# Balanced Korean Word Spacing with Structural SVM

**Changki Lee*** **Edward Choi** **Hyunki Kim**
*Kangwon National University, Chuncheon-si, Gangwondo, 200-701, Korea
Electronics and Telecommunications Research Institute, Daejeon, 305-350, Korea
leeck@kangwon.ac.kr mp2893@gmail.com hkk@etri.re.kr

## Abstract

Most studies on statistical Korean word spacing do not utilize the information provided by the input sentence and assume that it was completely concatenated. This makes the word spacer ignore the correct spaced parts of the input sentence and erroneously alter them. To overcome such limit, this paper proposes a structural SVM-based Korean word spacing method that can utilize the space information of the input sentence. The experiment on sentences with 10% spacing errors showed that our method achieved 96.81% F-score, while the basic structural SVM method only achieved 92.53% F-score. The more the input sentence was correctly spaced, the more accurately our method performed.

## 1 Introduction

Automatic word spacing is a task to decide boundaries between words, which is frequently used for correcting spacing errors of text messages, Tweets, or Internet comments before using them in information retrieval applications (Lee and Kim, 2012). It is also often used in post-processing optical character recognition (OCR) or voice recognition (Lee et al., 2007). Except for some Asian languages such as Chinese, Japanese and Thai, most languages have explicit word spacing that improves the readability of the text and helps readers better understand the meaning of it. Korean especially has a tricky word spacing system and users often make mistakes, which makes automatic word spacing an interesting and essential task.

In order to easily acquire the training data, most studies on statistical Korean word spacing assume that well-spaced raw text (e.g. newspaper articles) is perfectly spaced and use it for training (Lee and Kim, 2012; Lee and Kim, 2013; Lee et al., 2007; Shim, 2011). This approach, however, cannot observe incorrect spacing since the assumption makes the training data devoid of negative example. Consequently, word spacers cannot use the spacing information given by the user, and erroneously alter the correctly spaced parts of the sentence. To utilize the user-given spacing information, a corpus of input sentences and their correctly spaced version is necessary. Constructing such corpus, however, requires much time and resource.

In this paper, to resolve such issue, we propose a structural SVM-based Korean word spacing model that can utilize the word spacing information given by the user. We name the proposed model "Balanced Word Spacing Model (BWSM)". Our approach trains a basic structural SVM-based Korean word spacing model as in (Lee and Kim, 2013), and tries to obtain the sentence which achieves the maximum score for the basic model while minimally altering the input sentence.

In the following section, we discuss related studies. In Section 3, the proposed method and its relation to Karush-Kuhn-Tucker (KKT) condition are explained. The experiment and discussion is presented in Section 4. Finally, in Section 5, the conclusion and future work for this study is given.

## 2 Related Work

There are two common approaches to Korean word spacing: rule-based approach and statistical approach. In rule-based approach, it is not easy to construct rules and maintain them. Furthermore, it requires morphological analysis to apply rule-based approach, which slows down the process. Recent studies, therefore, mostly focus on the statistical approach.

Most statistical approaches use well-spaced raw corpus as training data (e.g. newspaper articles) assuming that they are perfectly spaced. This is to avoid the expensive job of constructing new training data. Lee et al. (2007) treated the word spacing task as a sequence labeling problem on the input sentence which is a sequence of syllables. They proposed a method based on Hidden Markov Model (HMM). Shim (2011) also considered the word spacing task as a sequence labeling problem and proposed a method using Conditional Random Field (CRF) (Lafferty et al., 2001), which is a well-known powerful model for sequence labeling tasks. Lee and Kim



*Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 875–879,
October 25-29, 2014, Doha, Qatar. 

(2013) tried to solve the sequence labeling problem using structural SVM (Tsochantaridis et al., 2004; Joachims et al., 2009; Lee and Jang 2010; Shalev-Shwartz et al., 2011).

The studies above (Lee and Kim, 2013; Lee et al., 2007; Shim, 2011), however, do not take advantage of the spacing information provided by the user, and often erroneously alter the correctly spaced part of the sentence. Lee et al. (2007) tries to resolve this issue by combining an HMM model with an additional confidence model constructed from another corpus. Given an input sentence, they first apply the basic HMM model to obtain a candidate sentence. For every different word spacing between the input sentence and the candidate sentence, they calculate and compare the confidence using the confidence model, and whichever gets the higher confidence is used. The spacing accuracy was improved from 97.52% to 97.64%[1].

