*Natural Language Engineering* **7** (2): 167–185. Printed in the United Kingdom

© 2001 Cambridge University Press

# *Compound noun segmentation based on lexical data extracted from corpus*

JUNTAE YOON

*IRCS, University of Pennsylvania, 3401 Walnut St., Suite 400A, Philadelphia, PA 19104-6228, USA*
*e-mail:* jtyoon@linc.cis.upenn.edu

(*Received 11 November 1999; revised 5 December 2000*)

**Abstract**

Compound noun segmentation is one of the crucial problems in Korean language processing because a series of nouns in Korean may appear without space in real text, which makes it difficult to identify its morphological constituents. This paper presents an effective method of Korean compound noun segmentation based on lexical data extracted from a corpus. The segmentation consists of two tasks: First, it uses a Hand-Build Segmentation Dictionary (HBSD) to segment compound nouns which frequently occur or need an exceptional process. Second, a segmentation algorithm using data from a corpus is proposed, where simple nouns and their frequencies are stored in a Simple Noun Dictionary (SND) for segmentation. The analysis is executed based on modified tabular parsing using min-max operation. Our experiments have shown a very effective accuracy rate of about 97.29%, which turns out to be very effective.

## 1 Introduction

Morphological analysis is crucial for processing an agglutinative language such as Korean, since words in such a language have a lot of morphological variants. In Korean, a sentence is represented with a sequence of *eojeol*, which are the syntactic units delimited by spacing characters. An eojeol consists of a lexical morpheme[1] and functional morphemes. This spacing rule in Korean alleviates the difficulty of word segmentation for a sentence which is necessarily required in other oriental languages such as Chinese and Japanese. However, an eojeol in real text often contains one or more nouns combined, although one lexical morpheme must be included in an eojeol by the spacing rule. Thus, we cannot get a proper

[1] A morpheme in general is a minimal meaningful unit, but it is sometimes taken with a broader meaning for computational convenience in Korean language processing. For instance, *bunseoggi* (analyzer) which is composed of a noun *bunseog* (analysis) and a suffix *gi* (system) is treated as one morpheme. In addition, a Korean proper noun *seol'agsan* (Seol'ag Mountain) which corresponds to two words in English is registered as one entry in most morphological analysis dictionaries. The term *morpheme*, which could be referred to as *word*, has been used by most Korean researchers, since the main task of the analysis is to separate functional morphemes from a lexical morpheme in an eojeol.

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

interpretation of a sentence or phrase without their accurate segmentation. As a result, compound noun segmentation is an important issue in Korean morphological analysis.

The problem of compound noun segmentation in Korean comes from the fact that it is not possible to register all compound nouns in a dictionary. The number of them is huge as nouns are in the open set of words, which means that they must be treated as unseen words without a segmentation process. Furthermore, accurate compound noun segmentation plays an important role in application systems. Compound noun segmentation is needed to improve recall and precision in Korean Information Retrieval (IR) and to obtain better translations in Machine Translation (MT). For example, suppose that a compound noun '*seol'agsangugribgongwon* (*seol'agsan, gugrib, gongwon*; Seol'ag Mountain National Park)' appears in documents. If a user wants to get documents about '*seol'agsan* (Seol'ag Mountain)', then it is likely that the documents with '*seol'agsangugribgongwon*' are also of interest, as Seol'ag Mountain and Seol'ag Mountain National Park are closely related. Therefore, it should be exactly segmented before indexing for the documents to be retrieved for the query '*seol'agsan*'. Also, to translate '*seol'agsangugribgongwon*' to Seol'ag Mountain National Park in MT, the constituents must be identified first through the segmentation process.

In this paper, we propose a new method for segmentation of Korean compound nouns. The compound noun segmentation is conducted in two stages: (1) using a hand-built segmentation dictionary; and (2) applying a segmentation algorithm. For the construction of an HBSD, we first extract compound noun candidates from a large corpus, and manually divide them into simple nouns. The dictionary includes built-in segmentations for compound nouns which are frequently used or need an exceptional process. The number of compound nouns in the HBSD is about 100,000.

Next, the segmentation algorithm is applied if the compound noun does not exist in the HBSD. Basically, the segmenter uses the frequency of individual nouns. One problem with this process is that a lot of lexical information is required for compound noun segmentation. As first glance, we could use a Part Of Speech (POS) tagged corpus to obtain the lexical information, but it is not possible to get sufficient statistics of words due to its small size. A raw corpus is also inappropriate since the data obtained may contain a lot of incorrect words. Thus, we do not expect good results with lexical information from a small POS tagged corpus and a raw corpus just depending on auto tagging.

