity between input document and training data. And there is a crucial problem in the statistical method resulted from not considering the previous spacing state. For example, consider a sentence "공부할수있다" of which correctly word spaced sentence is "공부할 수 있다". According to Equation 1, the word spacing probability of "수" and "있" will be calculated as follows:

$$P(\text{수}, \text{있}) = 0.25 \times P_R(\text{할}, \text{수}) + 0.5 \times P_M(\text{수}, \text{있}) + 0.25 \times P_L(\text{있}, \text{다})$$

The probability $P_R(\text{할}, \text{수})$ as follows:

$$P_R(\text{할}, \text{수}) = \frac{freq(\text{할}, \text{수}, SPACE)}{freq(\text{할}, \text{수})}$$

But a space should have been inserted between "할" and "수" in the correct sentence, we should use $freq(SPACE, \text{수}, SPACE)$ instead of $freq(\text{할}, \text{수}, SPACE)$ in order to get the correct word spacing probability. This phenomenon comes from not considering the previous spacing state. To alleviate this problem, we can consider the previous spacing state that the system has decided before. But errors can be propagated from the previous false word spacing result. Eventually, to avoid such propagated errors, the system has to generate all possible interpretations from a given sentence and choose the best one. To choose the best state from all possible states, we use an HMM in this paper.

[3]syllable bigram is defined to be any combination of two syllables with or without a space.

3 Word Spacing Model based on Hidden Markov Model

POS tagging is the most representative area for HMM. Before explaining our word spacing model using HMM, let's consider the POS tagging model using an HMM. POS tagging function $\Gamma(W)$ is to find the most likely sequence of POS tags $T = (t_1, t_2, \ldots, t_n)$ for a given sentence of words $W = (w_1, w_2, \ldots, w_n)$ and is defined in Equation 2:

$$\Gamma(W) \stackrel{def}{=} \operatorname*{argmax}_T P(T \mid W) \quad (2)$$

$$= \operatorname*{argmax}_T \frac{P(T)P(W \mid T)}{P(W)} \quad (3)$$

$$= \operatorname*{argmax}_T P(T)P(W \mid T) \quad (4)$$

$$= \operatorname*{argmax}_T P(T, W) \quad (5)$$

Using Bayes' rule, Equation 2 becomes Equation 3. Since $P(W)$ is a constant for T, Equation 3 is transformed into Equation 4.

The probability $P(T, W)$ is broken down into the following equations by using the chain rule:

$$P(T, W) = P(t_{1,n}, w_{1,n}) \quad (6)$$

$$= \prod_{i=1}^{n} \begin{pmatrix} P(t_i \mid t_{1,i-1}, w_{1,i-1}) \\ \times P(w_i \mid t_{1,i}, w_{1,i-1}) \end{pmatrix} \quad (7)$$

$$\approx \prod_{i=1}^{n} P(t_i \mid t_{i-K,i-1}) P(w_i \mid t_i) \quad (8)$$

Markov assumptions (conditional independence) used in Equation 8 are that the probability of a current tag t_i conditionally depends on only the previous K tags and that the probability of a current word w_i conditionally depends on only the current tag. In Equation 8, $P(t_i \mid t_{i-K,i-1})$ is called transition probability and $P(w_i \mid t_i)$ is called lexical probability. Models are classified in terms of K. The larger K is, the more context can be considered. Because of the data sparseness problem, bigram model (K is 1) and trigram model (K is 2) are used in general.

The word spacing problem can be considered similar to POS tagging. We define a word spacing task as a task to find the most likely sequence of word spacing tags $T = (t_1, t_2, \ldots, t_n)$ for a given sentence of syllables

$S = (s_1, s_2, \ldots, s_n)$. Our word spacing model is defined as in Equation 9:

$$\underset{T}{\operatorname{argmax}}\, P(T \mid S) \tag{9}$$

Word spacing tag is a tag to indicate whether the current syllable and the next one should be spaced or not. Tag, 1 means that a space should be put after the current syllable. Tag, 0 means that the current and the next syllable should not be spaced. For example, if we attach the word spacing tags to a sentence "공부할 수 있다. (I can study)", then it is tagged as "공/0+부/0+할/1+수/1+있/0+다/0+./1".