This study is similar to (Lee et al., 2007) in that it utilizes the spacing information given by the user. But unlike (Lee et al., 2007), BWSM uses structural SVM as the basic model and do not require an additional confidence model. Furthermore, while Lee et al. (2007) compares the spacing confidence for each syllable to obtain the final outcome, BWSM considers the whole sentence when altering its spacing, enabling it to achieve higher improvement on performance (from 92.53% F-score to 96.81% F-score).

## 3 Balanced Word Spacing Model

Like previous studies, the proposed model treats the Korean word spacing task as a sequence labeling problem. The label consists of *B* and *I*, which are assigned to each syllable of the sentence. Assuming that $\mathbf{x} = \langle x_1, x_2, \ldots, x_T \rangle$ is a sequence of total *T* syllables of the input sentence and $\mathbf{y} = \langle y_1, y_2, \ldots, y_T \rangle$ is a sequence of labels for each syllable, an example could be given as follows[2]:

| |
|---|
| Input: ah/beo/ji/ga bang/eh deul/eo/ga/sin/da (Father entered the room) |
| **x** = <ah, beo, ji, ga, bang, eh, deul, eo, ga, sin, da> |
| **y** = <B, I, I, I, B, I, B, I, I, I, I> |

Figure 1: An example of word spacing.

In order to utilize the spacing information provided by the user, we propose a new model, the Balanced Word Spacing Model that adheres to the following principles:

1. The model must obtain the most likely sequence of labels($\mathbf{y}^*$), while minimally altering the user-given sequence of labels ($\mathbf{y}_{input}$).
2. We assume that it costs α per syllable to change the spacing of the original sentence, in order to keep the original spacing information as much as possible.

Mathematically formulating the above principles would give us the following equation:

$$\mathbf{y}^* = \text{argmax}\{score(\mathbf{x}, \mathbf{y}) - \alpha \cdot L(\mathbf{y}_{input}, \mathbf{y})\} \quad (1)$$

In Equation 1, $score(\mathbf{x}, \mathbf{y})$ calculates how compatible the sequence of label **y** is with the input sentence **x**. It is calculated by a basic word spacing model as in (Lee and Kim, 2013). $L(\mathbf{y}_{input}, \mathbf{y})$ counts the number of different labels between the user-given sequence of labels $\mathbf{y}_{input}$ and an arbitrary sequence of labels **y**. $\mathbf{y}^*$ of Equation 1 can be obtained by setting the gradient of $score(\mathbf{x}, \mathbf{y}) - \alpha \cdot L(\mathbf{y}_{input}, \mathbf{y})$ to 0, which is equivalent to the following equation:

$$\nabla score(\mathbf{x}, \mathbf{y}^*) = \alpha \cdot \nabla L(\mathbf{y}_{input}, \mathbf{y}^*) \quad (2)$$

In order to view the proposed model in a different perspective, we consider BWSM in terms of Karush-Kuhn-Tucker (KKT) condition. KKT condition is a technique for solving optimization problems with inequality constraints. It is a generalized version of Lagrange multipliers, which is a technique for solving optimization problems with equality constraints. Converting the aforementioned principles to a constrained optimization problem gives:

$$\text{Maximize: } score(\mathbf{x}, \mathbf{y})$$
$$\text{subject to } L(\mathbf{y}_{input}, \mathbf{y}) \leq b \quad (3)$$

Equation 3 tries to obtain **y** that maximizes $score(\mathbf{x}, \mathbf{y})$, namely the score of the basic model, while maintaining $L(\mathbf{y}_{input}, \mathbf{y})$ below *b*, which is equivalent to altering the word spacing of the input sentence less than or equal to *b* times. To solve this constrained optimization problem, we apply KKT condition and define a new Lagrangian function as follows:

$$\Lambda(\mathbf{x}, \mathbf{y}, \alpha) = score(\mathbf{x}, \mathbf{y}) - \alpha\{L(\mathbf{y}_{input}, \mathbf{y}) - b\} \quad (4)$$

[1] Accuracy was calculated based on syllables.
[2] Slashes are used for distinguishing between syllables.

