IEICE TRANS. INF. & SYST., VOL.E101–D, NO.2 FEBRUARY 2018

PAPER

# Phrase-Based Statistical Model for Korean Morpheme Segmentation and POS Tagging

**Seung-Hoon NA**[†a)], ***Member and*** **Young-Kil KIM**[††b)], ***Nonmember***

**SUMMARY** In this paper, we propose a novel phrase-based model for Korean morphological analysis by considering a phrase as the basic processing unit, which generalizes all the other existing processing units. The impetus for using phrases this way is largely motivated by the success of phrase-based statistical machine translation (SMT), which convincingly shows that the larger the processing unit, the better the performance. Experimental results using the SEJONG dataset show that the proposed phrase-based models outperform the morpheme-based models used as baselines. In particular, when combined with the conditional random field (CRF) model, our model leads to statistically significant improvements over the state-of-the-art CRF method.

***key words:*** *phrase-based model, segmentation, tagging, morphological analysis, Korean morphological analysis*

## 1. Introduction

In this paper, we address Korean morphological analysis consisting of three processing sub-tasks [1]–[7] - morpheme segmentation, part of speech (POS) tagging, and lemmatization, which are defined below.

1. Morpheme segmentation: Splitting an input sentence into morphemes.
2. POS tagging: Assigning a POS tag to each morpheme.
3. Lemmatization: Determining the lemma of each morpheme (or restoring the original form of each morpheme).

Table 1 presents an example of the three processing sub-tasks for the input sentence "na-neun hag-gyo-e gass-da" (I went to school). We use "-" as a separator between consecutive syllables within an eojeol. Eojeols are Korean word phrases that are defined as combinations of words and morphemes separated by spacing units. As shown in the example, both the morpheme segmentation and the POS tagging of Korean language are similar to the segmentation/tagging of other East Asian languages. In Korean morphological analysis, lemmatization not only recovers the original lemma of the surface form but also performs additional *internal segmentation* that decomposes the morpheme further into the atomic morphemes comprising a compound morpheme [8]. In the example above, POS tagging the input sentence produces "gass/VV~EP" (went), which is a compound morpheme*. Then, lemmatization decomposes the compound morphemes "gass/VV~EP" further into the atomic morphemes – "ga/VV" (go) and "at/EP" (past tense).

We conceive of Korean morphological analysis using a *noise-channel* model, where the goal is to recover the correct morphological analysis result given an input sentence. Given this perspective, probabilistic approaches to Korean morphological analysis can be viewed as specific cases of SMT (statistical machine translation) in which the source sentences correspond to sequences of input syllables and the target sentences correspond to morphological analysis results.

In the SMT literature, the achievement of phrase-based SMT represents a significant advance over the classical word-based model [9]–[11]. Unlike a word-based model that translates an input sentence in a word-by-word manner, phrase-based SMT translates it to an arbitrary word sequence, namely a *phrase*, which is a much larger processing unit than a single word. The success of phrase-based SMT strongly motivates us to enlarge the size of the basic processing unit for Korean morphological analysis given its analogousness to SMT.

Based on these considerations, we propose a novel *discriminative phrase-based model* that considers a "phrase" as the basic processing unit for Korean morphological analysis. In our model, a phrase represents a general processing unit that covers not only a single morpheme but also an eojeol (word), multiple eojeols (words), and any sequence of syllables across the word spacing as a candidate phrase.

Similar to phrase-based SMT, in Korean morphological analysis, considering a phrase as the basic morphological unit can reduce the ambiguity significantly because of its large size, and it can often even provide sufficient evidence for disambiguation without requiring contextual features**. For example, the source phrases "nal-seon nun-bit" (sharp eye) and "ne bang-eu-ro" (to your room) are generally analyzed as "nal/NNG+seo/VV+n/ETM nun-bit/NNG" and "neo/NP + ui/JKG bang/NNG + eu-ro/JKB," respectively, regardless of the surrounding context. These phrase-

Manuscript received March 13, 2017.
Manuscript revised September 10, 2017.
Manuscript publicized November 13, 2017.

†The author is with Dept. of Computer Science, Chonbuk National University, South Korea.
††The author is with Electronics and Telecommunications Research Institute, South Korea.
a) E-mail: nash@jbnu.ac.kr
b) E-mail: kimyk@etri.re.kr

DOI: 10.1587/transinf.2017EDP7085

*The detailed definition of a compound morpheme was presented in [8].

**More precisely, the source phrase corresponds to any syllabic input sentence sequence (limited to the maximum number of syllables) and, the target phrase corresponds to its candidate morphological analysis.

**Table 1** Example of the three processing tasks for Korean morphological analysis for the input sentence "na-neun hag-gyo-e gass-da" (I went to school). In POS tagging and lemmatization, the notation $m/t$ is used for describing individual morphemes (e.g. hag-gyo/NNG), with morpheme $m$ and POS tag $t$. The separator "-" is used between consecutive syllables within an eojeol.

| Input sentence | na-neun hag-hyo-e gass-da |
|---|---|
| Morpheme segmentation | <na, neum, hag-gyo, e, gass, da> |
| POS tagging | <na/NP, neun/JX, hag-gyo/NNG, e/JKB, gass/VV~ EP, da/EF> |
| Lemmatization (restoration of original form) | <na/NP, neun/JX, hag-gyo/NNG, e/JKB, ga/VV, at/EP, da/EF> |

level patterns are becoming increasingly available because the sizes of annotated corpora, including bilingual corpora (which can be used as indirectly annotated data), have been growing rapidly. In contrast, the previous syllable-level and morpheme-level models have to rely on the surrounding contexts to correctly assign POS tags to morphemes due to the relatively large ambiguity they introduced, which requires strong disambiguation components to deal with.

Furthermore, we propose a novel combination method that uses the results of the proposed phrase-based models as *guide* features for the syllable-based conditional random fields (CRFs) model.

Experimental results on the SEJONG dataset show that our phrase-based model is as a promising and effective tool for Korean morphological analysis tasks. Specifically, the combination of our phrase-based model with the CRF model leads to statistically significant improvements over the state-of-the-art syllable-based CRF method.

The remainder of this paper is organized as follows. Section 2 describes other related works, Sect. 3 describes the proposed phrase-based method, and Sect. 4 provides the experimental results. Finally, our concluding remarks and a description of possible future work are given in Sect. 5.

## 2. Related Works

With regards to machine translation, phrase-based models have been studied extensively in the literature on SMT. This has led to the creation of a phrase-based SMT [9], [10], which is a state-of-the-art SMT method. Moses, which is one of the most popular open-source SMT toolkits, originated from phrase-based SMT [11]. It is generally accepted that phrase-based SMT exhibits significant improvements over word-based models. SMT has recently matured even further by incorporating syntax [12]–[16]. However, phrase-based SMT remains one of the best-performing methods, exhibiting performances comparable to those of advanced syntax-based SMT.