To get lexical information used in compound noun segmentation, we propose a method to acquire simple nouns and their frequencies from a subset of a raw corpus, which consists of frequently occurring eojeols based on the repetitiveness of natural language. The number of eojeols that we extracted is manually tractable, from which we obtain frequency-used nouns that are crucial for compound noun segmentation. Furthermore, we propose *min-max* composition as a metric to divide a sequence of syllables in a compound noun. The *min-max* operation is a main compositional rule of inference used in fuzzy logic (Zadeh 1965; Fukami, Mizumoto and Tanaka 1980). To briefly show why we selected this operation, let us consider the following example. Suppose that a compound noun is composed of four Korean

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

syllables '$s_1s_2s_3s_4$', for example '*haggyosaenghwal*' comprising syllables *hag, gyo, saeng* and *hwal*. While there are several possibilities of segmentation in the sequence of syllables, we consider the possibilities $s_1 + s_2s_3s_4$[2] and $s_1s_2 + s_3s_4$. Assume that $s_1$ is a frequently appearing word in text, whereas $s_2s_3s_4$ is a rarely occurring sequence of syllables. On the other hand, $s_1s_2$ and $s_3s_4$ occur frequently, although they do not occur as frequently as $s_1$. In this case, the more likely segmentation would be $s_1s_2 + s_3s_4$. It means that a sequence of syllables tends to be divided into two frequent sequences of syllables, rather than very frequently and rarely occurring ones. In this sense, min-max is the appropriate operation for the selection. In other words, a min value is selected between two sequences of syllables, and then max is taken from min values selected. To apply the operation repetitively, we use the CYK (Cocke–Younger–Kasami) tabular parsing style mechanism (Kasami and Torii 1965; Younger 1967).

Our discussion will proceed as follows. In section 2, we briefly mention previous works related to word segmentation. Section 3 discusses construction of the HBSD. In addition, we give a method for lexical knowledge acquisition for segmentation from a corpus. In section 4, we describe the segmentation algorithm using the lexical information extracted. Finally, in section 5 we show experimental results, and discuss the evaluation of our system.

## 2 Related works

In oriental language processing, there has been much work done on word segmentation, since it is crucial as the first step. Although some of the work done was remarkably successful, it is still a very difficult and challenging task (Chen and Liu 1992; Sproat, Shih, Gale and Chang 1994; Li and Wang 1995; Yun, Cho and Rim 1997).

Chen and Liu (1992) used a dictionary to identify words in a sentence. Their method is useful for finding known words, but was not applied to the identification of unknown words. In addition, another problem with the dictionary-based approach is that it does not present a proper measure for selecting the best result from several possibilities. Li and Wang (1995) proposed a Chinese word segmentation algorithm using a raw corpus. They paid attention to the fact that frequently occurring common strings in a corpus are likely to be words, and thus used the common strings to segment a sentence into words. However, since simple nouns for compound noun segmentation are not effectively extracted with this approach, the performance has negative results when similarly applied to Korean compound noun segmentation (Yoon, Lee and Choi 1999).

Sproat *et al.* (1994) used a stochastic finite state model to segment a Chinese sentence into words. In their research, words are represented with the finite state transducer. Given a sentence, the possible segmentations are expressed as a form of lattice. Then the task of the segmenter is to find the most likely segmentation from

[2] We use the character '+' to represent the combination of sequences of syllables.

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

the possibilities. Specifically, they tried to treat derived words, person names and transliterated foreign words in Chinese, which was reported to have good results. One problem of applying this method to our work is to select nouns for segmentation. They used a dictionary estimated from a large corpus in order to segment sentences. In general, too many dictionary entries give negative results in segmentation since nouns which should not be segmented are often segmented because rarely occurring words commonly take a part of sequences of syllables which are not segmented. Thus, the rarely used words cause over-segmentation, which has a bad influence on morphological analysis. In this paper we propose an effective method to select such simple nouns and their frequencies.

What should be noted is that the Korean language has a different orthographical rule from Chinese and Japanese, in that whitespace is used as a delimiter. A sentence in Korean is a sequence of eojeols which are delimited by whitespace. Accordingly, the main task of Korean morphological analysis is not to divide a sentence into words, but to segment an eojeol into its constituent morphemes. As mentioned before, an eojeol consists of a lexical morpheme and functional morphemes, which means that two lexical morphemes are separated as different eojeols by whitespace. Nouns, however, are allowed to be included in one eojeol without space, and the lexical part with nouns combined is a compound noun. By analyzing the compound noun, we can obtain individual nouns from it.