Setting the gradient of the Equation 4 to zero, namely $\nabla\Lambda(\mathbf{x}, \mathbf{y}, \alpha) = 0$, we get the following necessary conditions:

$$\begin{aligned}&\text{Stationarity: } \nabla score(\mathbf{x}, \mathbf{y}^*) = \alpha^*\{\nabla L(\mathbf{y}_{input}, \mathbf{y}^*)\}\\&\text{Primal feasibility: } L(\mathbf{y}_{input}, \mathbf{y}^*) \leq b\\&\text{Dual feasibility: } \alpha^* \geq 0\\&\text{Complementary slackness: } \alpha^*\{L(\mathbf{y}_{input}, \mathbf{y}^*) - b\} = 0 \quad (5)\end{aligned}$$

Comparing Equation 1 with Equation 4 reveals that they are the same except the constant $b$. And $\mathbf{y}^*$ which satisfies the conditions of Equation 5, and hence the solution to Equation 4, is also the same as $\mathbf{y}^*$ which satisfies Equation 2, and hence the solution to Equation 1.

For the basic word spacing model, we use margin rescaled version of structural SVM as Lee and Kim (2013). The objective function of structural SVM is as follows:

$$\begin{aligned}&\min_{w,\xi} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{C}{n}\sum_i^n \xi_i, \ \ s.t. \ \forall i, \ \xi_i \geq 0\\&\forall i, \forall \mathbf{y} \in Y \backslash \mathbf{y}_i : \mathbf{w}^{\mathrm{T}} \delta\Psi(\mathbf{x}_i, \mathbf{y}) \geq L(\mathbf{y}_i, \mathbf{y}) - \xi_i\\&\text{where } \delta\Psi(\mathbf{x}_i, \mathbf{y}) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y}) \quad (6)\end{aligned}$$

In Equation 6, $(\mathbf{x}_i, \mathbf{y}_i)$ represents the i-th sequence of syllables and its correct spacing labels. $L(\mathbf{y}_i, \mathbf{y})$ is a loss function that counts the number of different labels between the correct labels $\mathbf{y}_i$ and the predicted sequence of labels $\mathbf{y}$. $\Psi(\mathbf{x}, \mathbf{y})$ is a typical feature vector function. The features used for the basic word spacing model are the same features used in (Lee and Kim, 2013). Since structural SVM was used for the basic word spacing model, the score function of Equation 1 becomes $score(\mathbf{x}, \mathbf{y}) = \mathbf{w}^{\mathrm{T}}\Psi(\mathbf{x}, \mathbf{y})$.

We propose two approaches for implementing Equation 1.

1. N-best re-ranking: N-best sequences of spacing labels are obtained using the basic structural SVM model. For each of the sequence, $\alpha^* L(\mathbf{y}_{input}, \mathbf{y}^*)$ is calculated and subtracted from $score(\mathbf{x}, \mathbf{y})$. The result of the subtraction is used to re-rank the sequences, and the one with the highest rank is chosen.
2. Modified Viterbi search: Viterbi search algorithm, which is used in the basic word spacing model to solve $\mathbf{y}^* = \text{argmax}\{\mathbf{w}^{\mathrm{T}}\Psi(\mathbf{x}, \mathbf{y})\}$, is modified to solve $\mathbf{y}^* = \text{argmax}\{\mathbf{w}^{\mathrm{T}}\Psi(\mathbf{x}, \mathbf{y}) - \alpha \cdot L(\mathbf{y}_{input}, \mathbf{y})\}$. Both $\Psi(\mathbf{x}, \mathbf{y})$ and $L(\mathbf{y}_{input}, \mathbf{y})$ can be calculated syllable by syllable, which makes it easy to modify Viterbi search algorithm.

The first approach seems straightforward and easy, but it would take a long time to obtain N-best sequences of labels. Furthermore, the correct label sequence might not be in those N-best sequences, hence degrading the overall performance. The second approach is fast since it does not calculate N-best sequences, and unlike the first approach, will always consider the correct label sequence as a candidate.

## 4 Experiment

In order to compare the performance of BWSM with HMM-based Korean word spacing and structural SVM-based Korean word spacing, we use Sejong raw corpus (Kang and Kim, 2004) as train data and ETRI POS tagging corpus as test data[3]. Pegasos-struct algorithm from (Lee and Kim, 2013) was used to train the basic structural SVM-based model. The optimal value for the tradeoff variable $C$ of structural SVM was found after conducting several experiments[4].