The segmentation and tagging problem, which is the problem addressed herein, has been studied extensively in East Asian language analysis. The authors in [17] and [18] applied CRFs to Japanese morphological analysis. In their method, a lattice is first constructed using a lexicon. Then, a Viterbi path is provided over the lattice by using the CRF models, which joins the segmentation and tagging into a single model that is like a semi-CRF [19]. In contrast to a semi-CRF, the method in [18] relies on both a lexicon and additional word processing that is unknown. These components are often unavailable when a fully statistical method is designed. In [20], the authors used a point-wise approach to both word segmentation and POS tagging without the use of a sequential structure. In the point-wise approach, unlike in sequential tagging, classification is performed separately for each input word or character. Two-stage approaches that consist of word segmentation and POS tagging are the most popular approaches to the Chinese language [21], [22]. In Chinese POS tagging, the authors of [23] demonstrated that a character-based tagger outperforms a word-based tagger.

The authors of [3] proposed syllable, morpheme, and eojeol-based models that performed morpheme segmentation, POS tagging, and lemmatization under a probabilistic framework for Korean morphological analysis. However, their models were based mainly on generative models, thus carrying restrictions on the number of features that could be exploited. On the other hand, our phrase-based models are based on a log-linear discriminative model, and the various type of features and sub-models that are introduced by different types of processing units are integrated effectively and in a unified manner.

To the best of our knowledge, no previous studies have extended the basic processing unit of Korean morphological analysis beyond the morpheme or eojeol units. Thus, the present study represents the first use of phrases as generalized processing units in the Korean morphological analysis literature.

The work related most closely to ours is [24], in which the authors proposed a phrase-based POS tagging model and applied it to English POS tagging, resulting in improvements over existing word-based models. However, our model advances the work presented in [24] in the following aspects: 1) we generalize their models further in that the phrase-based model is applied not only to the POS tagging problem from [24] but also to the segmentation and tagging problem, which is, more generally, a Korean morphological analysis task that consists of three sub-tasks, and 2) because of its generalized nature, our model includes two different types of language models (LMs): the morpheme and tag language models, whereas [24] used a tag LM alone.

Note that the underlying idea in [24] of using the phrases as the basic processing unit for an NLP task is the same as that of our work. Thus, our proposed framework is not intended to oppose the method used in [24] but rather to generalize it. To generalize the method of [24] for the segmentation and tagging problem, we redefine a target

phrase as a segment that captures both the segmentation and tagging information, as mentioned in Sect. 3.1. Therefore, the proposed model degenerates the work in [24] precisely when the segmentation information is not included in the target phrase.

Some studies have utilized phrases for other NLP tasks. The authors of [25] used a phrase-based model for dependency parsing, and those of [26] proposed a hybrid approach to phrase- and syntax-based SMTs. However, the tasks in these studies differed from the problems addressed in the present study: the segmentation and tagging problem and Korean morphological analysis.

## 3. Phrase-Based Segmentation and Tagging

In this section, we present the proposed phrase-based model for Korean morphological analysis.

### 3.1 Phrase-Based Segmentation and Tagging

Our phrase-based model is a simple extension of the word-based model. Suppose that an input sentence $\mathbf{x}$ is given as n contiguous characters, $x_1,\cdots,x_n$, where $x_i$ is the $i$-th character of the input sentence. Let $\mathbf{y}=(\mathbf{s},\mathbf{t})$ be a *phrase segmentation* of $\mathbf{x}$, decomposed into $y_1,\cdots,y_m$. Here, $y_i=(\hat{s}_i,\hat{t}_i)$ is called the $i$-th *phrase segment*, in which $\hat{s}_i$ is the $i$-th phrase of the segmentation, and $\hat{t}_i$ is the tag sequence assigned to the $i$-th phrase, called a *tag phrase*.

Adhering to the definition of a phrase used in [9], the phrase $\hat{s}_i$ indicates a sequence of segmented words. For convenience, we refer to $\mathbf{s}$ and $\mathbf{t}$ as $(\hat{s_1},\cdots,\hat{s_m})$ and $(\hat{t_1},\cdots,\hat{t_m})$, respectively. A segment is called a *morpheme-based segment* if its segment phrase is annotated by a single tag, or the unit-length phrase[†]. As a specific case, we call $\mathbf{y}$ a *morpheme-based segmentation*, if $\mathbf{y}$ consists only of morpheme-based segments.

To illustrate our definition, Fig. 1 shows an example Korean sentence, "hak-gyo-e gan-da," that consists of five syllables and means "(I) go to school." Each arc refers to an input syllable of the sentence. The sentence in Fig. 1 consists of two eojeols, "hak-gyo-e" (to school) and "gan-da" (go), and formulates $\mathbf{x}$ as follows:

$$\mathbf{x} = \langle hak, gyo, e, gan, da\rangle$$

Figure 2 shows a morpheme-based segmentation of the example sentence, where each arc denotes a morpheme-based segment. This segmentation $\mathbf{y}_{morph}$ is formulated as follows :

$$\mathbf{y}_{morph} = \langle (hak\text{-}gyo, NNG), (e, JKB), (ga, VV), (n\text{-}da, EF)\rangle$$

Here, each element is a morpheme-based segment $(s,t)$ where $s$ is a morpheme and $t$ is a POS tag. In Fig. 2, the morpheme-based segmentation $\mathbf{y}_{morph}$ is equivalently defined as the path from the starting vertex to the ending vertex. Note that we insert the symbol "*" when a morphological inflection occurs before or after the corresponding morpheme. In the third morphemes "ga*" indicates that a morphological inflection occurs after "ga," and "*n-da" indicates that an inflection occurs before "n-da."

Figure 3 shows a phrase segmentation of the example sentence in which each arc represents a phrase segment. This segmentation $\mathbf{y}_{phrase}$ is formulated as follows:

$$\mathbf{y}_{phrase} = \langle (hak\text{-}gyo{+}e, NNG{+}JKB), (ga{+}n\text{-}da, VV{+}EF)\rangle$$

where "+" indicates a separator between adjacent words. Here, $\mathbf{y}$ consists of larger processing units and a smaller number of elements than $\mathbf{y}_{morph}$.

### 3.2 Log-Linear Model: Scoring Phrase Segmentation

The way the score of a phrase segmentation is defined is a significant issue. Our definition utilizes the conditional probability $\mathbf{y}$ given $\mathbf{x}$ based on a log-linear model as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\prod_k \exp\left(\lambda_k \cdot f_k(\mathbf{x},\mathbf{y})\right) \propto \sum_k \lambda_k f_k(\mathbf{x},\mathbf{y}) \quad (1)$$

where $Z(\mathbf{x})$ is the normalization factor, $f_k(\mathbf{x},\mathbf{y})$ is the $k$-th feature function, and $\lambda_k$ is the feature weight of the $k$-th feature. The goal of segmentation is to find the result that has the largest value of $p(\mathbf{y}|\mathbf{x})$.