Our proposed word spacing model is to find the tag sequence T for maximizing the probability $P(T, S)$.

$$\begin{aligned}
&P(T,\ S\ ) \\
&\quad = P(t_{1,n}, s_{1,n}) && (10)\\
&\quad = \Big(P(t_1) \times p(s_1 \mid t_1)\Big) \\
&\qquad \times \Big(P(t_2 \mid t_1, s_1) \times P(s_2 \mid t_{1,2}, s_1)\Big) \\
&\qquad \times \begin{pmatrix} P(t_3 \mid t_{1,2}, s_{1,2}) \\ \times P(s_3 \mid t_{1,3}, s_{1,2}) \end{pmatrix} \times \cdots \\
&\qquad \times \begin{pmatrix} P(t_n \mid t_{1,n-1}, s_{1,n-1}) \\ \times P(s_n \mid t_{1,n}, s_{1,n-1}) \end{pmatrix} && (11)\\
&\quad = \prod_{i=1}^{n} \begin{pmatrix} P(t_i \mid t_{1,i-1}, s_{1,i-1}) \\ \times P(s_i \mid t_{1,i}, s_{1,i-1}) \end{pmatrix} && (12)\\
&\quad \approx \prod_{i=1}^{n} \begin{pmatrix} P(t_i \mid t_{i-K,i-1}, s_{i-J,i-1}) \\ \times P(s_i \mid t_{i-L,i}, s_{i-I,i-1}) \end{pmatrix} && (13)
\end{aligned}$$

There are two Markov assumptions in Equation 13. One is that the probability of a current tag t_i conditionally depends on only the previous K (word spacing) tags and the previous J syllables. The other is that the probability of a current syllable s_i conditionally depends on only the previous L tags, the current tag t_i, and the previous I tags. This model is denoted by $\Lambda(T_{(K:J)}, S_{(L:I)})$. Similar to the POS tagging model, $P(t_i \mid t_{i-K,i-1}, s_{i-J,i-1})$ is called transition probability, and $P(s_i \mid t_{i-L,i}, s_{i-I,i-1})$ is called syllable probability in Equation 13. On the other hand, our word spacing model uses less strict Markov assumptions to consider a larger context. The larger the values of K, J, L, and I are, the more context can be considered. In order to avoid the data sparseness and excessively increasing parameters of a model, it is important to select proper values. In our current work, they are restricted as follows:

$$0 \le K, J, L, I \le 2$$

Thus, $3 \times 3 \times 3 \times 3 = 81$ models are possible. But we do not use the case of $(K, J) = (0, 0)$ in the trasition probabilities. As a result, we actually use 72 models. It has not yet been known that which model is the best. We can verify this only by means of experiments. Some possible models and their equations are listed in Table 1.

Probabilities can be estimated simply by the maximum likelihood estimator (MLE) from raw texts. The syllable probabilities and the transition probabilities of the model $\Lambda(T_{(1:2)}, S_{(1:2)})$ are estimated as follows:

$$\begin{aligned}
&P_{MLE}(t_i \mid t_{i-1}, s_{i-2,i-1}) \\
&\qquad = \frac{freq(s_{i-2}, t_{i-1}, s_{i-1}, t_i)}{freq(s_{i-2}, t_{i-1}, s_{i-1})} \\
&P_{MLE}(s_i \mid t_{i-1,i}, s_{i-2,i-1}) \\
&\qquad = \frac{freq(s_{i-2}, t_{i-1}, s_{i-1}, t_i, s_i)}{freq(s_{i-2}, t_{i-1}, s_{i-1}, t_i)}
\end{aligned}$$

To avoid zero probability, we just set very low value such as 0.00001 if an estimated probability is 0.

The probability that the model $\Lambda(T_{(1:1)}, S_{(0:1)})$ outputs "공/0+부/0+할/1+수/1+있/0+다/0+./1" from a sentence "공부할수있다." is calculated as follows:

$$\begin{aligned}
P(T, S) = {} & P(t_0 = 0 \mid s_{-1} = \$, t_{-1} = 1) \\
& \times P(s_0 = 공 \mid s_{-1} = \$, t_0 = 0) \\
& \times P(t_1 = 0 \mid s_0 = 공, t_0 = 0) \\
& \times P(s_1 = 부 \mid s_0 = 공, t_1 = 0) \\
& \times P(1 \mid 부0) \cdot P(할 \mid 부1) \\
& \times P(1 \mid 할1) \cdot P(수 \mid 할1) \\
& \times P(0 \mid 수1) \cdot P(있 \mid 수0) \\
& \times P(0 \mid 있0) \cdot P(다 \mid 있0) \\
& \times P(1 \mid 다0) \cdot P(. \mid 다1)
\end{aligned}$$

"$" is a pseudo syllable which denotes the start of a sentence, and its tag is always 1.[4] The

[4]Because any two adjacent sentences should always be spaced.