The rate of word spacing error varies depending on the corpus. Newspaper articles rarely have word spacing errors but text messages or Tweets frequently contain word spacing errors. To reflect such variety, we randomly insert spacing errors into the test set to produce various test sets with spacing error rate 0%, 10%, 20%, 35%, 50%, 60%, and 70%[5].

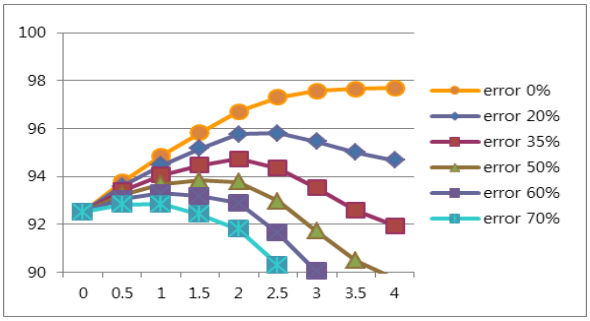


Figure 2: Word-based F-score of N-best re-ranking approach.

Figure 2 shows the relation between $\alpha$(x-axis) and word-based F-score[6](y-axis) of N-best re-

[3] The number of words for the training set and test set are 26 million and 290,000 respectively.
[4] We experimented with 10, 100, 1000, 10000, 100000 and 1000000, the optimal value being 100000.
[5] We altered the input to the system and retained the original gold standard's space unit.
[6] Word-based F-score = 2*$\text{Prec}_{\text{word}}$*$\text{Recall}_{\text{word}}$ / ($\text{Prec}_{\text{word}}$ + $\text{Recall}_{\text{word}}$),
$\text{Prec}_{\text{word}}$ = (# of correctly spaced words) / (the total number of words produced by the system),

ranking approach using test sets with different spacing error rate. When $\alpha = 0$, BWSM becomes a normal structural SVM-based model. As $\alpha$ increases, F-score also increases for a while but decreases afterward. And F-score increases more when using test sets with low error rate. It is worth noticing that when using the test set with 0% error rate, as $\alpha$ increases, F-score converges to 98%. The reason it does not reach 100% is that the correct label sequence is sometimes not included in the N-best sequences.

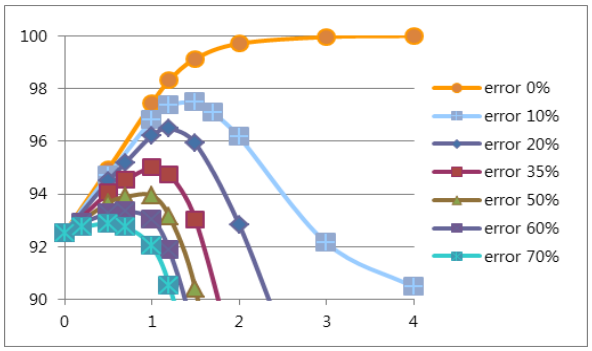


Figure 3: Word-based F-score of modified Viterbi search.

Figure 3 shows the relation between $\alpha$(x-axis) and word-based F-score(y-axis) of modified Viterbi search approach using test sets with different spacing error rate. The graphs are similar to Figure 2, but F-score reaches higher values compared to N-best re-ranking approach. Notice that, when using the test set with 0% error rate, F-score becomes 100% as $\alpha$ surpasses 3. This is because, unlike N-best re-ranking approach, modified Viterbi search approach considers all possible sequences as candidates.

From Figure 2 and 3, it can be seen that BWSM, which takes into consideration the spacing information provided by the user, can improve performance significantly. It is also apparent that modified Viterbi search approach outperforms N-best re-ranking approach. The optimal value for $\alpha$ varies as test sets with different error rate are used. It is natural that, for test sets with low error rate, the optimal value of $\alpha$ increases, thus forcing the model to more utilize the user-given spacing information. It is difficult to automatically obtain the optimal $\alpha$ for an arbitrary input sentence. Therefore we set $\alpha$ to 1, which, according to Figure 3, is more or less the optimal value for most of the test sets.

$\text{Recall}_{\text{word}}$ = (# of correctly spaced words) / (the total number of words in the test data)

| Model | Syllable based precision | Word based precision |
|---|---|---|
| HMM (Lee et al., 2007) | 98.44 | 90.31 |
| S-SVM (Lee and Kim, 2013) | 99.01 | 92.53 |
| Modified Viterbi (error rate 10%) | **99.64** | **96.81** |
| Modified Viterbi (error rate 20%) | **99.55** | **96.21** |
| Modified Viterbi (error rate 35%) | **99.35** | **95.01** |

Table 1: Precision of BWSM and previous studies

With $\alpha$ set to 1, and using modified Viterbi search algorithm, the performance of BWSM is shown in Table 1 with other previous studies (Lee and Kim, 2013; Lee et al., 2007). Table 1 shows that BWSM gives superior performance than other studies that do not utilize user-given spacing information.