We consider six types of feature functions for $f_k(\mathbf{x},\mathbf{y})$:

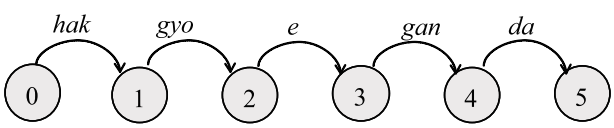


**Fig. 1** Example input sentence "*hak-gyo-e gan-da*," which means "I go to school.".

[†]We assume that a compound morpheme is also a unit morpheme, following the definition of [8].

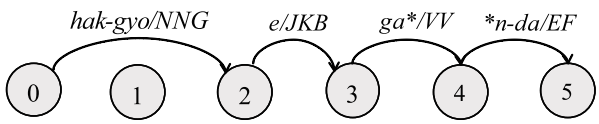


**Fig. 2** Example of a morpheme-based segmentation of the input sentence "*hak-gyo-e gan-da*" which corresponds to $\mathbf{y}_{morph} = \langle (hak\text{-}gyo, NNG), (e, JKB), (ga, VV), (n\text{-}da, EF)\rangle$.

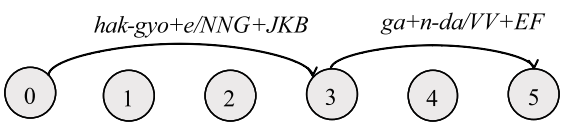


**Fig. 3** Example of a phrase-based segmentation of the input sentence "*hak-gyo-e gan-da*" which corresponds to $\mathbf{y}_{phrase} = \langle (hak\text{-}gyo{+}e, NNG{+}JKB), (ga{+}n\text{-}da, VV{+}EF)\rangle$.

**Table 2** The summarized description of the proposed phrase-based models, comparing to morpheme-based models.

| Type | Model name | Scoring function $score(\mathbf{x}, \mathbf{y})$ | Constraint on phrase length |
|---|---|---|---|
| morpheme-based | M-Model0 | $f_{emit}(\mathbf{x}, \mathbf{y}) + f_{tag\text{-}LM}(\mathbf{x}, \mathbf{y})$ | $\lvert\tilde{s}_i\rvert = \lvert\tilde{t}_i\rvert = 1$ |
| | M-Model1 | $\lambda_{emit} f_{emit}(\mathbf{x}, \mathbf{y}) + \lambda_{p2t} f_{p2t}(\mathbf{x}, \mathbf{y}) + \lambda_{memit} f_{memit}(\mathbf{x}, \mathbf{y}) + \lambda_{tag\text{-}LM} f_{tag\text{-}LM}(\mathbf{x}, \mathbf{y}) + \lambda_{morph\text{-}LM} f_{morph\text{-}LM}(\mathbf{x}, \mathbf{y}) + \lambda_{length} f_{length}(\mathbf{x}, \mathbf{y})$ | |
| phrase-based | P-Model0 | $f_{emit}(\mathbf{x}, \mathbf{y}) + f_{tag\text{-}LM}(\mathbf{x}, \mathbf{y})$ | $\lvert\tilde{s}_i\rvert \geq 1, \lvert\tilde{t}_i\rvert \geq 1$ |
| | P-Model1 | $\lambda_{emit} f_{emit}(\mathbf{x}, \mathbf{y}) + \lambda_{p2t} f_{p2t}(\mathbf{x}, \mathbf{y}) + \lambda_{memit} f_{memit}(\mathbf{x}, \mathbf{y}) + \lambda_{tag\text{-}LM} f_{tag\text{-}LM}(\mathbf{x}, \mathbf{y}) + \lambda_{morph\text{-}LM} f_{morph\text{-}LM}(\mathbf{x}, \mathbf{y}) + \lambda_{length} f_{length}(\mathbf{x}, \mathbf{y})$ | |

1) **The log of the tag-to-phrase translation probability**, which is defined as

$$f_{emit}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} \ln p(\hat{s}_i|\hat{t}_i) \tag{2}$$

where $p(\hat{s}|\hat{t})$ is the phrase emission probability, which is computed via additive smoothing as follows:

$$p(\hat{s}|\hat{t}) = \frac{N(\hat{s}, \hat{t}) + \delta}{N(\hat{t}) + \delta|V(\hat{t})|} \tag{3}$$

where $\delta$ is set to 0.001, $V(\hat{t})$ is the set of different phrases given $\hat{t}$, $N(\hat{t})$ is the count of the tag phrase $\hat{t}$, and $N(\hat{s}, \hat{t})$ denotes the count of the phrase segment $(\hat{s}, \hat{t})$, which is an event in which $\hat{s}$ is the phrase of $\hat{t}$.

2) **The log of the phrase-to-tag translation probability**, which is defined as

$$f_{p2t}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} \ln p(\hat{t}_i|\hat{s}_i) \tag{4}$$

where $p(\hat{t}|\hat{s})$ is the conditional tag probability given the source phrase, computed via additive smoothing as follows:

$$p(\hat{t}|\hat{s}) = \frac{N(\hat{s}, \hat{t}) + \delta}{N(\hat{s}) + \delta|V(\hat{s})|} \tag{5}$$

where $V(\hat{s})$ is the set of different tags given $\hat{s}$ and $N(\hat{s})$ is the count of the phrase $\hat{s}$.

3) **The log of the morpheme-level tag-to-phrase translation**, which is defined as

$$f_{memit}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} \ln p_{morph}(\hat{s}_i|\hat{t}_i) \tag{6}$$

where $p_{morph}(\hat{s}|\hat{t})$ is the morpheme-level emission probability. Suppose that $y = (\hat{s}, \hat{t})$ is a phrase segment with length $k = |\hat{s}| = |\hat{t}|$, with $\hat{s} = s^{(1)}, \cdots, s^{(k)}$ and $\hat{t} = t^{(1)}, \cdots, t^{(k)}$ where $s^{(i)}$ and $t^{(i)}$ indicate $i$-th morpheme and its POS tag, respectively. $p_{morph}(\hat{s}|\hat{t})$ is defined as:

$$p_{morph}(\hat{s}|\hat{t}) = \prod_{i=1}^{|\hat{s}|} p\left(s^{(i)}|t^{(i)}\right) \tag{7}$$

where $p(s^{(i)}|t^{(i)})$ is the morpheme emission probability, which is computed using Eq. (3).

4) **The log-probability of a tag sequence** , which is defined as

$$f_{tag\text{-}LM}(\mathbf{x}, \mathbf{y}) = \ln p(\mathbf{t}) \tag{8}$$

Here, $p(\mathbf{t})$ is estimated at the atomic morpheme level based on the $N$-gram tag language model. Unless otherwise noted, we assume that $N$ is 3.

5) **The log-probability of a morpheme sequence** , which is defined as

$$f_{morph\text{-}LM}(\mathbf{x}, \mathbf{y}) = \ln p(\mathbf{s}) \tag{9}$$

Here, $p(\mathbf{s})$ is estimated at the atomic morpheme level based on the $N$-gram morpheme language model.