Table 1: Some models and their equations

Model	Equation
$\Lambda(T_{(1:0)}, S_{(0:0)})$	$\prod_{i=1}^{n} P(t_i \mid t_{i-1}) \cdot P(s_i \mid t_i)$
$\Lambda(T_{(1:1)}, S_{(0:1)})$	$\prod_{i=1}^{n} P(t_i \mid t_{i-1}, s_{i-1}) \cdot P(s_i \mid t_i, s_{i-1})$
$\Lambda(T_{(1:1)}, S_{(1:1)})$	$\prod_{i=1}^{n} P(t_i \mid t_{i-1}, s_{i-1}) \cdot P(s_i \mid t_{i-1,i}, s_{i-1})$
$\Lambda(T_{(1:2)}, S_{(1:2)})$	$\prod_{i=1}^{n} P(t_i \mid t_{i-1}, s_{i-2,i-1}) \cdot P(s_i \mid t_{i-1,i}, s_{i-2,i-1})$
$\Lambda(T_{(2:2)}, S_{(2:2)})$	$\prod_{i=1}^{n} P(t_i \mid t_{i-2,i-1}, s_{i-2,i-1}) \cdot P(s_i \mid t_{i-2,i}, s_{i-2,i-1})$

most probable sequence of word spacing tags is efficiently computed by using the Viterbi algorithm.

4 Experimental Results

We used balanced 21st Century Sejong Project's raw corpus of 26 million word size. As the balanced corpus is used as training data, we expect that the performance would not be sensitive too much to a certain document genre. The ETRI POS tagged corpus of 288,269 word size was used for evaluation. We modified the corpus with no word boundary form for automatic word spacing evaluation.

We used three kinds of evaluation measures: syllable-unit accuracy (P_{syl}), word-unit recall (R_{word}), and word-unit precision (P_{word}). The word-unit recall is the rate of the number of correctly spaced words compared to the number of total words in a test document. The word-unit precision measures how accurate the system's results are. The reason why we do not divide the syllable-unit accuracy as recall and precision is that the number of syllables in a document and that of the system created are the same. Each measure is defined as follows:

$$P_{syl} = \frac{S_{correct}}{S_{total}} \times 100(\%)$$

$$R_{word} = \frac{W_{correct}}{W_{Dtotal}} \times 100(\%)$$

$$P_{word} = \frac{W_{correct}}{W_{Stotal}} \times 100(\%)$$

Where, $S_{correct}$ is the number of correctly spaced syllables, S_{total} is the total number of syllables in a document, $W_{correct}$ is the number of correctly spaced words, W_{Dtotal} is the total number of words in a document, and W_{Stotal} is the total number of words created by a system.

To investigate every model, we calculated the two accuracies for different K, J, L, and I. Accuracies for each model are listed in Table 2.

According to the experimental results, we are sure that models considering more contexts show better results. The model $\Lambda(T_{(2:2)}, S_{(1:2)})$ is the best for all measures.

Note that some models show the better accuracies than the model $\Lambda(T_{(2:2)}, S_{(2:2)})$, which uses the largest context. It seems that this is caused by sparseness of data. After evaluating the method of Kang and Woo (2001) for our training and test data, it shows 93.06% syllable-unit accuracy, 76.71% word-unit recall, and 67.80% word-unit precision. Compared with these results, our model shows much better performance. If I is two in $\Lambda(S_{(K:J)}, T_{(L:I)})$, syllable trigrams are used. Although I is less than two (such as the model $\Lambda(T_{(2:1)}, S_{(1:1)}$, which uses syllable bigrams), our model is better than Kang and Woo (2001)'s. This fact tells us that our model is also more effective even when used the same number of parameters of the model.

There are two questions that we want to know about the word spacing models: First, how much training data is required to get the best performance of a given model. Second, which model best fits a given training corpus. To answer these questions, we compare the performance of various models according to the size of training corpus in Figure 2. The left plot shows the syllable-unit precision and the right plot shows the word-unit precision. In the figure, "HMM" denotes the proposed model, and its number decides the model's type. "Kang" denotes Kang and Woo (2001)'s model. "HMM2110" uses syllable unigrams, "HMM2111" and "Kang" use syllable bigrams, and "HMM2212" uses syllable trigrams. The models used here are the models that show the best accuracies among the models that use same

Table 2: Experimental results according to (K, J, L, I)