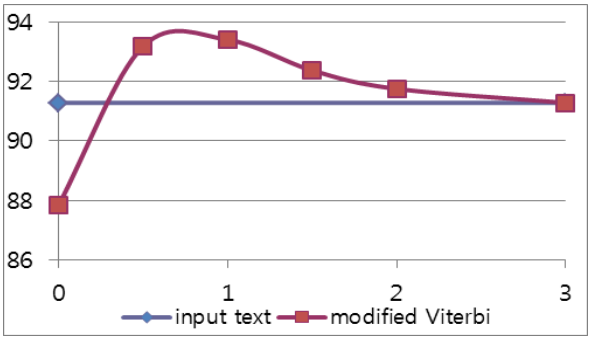


Figure 4: Word-based F-score of modified Viterbi search on Tweets.

We also collected Tweets from Twitter and tested modified Viterbi algorithm on them. Figure 4 shows the relation between $\alpha$ (x-axis) and word-based F-score (y-axis). The raw Tweets showed word-based F-score of approximate 91%, and the basic structural SVM model ($\alpha = 0$) showed somewhat inferior 88%. Modified Viterbi algorithm showed the similar behavior as Figure 3, showing 93.2~93.4% word-based F-score when $\alpha$ was set to 0.5~1. Figure 4 shows that BWSM is effective not only on text with randomly inserted spacing errors, but also on actual data, Tweets.

## 5 Conclusion

In this paper, we proposed BWSM, a new structural SVM-based Korean word spacing model that utilizes user-given spacing information. BWSM can obtain the most likely sequence of spacing labels while minimally altering the word spacing of the input sentence. Experiments on test sets with various error rate showed that BWSM significantly improved word-based F-

score, from 95.47% to 98.39% in case of the test set with 10% error rate.

For future work, there are two interesting directions. First is to improve BWSM so that it can automatically obtain the optimal value of $\alpha$ for an arbitrary sentence. This will require a training set consisting of text with actual human spacing errors and its corrected version. Second is to apply BWSM to other interesting problems such as named entity recognition (NER). Newspaper articles often use certain symbols such as quotation marks or brackets around the titles of movies, songs and books. Such symbols can be viewed as user-given input, which BWSM will try to respect as much as possible while trying to find the most likely named entities.

## Acknowledgments

This work was supported by the IT R&D program of MSIP/KEIT (10044577, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services). We would like to thank the anonymous reviewers for their comments.

## References

Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. 2009. Cutting-plane training of structural SVMs. *Machine Learning*, Vol. 77, No. 1.

Beom-mo Kang and Hunggyu Kim. 2004. Sejong Korean Corpora in the making. In *Proceedings of the LREC*, 1747-1750.

John Lafferty, Andrew McCallum and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the ICML*, 282-289.

Changki Lee and Myung-Gil Jang. 2010. A Modified Fixed-threshold SMO for 1-Slack Structural SVM. *ETRI Journal,* Vol.32, No.1, 120-128.

Changki Lee and Hyunki Kim. 2012. Automatic Korean word spacing using structural SVM. In *Proceedings of the KCC,* 270-272.

Changki Lee and Hyunki Kim. 2013. Automatic Korean word spacing using Pegasos algorithm. *Information Processing and Management*, Vol. 49, No. 1, 370-379.

Do-Gil Lee, Hae-Chang Rim and Dongsuk Yook. 2007. Automatic word spacing using probabilistic models based on character n-grams. *Intelligent Systems IEEE*, Vol. 22, No. 1, 28-35.

Seung-Wook Lee, Hae-Chang Rim and So-Young Park. 2007. A new approach for Korean word spacing incorporating confidence value of user's input. In *Proceedings of the ALPIT*, 92-97.

Kwang-Sup Shim. 2011. Automatic word spacing based on Conditional Random Fields. *Korean Journal of Cognitive Science*, Vol. 22, No. 2, 217-233.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. 2004. Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming*, Vol. 127, No. 1.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the ICML*, 104-111.