6) **The number of phrase segments**, which is defined as

$$f_{length}(\mathbf{x}, \mathbf{y}) = |\mathbf{y}| \tag{10}$$

where $|\mathbf{y}|$ is the number of phrase segments in the segmentation results $\mathbf{y}$. This function plays the role of the bias term that is adjusted to the values of the other feature functions.

Combining the six feature functions, our proposed phrase-based model can be summarized as follows:

$$\begin{aligned} &score(\mathbf{x}, \mathbf{y}) = \\ &\lambda_{emit} f_{emit}(\mathbf{x}, \mathbf{y}) + \lambda_{p2t} f_{p2t}(\mathbf{x}, \mathbf{y}) + \\ &\lambda_{memit} f_{memit}(\mathbf{x}, \mathbf{y}) + \lambda_{tag\text{-}LM} f_{tag\text{-}LM}(\mathbf{x}, \mathbf{y}) + \\ &\lambda_{morph\text{-}LM} f_{morph\text{-}LM}(\mathbf{x}, \mathbf{y}) + \\ &\lambda_{length} f_{length}(\mathbf{x}, \mathbf{y}) \end{aligned} \tag{11}$$

which is hereafter referred to as **P-Model1**.

As a specific case, when $\lambda_{emit} = \lambda_{tag\text{-}LM} = 1$, and all the other feature weights are zeros, Eq. (11) can be simplified to

$$score(\mathbf{x}, \mathbf{y}) = f_{emit}(\mathbf{x}, \mathbf{y}) + f_{tag\text{-}LM}(\mathbf{x}, \mathbf{y}) \tag{12}$$

which corresponds exactly to the "generative" phrase-based model. In this paper, the phrase-based model using (12) is referred to as **P-Model0**.

The morpheme-based model is degenerated from the phrase-based model when all phrases have lengths of 1, meaning that $|\hat{s}_i| = |\hat{t}_i| = 1$. We refer to the morpheme-based model that uses Eq. (11) when $|s^i| = |t^i| = 1$ as M-Model1 and the model that uses Eq. (12) when $|s^i| = |t^i| = 1$ as M-Model0.

Table 2 summarizes the description of the four segmentation and tagging models, M-Model0, M-Model1, P-Model0, and P-Model1. Note that all the features used in our proposed model are generative model features in the sense that their corresponding probabilities are all genera-

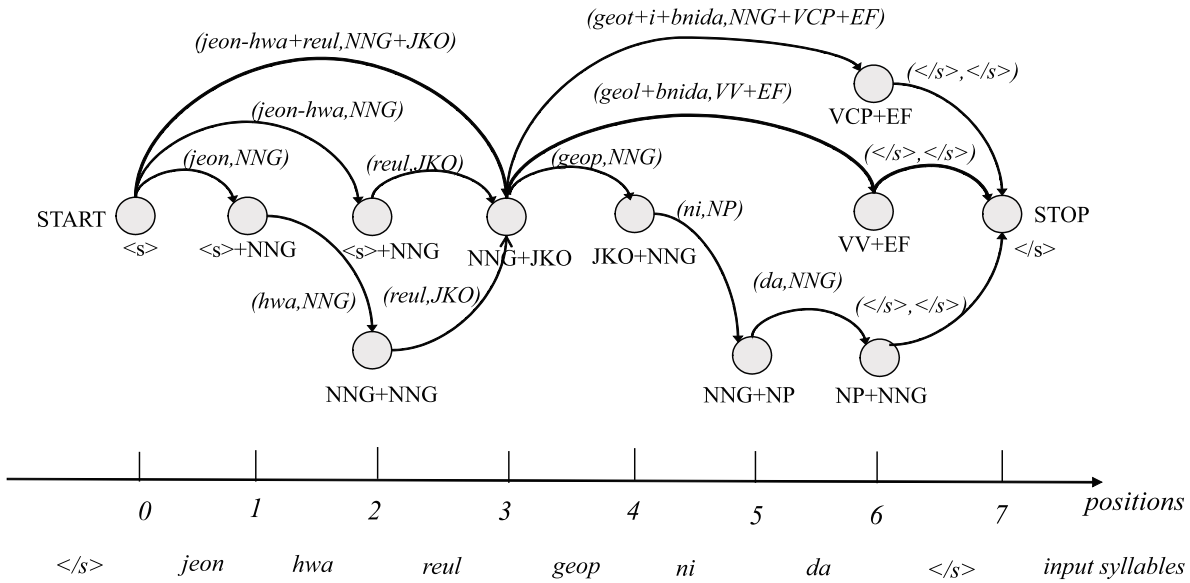


**Fig. 4** An example of the phrase lattice for the input sentence "jeon-hwa-reul geop-ni-da" (calling a phone).

tive individually as components of the joint model $p(\mathbf{s}, \mathbf{t})$.

### 3.3 Decoding

To find the segmentation result that maximizes $score(\mathbf{x}, \mathbf{y})$, we use an extension of the decoding algorithm for the hidden semi-Markov model (HSMM) [27], by taking a phrase lattice as an input. In a phrase lattice, the state of each node is uniquely specified by $(i, \hat{t})$, where $\hat{t}$ represents the last $N-1$ POS tags ending at the $i$-th syllable given this state. Given $n$ input syllables, two special nodes called the START and STOP nodes whose states correspond to $(0, \langle \mathrm{s} \rangle)$ and $(n+1, \langle \backslash \mathrm{s} \rangle)$ are located virtually at the 0-th and $(n+1)$-th syllables, respectively. The score of an edge between two adjacent nodes is computed using the definitions of our feature functions (i.e., Eq. (2), Eq. (4), Eq. (6), Eq. (8), Eq. (9), and Eq. (10)). The score of a path is defined as the summation of the scores of the edges it comprises. The decoding algorithm is designed to find the path that maximizes the score given a phrase lattice with scored edges. Following our definition of the node state, the decoding algorithm can be effectively designed using dynamic programming, which is similar to the Semi-CRF method from [19].

Figure 4 shows an example of the phrase lattice for the input sentence "jeon-hwa-reul geop-ni-da" (calling a phone). Each node is represented as a gray circle. Each node state, corresponding last $N-1$ POS tags, is shown below in the node circle. As shown in Fig. 4, an edge is created for each phrase and its candidate analysis result. The bold path indicates the highest scoring path on the phrase lattice for which the scores of edges are omitted. For the second input eojeol "geop-ni-da," most of the morphological analysis candidates are restricted to "geol/VV + bnida/EF" and "geot/NNB + i/VCP + bnida/EF." These two candidates correspond to the phrase edges in Fig. 4, whose incoming node is $(3, NNG{+}JKB)$ and whose outgoing nodes are $(6, VV{+}EF)$ and $(6, VCP{+}EF)$, respectively.