Model	P_{syl}	R_{word}	P_{word}	Model	P_{syl}	R_{word}	P_{word}	Model	P_{syl}	R_{word}	P_{word}
(0,1,0,0)	84.26	41.28	44.06	(0,1,0,1)	88.93	55.38	57.10	(0,1,0,2)	88.45	53.83	55.88
(0,1,1,0)	89.44	56.91	61.34	(0,1,1,1)	95.58	79.31	82.58	(0,1,1,2)	95.74	79.76	83.68
(0,1,2,0)	84.44	42.15	47.02	(0,1,2,1)	92.86	70.26	71.63	(0,1,2,2)	94.97	76.90	79.45
(0,2,0,0)	85.48	45.65	47.52	(0,2,0,1)	88.93	56.24	57.21	(0,2,0,2)	89.59	58.23	59.88
(0,2,1,0)	90.22	59.12	63.74	(0,2,1,1)	95.60	79.26	82.94	(0,2,1,2)	95.92	80.41	84.56
(0,2,2,0)	86.46	47.62	52.15	(0,2,2,1)	93.44	72.06	73.90	(0,2,2,2)	95.22	77.84	80.59
(1,0,0,0)	85.75	47.05	48.96	(1,0,0,1)	90.24	60.73	62.20	(1,0,0,2)	89.74	58.68	61.09
(1,0,1,0)	89.28	59.80	59.98	(1,0,1,1)	95.64	81.17	81.81	(1,0,1,2)	95.90	81.50	83.56
(1,0,2,0)	82.85	45.10	45.38	(1,0,2,1)	93.30	73.04	73.39	(1,0,2,2)	94.94	77.52	78.88
(1,1,0,0)	85.83	49.95	50.43	(1,1,0,1)	90.96	63.18	64.89	(1,1,0,2)	90.21	62.99	62.58
(1,1,1,0)	89.85	61.47	62.80	(1,1,1,1)	96.15	82.88	84.10	(1,1,1,2)	96.17	82.67	84.86
(1,1,2,0)	84.21	49.44	49.29	(1,1,2,1)	94.07	75.54	76.87	(1,1,2,2)	95.62	80.32	82.13
(1,2,0,0)	87.21	54.25	54.85	(1,2,0,1)	90.83	63.34	64.59	(1,2,0,2)	91.54	66.39	67.00
(1,2,1,0)	90.74	64.14	65.63	(1,2,1,1)	96.07	82.44	84.09	(1,2,1,2)	96.39	83.51	85.91
(1,2,2,0)	86.96	55.50	55.95	(1,2,2,1)	94.67	77.53	79.28	(1,2,2,2)	95.90	81.39	83.42
(2,0,0,0)	86.18	50.25	51.42	(2,0,0,1)	90.44	61.97	63.61	(2,0,0,2)	89.77	61.52	62.17
(2,0,1,0)	89.49	61.07	61.32	(2,0,1,1)	95.83	82.11	82.73	(2,0,1,2)	95.91	82.09	83.39
(2,0,2,0)	83.37	46.52	47.15	(2,0,2,1)	93.55	73.91	74.63	(2,0,2,2)	95.03	78.36	78.96
(2,1,0,0)	86.51	52.60	53.46	(2,1,0,1)	91.10	64.81	65.85	(2,1,0,2)	90.69	65.11	65.10
(2,1,1,0)	90.34	64.04	64.90	(2,1,1,1)	96.29	83.73	84.74	(2,1,1,2)	96.28	83.43	85.21
(2,1,2,0)	85.07	52.32	52.63	(2,1,2,1)	94.31	76.69	77.82	(2,1,2,2)	95.91	81.51	83.45
(2,2,0,0)	88.58	58.94	59.84	(2,2,0,1)	91.78	67.07	68.32	(2,2,0,2)	92.44	69.88	70.54
(2,2,1,0)	91.65	67.82	69.14	(2,2,1,1)	96.26	83.46	84.88	(2,2,1,2)	**96.69**	**84.93**	**86.82**
(2,2,2,0)	88.97	61.20	62.28	(2,2,2,1)	95.01	78.99	80.60	(2,2,2,2)	96.04	82.05	83.96