**Algorithm 1:** Decoding algorithm for finding the best segmentation result that maximizes $score(\mathbf{x}, \mathbf{y})$. $\oplus$ indicates the concatenation operator of two lists and get_last($list$) is a function to return the list with the last $N-1$ elements in $list$. $score(\mathbf{x}, y|lt, ly)$ is the score of the phrase segment $y$ using Eq. (11), given that the last tags and morphemes are $lt$ and $ly$, respectively.

**input** : $\mathbf{x}$: an input sentence that consists of $n$ syllables
**output:** the Viterbi path on phrase lattice
**for** $i \leftarrow 1$ **to** $n+1$ **do**
  **if** $i = n+1$ **then** $j \leftarrow n$; // STOP node $\langle \backslash \mathrm{s} \rangle$
  **else** $j \leftarrow 0$;
  **while** $j < i$ **do**
    **foreach** $u \in N[j]$ **do**
      **foreach** $y = (s, t) \in Y(j+1, i)$ **do**
        $t' \leftarrow u.lt \oplus y.t$
        $y' \leftarrow u.ly \oplus y$
        $edge\_score = score(\mathbf{x}, y|u.lt, u.ly)$ /* using Eq. (11) */
        $score = u.score + edge\_score$
        **if** $i = n+1$ **then**
          $lt' \leftarrow \langle \backslash \mathrm{s} \rangle$
          $ly' \leftarrow (\langle \backslash \mathrm{s} \rangle, \langle \backslash \mathrm{s} \rangle)$
        **else**
          $lt' = \mathrm{get\_last}(t')$
          $ly' = \mathrm{get\_last}(y')$
        $v \leftarrow N[i][lt']$
        **if** $v$ *is None* **or** $v.score < score$ **then**
          $N[i][lt'] = (score, u, i, lt', ly')$
/* Get the Viterbi path by backtracking from $N[n+1][\langle \backslash \mathrm{s} \rangle]$ */

To fully specify our decoding algorithm, let $N[i]$ be the

collection of all the nodes with locations of $i$. By abuse of notation, suppose that $N[i][tag]$ is a specific node whose state is $(i, tag)$. For each node, we need to store its Viterbi path scores, thereby introducing two additional data fields: the best score and the best previous state. Thus, together with the node state information, node $v$ consists of five fields - $(score, prev, i, lt, ly)$: 1) $score$: the score of the Viterbi path terminated at node $v$; 2) $prev$: the best previous node on the Viterbi path of $v$; 3) $i$: the location of the node state; 4) $lt$: the last $N-1$ tags of the node state. 5) $ly$: the last $N-1$ morphemes corresponding to the last $N-1$ tags at $i$-th location.

Now, suppose that an input sentence consists of the $n$ syllables $s[1]\cdots s[n]$. In addition, suppose that $Y(i, j)$ is the set of phrase segments whose source phrase is $s[i]\cdots s[j]$. For notational convenience, a phrase segment $y = (s, t)$ consists of two data fields: 1) $s$: its source phrase, and 2) $t$: its corresponding tag phrase.

Algorithm 1 presents the detailed pseudo code of the decoding algorithm utilized in the present study. In the algorithm, get_last($list$) is a function to return the list with the last $N-1$ elements in $list$.

To achieve a better decoding algorithm efficiency, we introduced the addition of MaxCharLen to avoid searching for extremely long phrases unnecessarily during decoding process, which are extremely rare in the tagged corpus. Unless otherwise specified, the value of MaxCharLen is set to 10.

The time complexity of the decoding algorithm is $O(nLT^{2(N-1)})$ where n is the number of input syllables, $T$ is the number POS tags and $L$ is the maximum phrase length (i.e., MaxCharLen). However, in practice, the value of $T^{2(N-1)}$ is reduced significantly because the last $N-1$ tags of the node state are only restricted in the cases that appear in the phrase table.

### 3.4 Parameter Training Using Lattice MERT

To train the parameters, we applied lattice minimum error rate training (MERT) [28]. In our implementation, the lattice MERT took a phrase lattice as its input. Given a node, all the envelopes of its incoming edges were merged by a sweep line algorithm. In this way, lattice MERT was effectively implemented on top of our decoding algorithm.

We applied the lattice MERT to a development set from which the phrases were not extracted. Rather than using a random direction, we used a coordinate direction for each iteration of the line search. The maximum number of iteration was set at 100. We then applied an early stopping method in which the performance of the development set was not improved until a predetermined value of patience had been achieved, which was set to 15.

### 3.5 Corpus Preprocessing: Automatically Extracting a Compound Morpheme

The original Korean tagged corpus was not composed of compound morphemes and had only the input eojeol and its list of atomic morphemes. To extract the compound morphemes as previously defined, we converted the original POS-tagged corpus to a form in which all its morphemes were segmented clearly at the syllable level without considering alphabet-level decomposition. All morphological inflections were then removed after the conversion. Finally, the converted POS-tagged corpus was used for then phrase extraction. The detailed preprocessing for this process was described in Sect. 3.2.1 of [8]†.

### 3.6 Phrase Extraction

At this point, the remaining unsolved problem is determining how to extract phrases from the training corpus. In our Korean morphological analysis, the phrase extraction procedure is much simpler than the SMT problem because each word is already aligned to a POS tag. Therefore, the application of a word alignment procedure is unnecessary, although it is necessary for phrase extraction in SMT.

To extract phrases from the tagged corpus, we specified a value for MaxNumMorphs, which indicates the maximum number of morphemes of a phrase. All sequences of morphemes and corresponding tags with lengths between 1 and MaxNumMorphs were extracted as phrases. For example, consider that "na-neun hak-gyo-e gan-da" (I am going to school) is given as the input sentence and its analysis result is given by:

| Eojeol | POS tagged results | Lemmatized results |
|---|---|---|
| na-neun | na/NP+neun/JX | – |
| hak-gyo-e | hak-gyo/NNG+e/JKB | – |
| gan-da | gan-da/VV~EF | ga*/VV+ *nda/EF |

where "gan-da/VV~EF" (going) is a compound morpheme that consists of the two morphemes "ga/VV" (go) and "nda/EF" (present tense) based on the definition in [8]. From these results, when MaxNumMorphs = 3, we obtained a subset of the phrases extracted from the tagged sentence, which are listed in Table 3.

Note that in phrase extraction, we considered a compound morpheme as the basic processing unit. Thus, "gan-da/VV~EF,';' whose original result was "ga*/VV + *nda/EF," was counted as a single morpheme.

In addition, when extracting a phrase, if there was a word space on the left or right boundary of the phrase, we extracted a new phrase that included the word space, while

†The extraction of compound morphemes is the process of matching original lemmas with their surface forms. In this type of matching, the uses of researcher-designed rules and dictionaries are possible for preprocessing a tagged corpus. The main reason that we use automatic processing is that our goal is to build an end-to-end Korean morphological analysis that trains all the knowledge from the given training corpus and relies only minimally on external dictionaries and resources. Another reason is that researcher-designed rules may not cover all irregular cases. To deal with such cases, an automatic procedure will be necessary.

**Table 3** Example phrases extracted from the tagged sentence "na/NP+neun/JX hak-gyo/NNG+e/JKB ga*/VV+*nda/EF" when the maximum number of words is fixed at 3.

| Phrase | Phrase segments | Number of morphs |
|---|---|---|
| na | na/NP | 1 |
| na-neun | na/NP+neun/JX | 2 |
| na-neun hak-gyo | na/NP+neun/JX+hak-gyo/NN | 3 |
| neun | neun/JX | 1 |
| neun hak-guo | neun/JX+hak-gyo/NNG | 2 |
| neun hak-guo-e | neun/JX+hak-gyo/NNG+e/JKB | 3 |
| ... | ... | ... |
| hak-guo-e gan-da | hak-guo/NNG+e/JKB+ga*/VV+*nda/EF | 3 |
| e gan-da | e/JKB+ga*/VV+*nda/EF | 2 |
| gan-da | ga*/VV+*nda/EF | 1 |

the target side remained changed.

## 4. Experimentation

### 4.1 Experimental Setting

To evaluate the proposed phrase-based model, we used a POS-tagged corpus called SEJONG. Table 4 presents an overview the basic statistics for the SEJONG regional corpus. To allow the program to learn the six types of features used in the proposed method, we separated the training set from the test set for each tagged corpus and used 80% of the sentences for training and the remaining 20% for testing. In training sentences, 1,000 sentences were selected for training parameters using lattice MERT.

For evaluation, we generally used the F-measure at the morpheme level and the accuracy at the eojeol level. In the following, we provide the details of these evaluation measures.

- **F-measure** (at the morpheme level): This measurement is computed at the morpheme level and compares how many morphemes are segmented or tagged correctly, either among the system outputs or the answer morphemes. The precision and recall, which are used for computing the F-measure, can be defined as follows[†]:
  - *Precision* = the number of correctly matched morphemes generated by the system / the total number of morphemes generated by the system
  - *Recall* = the number of correctly matched morphemes generated by a system / the total number of morphemes in a tagged corpus
- **Accuracy** (at the eojeol level): This metric indicates how many eojeols are analyzed correctly. The result is considered incorrect when it contains an error at the morpheme level. The accuracy is defined as follows:
  - Accuracy = the number of correctly analyzed eojeols / the total number of eojoels

All evaluations were conducted on the results obtained after performing either segmentation/tagging or the all steps of morphological analysis.

[†]Here, we say that a morpheme is matched correctly in the sense that a morpheme is correctly segmented and tagged.

**Table 4** Data statistics of the corpora used for the experiment.

| Corpus | SEJONG |
|---|---|
| Num of tags | 42 |
| Total num of sents | 253,139 |
| Total num of eojeols | 3,466,378 |
| Total num of morphemes | 7,698,715 |
| Total num of syllables | 30,925,184 |
| Average num of eojeols per sent | 13.69 |
| Num of sents for training | 201,508 |
| Num of sents for development | 1,000 |
| Num of sents for test | 50,631 |

### 4.2 Morpheme-Based Model vs. Phrase-Based Model

For building tag and word trigram language models, we used BerkeleyLM with KN smoothing [29]. In our first evaluation, we compared morpheme-based models with the proposed phrase-based models. As summarized in Table 2, the morpheme-based model is the specific case of the phrase-based model using a MaxNumMorphs of 1. For phrase-based models, MaxNumMorphs was set to 3. We chose M-Model0 and P-Model0 for the morpheme- and phrase-based models, respectively, without using the morpheme LM component (i.e., $\lambda_{emit} = \lambda_{tag\text{-}LM} = 1$).

Table 5 shows a comparison of the results between the morpheme- and phrase-based models. In the Model0 case, the phrase-based model outperformed the morpheme-based model significantly, with an eojeol accuracy performance difference of approximately 3%. Because only two generative features were used for Model0, the results demonstrated the power of using a phrase in the context of Korean morphological analysis.

In the Model1 case, all feature functions were used, and the parameters for M-Model1 and P-Model1 were trained using lattice MERT, as described in Sect. 3.4. The results showed that the uses of all feature functions with parameter training significantly improved M-Model0.

Although the performance differences were reduced, the phrase-based model still led to improvements over the morpheme-based model, increasing eojeol accuracy by almost 1% of, which is considered as a relatively large margin in the literature. Interestingly, P-Model0, the naive phrase-based model with only two basic features, demonstrated a comparable performance to that of M-Model1 (the full-

**Table 5** Comparison of morpheme- and phrase-based models on the Korean morphological analysis task using SEJONG.

| | F-measure (at morpheme level) | Eojeol accuracy |
|---|---|---|
| M-Model0 | 93.49% | 90.33% |
| P-Model0 | 95.97% | 93.94% |
| M-Model1 | 95.96% | 93.87% |
| P-Model1 | **96.49**% | **94.77**% |

fledged version of the morpheme-based model).

Overall, the results show the power using a phrase in the context of Korean morphological analysis in the contexts of both generative and full-fledged settings.

### 4.3 Effects of Varying Maximum Number of Morphs in a Phrase

In further experiments, we varied the maximum number of words in a phrase (MaxNumMorphs), and reevaluated the proposed model. The maximum number of characters in a phrase (MaxCharLen) was set to 10. In the preliminary experiment, when using a larger value for MaxCharLen, we did not observe any significant differences.

Table 6 shows the performances of the phrase-based models obtained by using our approach for SEJONG and varying MaxNumMorphs. The results for MaxNumMorphs values of 1 and 3 correspond to M-Model0 and P-Model0 in Table 5, respectively.

Overall, when MaxNumMorphs $\leq$ 3, larger value for MaxNumMorphs gradually increased the performance. Specifically, the performance differences between the models using MaxNumMorphs = 1 and MaxNumMorphs = 2 was the largest among the models. This result implied that the performance of the phrase-based model relied primarily on the improvement gained using MaxNumMorphs = 2. When MaxNumMorphs $\geq$ 3, the performance difference among the different values for MaxNumMorphs was slight.

In another comparison, Table 7 shows the total number of phrases in the phrase table extracted from the entire training corpus when MaxNumMorphs ranged from 1 to 5. As MaxNumMorphs increased, the total number of phrases increased sharply, exhibiting an almost exponential tendency. The greater the total number of phrases used, the higher the time and space complexities generated. These results strongly suggest that the value MaxNumMorphs should remain low if it does not degrade the effectiveness of the model significantly. Based on the results shown Tables 6 and 7, a MaxNumMorphs value of 3 is recommended based on its relatively high effectiveness and efficiency.

### 4.4 Effects of Varying Maximum Character Length of Phrase

Table 8 shows the performances of the phrase-based model obtained from using our approach for SEJONG with a MaxNumMorphs of 3 and varying MaxCharLen. When MaxCharLen was lower than 4, the performance of the model was poor. When MaxCharLen $\geq$ 6, there were no significant differences in the performance gain for larger values of MaxCharLen.