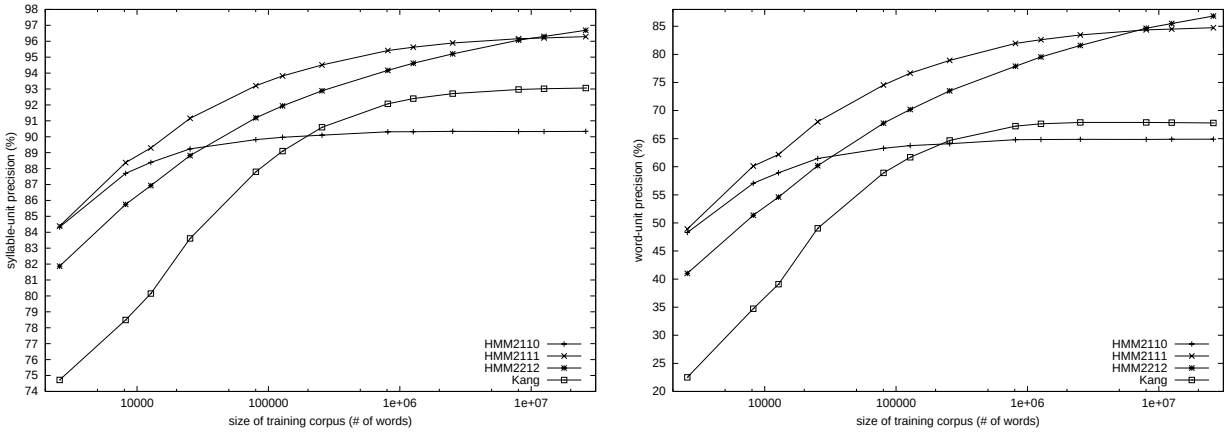

Figure 2: Accuracies according to the size of training corpus

syllable ngrams.

We can observe the changes of the accuracies according to the size of the training data. "HMM2110" using syllable unigrams converges quickly on small training data. "HMM2111" and "Kang" using syllable bigrams converge on much more training data. Note that "HMM2212" does not converge in these plots. Therefore, there is a possibility of improvement of this model's performance on more large training data. "HMM2212" shows lower performance than other models on small training

data. The reason is that the data sparseness problem occurs.

5 Conclusion

Recently, text resources available from the Internet have been rapidly increased. However, there are many word spacing errors in those resources, which cannot be used before correcting errors. Therefore, the need for automatic word spacing system to refine text corpora has been raised. In this paper, we have proposed an automatic word spacing model using an HMM. Our method is a statistical approach and does not require complex processes and costs in constructing and maintaining lexical information as in the rule-based approach. The proposed model can effectively solve the word spacing problem by using only syllable statistics automatically extracted from raw corpora. According to the experimental results, our model shows higher performance than the previous method even when using the same number of parameters. We used just MLE to estimate probability, but the more a model extends the context; the more the data sparseness problem may arise.

In future work, we plan to adopt a smoothing technique to increase the performance. Further research on an effective evaluation method for conflicting cases is also necessary.

References

E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz. 1993. Equations for part-of-speech tagging. In *National Conference on Artificial Intelligence*, pages 784–789.

J.-H. Choi. 1997. Automatic Korean spacing words correction system with bidirectional longest match strategy. In *Proceedings of the 9th Conference on Hangul and Korean Information Processing*, pages 145–151.

Y.-M. Chung and J.-Y. Lee. 1999. Automatic word-segmentation at line-breaks for Korean text processing. In *Proceedings of the 6th Conference of Korea Society for Information Mangement*, pages 21–24.

N.-Y. Jeon and H.-R. Park. 2000. Automatic word-spacing of syllable bi-gram information for Korean OCR postprocessing. In *Proceedings of the 12th Conference on Hangul and Korean Information Processing*, pages 95–100.

S.-S. Kang and C.-W. Woo. 2001. Automatic segmentation of words using syllable bigram statistics. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium*, pages 729–732.

S.-S. Kang. 1998. Automatic word-segmentation for Hangul sentences. In *Proceedings of the 10th Conference on Hangul and Korean Information Processing*, pages 137–142.

S.-S. Kang. 2000. Eojeol-block bidirectional algorithm for automatic word spacing of Hangul sentences. *Journal of the Korea Information Science Society*, 27(4):441–447.

J.-D. Kim, H.-S. Lim, S.-Z. Lee, and H.-C. Rim. 1998a. Twoply hidden markov model: A Korean pos tagging model based on morpheme-unit with word-unit context. *Computer Processing of Oriental Languages*, 11(3):277–290.

K.-S. Kim, H.-J. Lee, and S.-J. Lee. 1998b. Three-stage spacing system for Korean in sentence with no word boundaries. *Journal of the Korea Information Science Society*, 25(12):1838–1844.

S.-Z. Lee. 1999. New statistical models for automatic part-of-speech tagging. Ph.D. thesis, Korea University.

B. Merialdo. 1994. Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–172.

Kwangseob Shim. 1996. Automated word-segmentation for Korean using mutual information of syllables. *Journal of the Korea Information Science Society*, 23(9):991–1000.

J.-H. Shin and H.-R. Park. 1997. A statistical model for Korean text segmentation using syllable-level bigrams. In *Proceedings of the 9th Conference on Hangul and Korean Information Processing*, pages 255–260.