**Table 6** Performances of phrase-based model obtained using our approach for SEJONG and varying MaxNumMorphs.

| P-Model1 | F-measure (at morpheme level) | Eojeol accuracy |
|---|---|---|
| MaxNumMorphs = 1 | 95.96% | 93.87% |
| MaxNumMorphs = 2 | 96.36% | 94.55% |
| MaxNumMorphs = 3 | **96.49**% | **94.77**% |
| MaxNumMorphs = 4 | 96.45% | 94.71% |
| MaxNumMorphs = 5 | 96.43% | 94.65% |

**Table 7** Total number of phrases in phrase table extracted from entire training corpus.

| P-Model1 | Total number of phrases in phrase table |
|---|---|
| MaxNumMorphs =1 | 248,705 |
| MaxNumMorphs=2 | 2,412,570 |
| MaxNumMorphs =3 | 7,783,503 |
| MaxNumMorphs =4 | 14,954,935 |
| MaxNumMorphs =5 | 22,785,594 |

**Table 8** Performances of phrase-based model obtained by using our approach for SEJONG and varying MaxCharLen.

| | F-measure (at morpheme level) | Eojeol accuracy |
|---|---|---|
| MaxCharLen =3 | 85.00% | 79.81% |
| MaxCharLen =4 | 94.21% | 92.11% |
| MaxCharLen =5 | 96.05% | 94.33% |
| MaxCharLen =6 | 96.34% | 94.64% |
| MaxCharLen =7 | 96.42% | 94.71% |
| MaxCharLen =8 | 96.46% | 94.75% |
| MaxCharLen =10 | **96.49**% | **94.77**% |

### 4.5 Combining Proposed Model with CRF-Based Morpheme Segmentation and Tagging

To improve their performances further, we combined our phrase-based statistical models with the existing CRF-based model for Korean morpheme segmentation and tagging [8], which demonstrated one of the highest performances in the literature. The combination of our model with the CRF-based model was focused on feature-based integration [30], which was originally used for integrating graph- and transition-based methods in dependency parsing. Of the two components of the integrated model in [30], the *base model* corresponded to a CRF-based model and the *guide model* became our phrase-based model. The process of extracting the guide features from the guide model was as follows.

1) Perform morpheme segmentation and POS tagging using the guide model (phrase-based model) for a given input statement.

2) Attach label {B, I} to the result of Step 1 and convert it into a sequence of syllable tags. For example, suppose that the input Korean sentence is "hak-ko-e gass-da" (I went to school) and its analysis result is ⟨"hak-ko/NNG",

"e/JKB", "gass/VV~EP", "da/EF"⟩. Then, the resulting sequence of syllable tags is given by ⟨'B-NNG, I-NNG, B-JKB, B-VV~EP, B-EF⟩.

3) Extract guide features from tags surrounding sequence of syllable tags. Let $G_i$ ($G_{-i}$) be the next (previous) $i$-th syllable tag at the current position. Table 9 summarizes the types of guide features.

Finally, the guided model is a CRF-based model that learns using the basic features of [8] and guide features in Table 9.

Table 10 shows a comparison of the performance of the proposed combined method with that of revised CRF-based method, including its original performance in [8]. The second row named "CRF (revised)" indicates the revised version of the original CRF method of [8] by extending the definition of compound morpheme to include nominal morphemes [8]. As can be seen in Table 10, the proposed combined method can improve the performance of the existing CRF-based method even further.

To check whether the improvements provided by the combined method were statistically significant, we introduced a macro version of the evaluation measures for the F-measure and eojeol accuracies. The macro F1 refers to the average of the sentence-level F1 scores over all the test sentences. Similarly, the macro eojeol accuracy is defined as the average of the sentence-level eojeol accuracies over all the test sentences.

Table 11 shows the performance comparison for the proposed combined method and the CRF method at the macro level. We applied the t-test for statistical significance and appended * to the performance numbers of the combined methods. As shown in Table 11, the improvements provided by our combined methods are statistically significant.

In Table 11, we also report the "oracle" performance of the combined method in the row marked "oracle." The oracle performance means the performance of the virtual method that always selects the better of the P-Model1 and CRF methods for each input sentence. The oracle results in Table 11 imply that there is a lot of room for P-Model1 to make further improvements, both in its F1 score and in its eojeol accuracy for the final combination. There are non-trivial number of sentences that are correctly analyzed by P-Model1 but incorrectly analyzed by the CRF method. Both extensions of the phrase-based models and a better approach to combining the proposed model with the CRF method to maximize the effect of phrase-based evidence are worthy topics for future investigation.

**Table 9** The guide features guided by the phrase-based model of [8]

| Feature symbol | Description |
|---|---|
| $G_{-2}$,$G_{-1}$,$G_0$,$G_1$,$G_2$ | Uni-syllable tag information |
| $G_{-1}G_0$,$G_0G_1$,$G_1G_2$ | Bi-syllable tag information |
| $G_{-2}G_{-1}G_0$,$G_{-1}G_0G_1$, $G_0G_1G_2$ | Tri-syllable tag information |

**Table 10** Performance comparison between existing CRF-based method and combined method (measured in compound morpheme unit).

| Method | F-Measure | Eojeol accuracy | Sentence accuracy |
|---|---|---|---|
| CRF [8] | 97.60% | 96.14% | 64.40% |
| CRF (revised) | 97.63% | 96.18% | 64.74% |
| P-Model1+CRF | **97.74**% | **96.35**% | **65.68**% |

**Table 11** Comparison of performances between CRF-based method and combined method at macro level (measured in compound morpheme units). * indicates that performance difference is statistically significant in the t-test at 0.99 confidence level, and "oracle" in last row indicates optimal performance based on the better of P-Model1 and CRF methods for each sentence.

| Method | F-Measure | Eojeol accuracy | Sentence accuracy |
|---|---|---|---|
| CRF (revised) | 97.65% | 96.10% | 64.40% |
| P-Model1+CRF | **97.76**%* | **96.29**%* | **65.68**%* |
| P-Model1+CRF (oracle) | 98.42% | 97.20% | 72.13% |

## 4.6 Comparison with Other Systems

Finally, we compared the proposed method with other systems. Table 12 shows the performance results of some existing systems along with the performance results of our method on SEJONG. Because we do not have our own implementations of these systems, we only provide their performance numbers. They are therefore only meaningful as references and are not directly comparable, as the training/test sets for each system differ (even the same types of datasets are used). Nevertheless, the proposed system demonstrates performances comparable to those of other state-of-the-art methods and was even able to achieve higher eojeol accuracies on the same types of dataset.

## 5. Conclusion

In this paper, we proposed the use of phrases for the problem of segmentation and tagging and presented a novel phrase-based segmentation and tagging model. Like the concept

**Table 12** Comparison of results of our method to those of other systems (reported for different evaluation sets).

| | Corpus | # of tags | F-measure | Eojeol accuracy | Sentence accuracy |
|---|---|---|---|---|---|
| P-Model1 | SEJONG | 42 | 96.49% | 94.77% | 57.21% |
| P-Model1+CRF | SEJONG | 42 | **97.74**% | **96.35**% | **65.68**% |
| [31] | SEJONG | 42 | 92.96% | 87.92% | N/A |
| [3] | KAIST | 52 | N/A | 91.49% | N/A |
| [3] | SEJONG | 42 | N/A | 92.32% | N/A |

of a phrase in SMT, our definition of a phrase is not linguistically motivated but refers to a sequence of input syllables. Experiment results of a Korean morphological analysis show that our phrase-based model is a promising model that does not require linguistic rules or external dictionaries. Although we address Korean morphological analysis, several tasks in natural language processing are instances of the segmentation and tagging problem. In future works, we intend to apply phrase-based models to other tasks such as named entity recognition and chunking. We also plan to improve phrase-based models further by adding more features and exploring a better approach to combining phrase-based models with CRF methods.

## Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT)(R7119-16-1001, Core technology development of the real-time simultaneous speech translation based on knowledge enhancement).

### References

[1] S.S. Kang, Korean Morphological Analysis using Syllable Information and Multi-word Unit Information, Ph.D. thesis, Seoul National University, 1993.
[2] G.G. Lee, J. Cha, and J.-H. Lee, "Syllable-pattern-based unknown-morpheme segmentation and estimation for hybrid part-of-speech tagging of korean," Computational Linguistics, vol.28, no.1, pp.53–70, 2002.
[3] D.G. Lee and H.C. Rim, "Probabilistic modeling of korean morphology," IEEE Trans. Audio Speech & Language Process., vol.17, no.5, pp.945–955, 2009.
[4] J.S. Lee, "Three-step probabilistic model for korean morphological analysis," J. Korean Information Science Society: Software and Applications, vol.38, no.5, pp.257–268, 2011 (in Korean).
[5] K. Shim, "Syllable-based pos tagging without korean morphological analysis," J. Korean Society for Cognitive Science, (in Korean), vol.22, no.3, pp.327–345, 2011.
[6] S.H. Na, S.I. Yang, C.H. Kim, O.W. Kwon, and Y.K. Kim, "Crfs for korean morpheme segmentation and pos tagging," 24th Annual Conference on Human and Cognitive Language Technology (HCLT '12), 2012 (in Korean).
[7] C. Lee, "Joint models for korean word spacing and pos tagging using structural SVM," J. Korean Information Science Society: Software and Applications, vol.40, no.12, pp.826–832, 2013 (in Korean).
[8] S.H. Na, "Conditional random fields for korean morpheme segmentation and pos tagging," ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol.14, no.3, pp.10:1–10:16, 2015.
[9] P. Koehn, F.J. Och, and D. Marcu, "Statistical phrase-based translation," Proc. 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pp.48–54, 2003.
[10] F.J. Och and H. Ney, "The alignment template approach to statistical machine translation," Computational Linguistics, vol.30, no.4, pp.417–449, 2004.
[11] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," annual meeting of the association for computational linguistics (acl), ACL (demonstration session), 2007.
[12] D. Chiang, "Hierarchical phrase-based translation," Computational Linguistics, vol.33, no.2, pp.201–228, 2007.
[13] L. Huang, H. Zhang, D. Gildea, and K. Knight, "Binarization of synchronous context-free grammars," Computational Linguistics, vol.35, no.4, pp.559–595, 2009.
[14] M. Zhang, H. Jiang, A.T. Aw, J. Sun, S. Li, and C.L. Tan, "A tree-to-tree alignment-based model for statistical machine translation," MT Summit, 2006.
[15] H. Mi, L. Huang, and Q. Liu, "Forest-based translation," ACL, pp.192–199, 2008.
[16] Z. Tu, Y. Liu, Y.s. Hwang, Q. Liu, and L. Shouxun, "Dependency forest for statistical machine translation," COLING, 2008.
[17] T. Kudo, "Mecab: yet another part-of-speech and morphological analyzer," 2006.
[18] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to japanese morphological analysis," Proc. 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP '04, pp.230–237, 2004.
[19] S. Sarawagi and W.W. Cohen, "Semi-markov conditional random fields for information extraction," Eighteenth Annual Conference on Neural Information Processing Systems, NIPS '04, 2004.
[20] G. Neubig, Y. Nakata, and S. Mori, "Pointwise prediction for robust, adaptable japanese morphological analysis," Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers, ACL-HLT '11, vol.2, pp.529–533, 2011.
[21] F. Peng, F. Feng, and A. McCallum, "Chinese segmentation and new word detection using conditional random fields," Proc. 20th international conference on Computational Linguistics, COLING '04, 2004.
[22] N. Xue, "Chinese word segmentation as character tagging," Int. J. Computational Linguistics and Chinese Language Processing, vol.8, no.1, 2003.
[23] H.T. Ng and J.K. Low, "Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based?," Proc. 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP '04, pp.277–284, 2004.
[24] A. Finch and E. Sumita, "Phrase-based part-of-speech tagging," Natural Language Processing and Knowledge Engineering, 2007.
[25] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase dependency parsing for opinion mining," EMNLP '09, pp.1533–1541, 2009.
[26] K. Gimpel and N.A. Smith, "Phrase dependency machine translation with quasi-synchronous tree-to-tree features," Computational Linguistics, vol.40, no.2, pp.349–401, 2014.
[27] S.Z. Yu, "Hidden semi-markov models," Artifi. Intelli., vol.174, no.2, pp.215–243, 2010.
[28] W. Macherey, F.J. Och, I. Thayer, and J. Uszkoreit, "Lattice-based minimum error rate training for statistical machine translation," Proc. Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pp.725–734, 2008.
[29] The Berkeley LM toolkit, https://code.google.com/archive/p/berkeleylm/
[30] J. Nivre and R. Mcdonald, "Integrating graph-based and transition-based dependency parsers," ACL-08: HLT, pp.950–958, 2008.
[31] J.P. Hong and J.W. Cha, "A new korean morphological analyzer using eojeol pattern dictionary," Proc. 15th conference on Computational linguistics, COLING '94, vol.1, pp.535–539, 1994.

**Seung-Hoon Na** received his Ph.D. degrees in Computer Science from POSTECH in 2008. Currently, he is an assistant professor in Dept. of Computer Science at Chonbuk National University, Previously, he was a senior researcher at Electronics and Telecommunications Research Institute, South Korea, after he worked in School of Computing at Natural University of Singapore. His research interests include natural language processing, information retrieval, and machine learning.

**Young-Kil Kim** received his Ph.D. degrees in Computer Science from Hanyang University, in 1997. Currently, he is a principal researcher at Electronics and Telecommunications Research Institute, South Korea. His research interests include natural language processing, dialogue processing, and machine translation.