The best performance reported on work of compound noun segmentation in Korean is given by Yun, Cho and Rim (1997). They first selected featuring words from example compound nouns extracted from a dictionary. A featuring word is one that is frequently found in the prefix and suffix part of compound nouns. They used the featuring words to make an inference of the constituents in the compound noun, and combined the segmentation algorithm and heuristic rules for accurate estimation of uncertain segmentation. Their findings were fairly good, but one problem of the work was that they restricted the length of the compound noun because of the computation complexity. Another problem was the test set. As mentioned in Sproat *et al.* (1994), since the rules of segmentation are different from each other, a standardized test set such as ones extracted from a POS tagged corpus is required.

In this work, we present a generalized and more accurate segmentation method, with no relation to the length of compound noun, and conduct experiments using the gold standard test set extracted from a POS tagged corpus.

## 3 Lexical data acquisition

Since a compound noun consists of a series of nouns, a model using transition probability among parts of speech is not helpful. Instead, lexical information is required for compound noun segmentation. Our segmentation algorithm is based on a large collection of lexical information that consists of two kinds of data: one is a Hand-Built Segmentation Dictionary (HBSD), and the other is a Simple Noun Dictionary (SND) for segmentation.

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

Table 1. *Examples of compound noun and analysis information in HBSD*

| Compound nouns | Analysis information |
|---|---|
| *gajuggudu*(leather shoes) | *gajug*(leather)+*gudu*(shoes) |
| *gajugggeun*(leather string) | *gajug*(leather)+*ggeun*(string) |
| *gaguyong*(used for furniture) | *gagu*(furniture)+x*yong*(used for) |
| *sagwassi*(apple seed) | *sagwa*(apple)+n*ssi*(seed) |
| *podossi*(grape seed) | *podo*(grape)+n*ssi*(seed) |
| *chuggutim*(football team) | *chuggu*(football)+*tim*(team) |

### *3.1 Hand-built segmentation dictionary*

The HBSD is used in the first phase of compound noun segmentation. The advantage of using the built-in dictionary is that the segmentation could (1) be very accurate using hand-made data, and (2) become more efficient. In Korean compound nouns, nouns of one syllable are sometimes highly ambiguous between suffix and noun, but nonetheless, humans can easily identify them using semantic knowledge. For example, one syllable noun '*ssi*' in Korean is used either as a suffix or as a noun, which means 'Mr/Mrs' and 'seed', respectively. Without any semantic information, the best way to distinguish them is to record all the compound noun examples containing the meaning of seed in the dictionary, since use of the noun which means 'seed' is much more restricted in compound nouns. Besides, we can treat general spacing errors using the dictionary. Although there should be one lexical morpheme in an eojeol, except for a noun by the spacing rule for Korean, it turns out that one or more lexical morphemes of a short length sometimes appear without space in real text, which causes lexical ambiguity. It makes it inefficient for the system to deal with all these words in the basic morphological analysis phase.

To construct the dictionary, compound nouns were extracted from a corpus and manually refined. We first used a morphological analyzer to analyze a 30 million eojeol corpus, with only the simple morphological dictionary, and then considered the failed results as compound noun candidates. The candidates were modified and analyzed by hand after postpositions, if any, were removed from them. In addition, a collection of compound nouns of KAIST (Korea Advanced Institute of Science & Technology) was added to the dictionary in order to supplement it. The number of entries contained in the HBSD is about 100,000. Table 1 shows some examples in the HBSD. The characters such as 'n' or 'x' in the analysis information section (right-hand column) of the table are used to distinguish between nouns and suffixes.

### *3.2 Extraction of lexical information for segmentation from corpus*

As pointed out earlier, it is impossible for all compound nouns to be registered in the dictionary. This means that the HBSD cannot cover all compound nouns, even though it gives more accurate results. Therefore, we need a good model for compound noun segmentation. The segmentation algorithm uses lexical information,

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

which is presented by the Simple Noun Dictionary (SND). In this section, we describe our method to construct the SND.

In compound noun segmentation, we paid attention to lexical information which is crucial for segmenting noun compounds. Since a compound noun consists only of a sequence of nouns, i.e. *noun*$^{+}$, the transition probability of parts of speech is of no use. That is, the frequency of each noun plays an important role in compound noun segmentation. Besides, since the parameter space is huge, we cannot extract enough lexical information from about 270,000 eojeols of a POS tagged corpus[3], even though accurate lexical information can be extracted from an annotated corpus. Thus, a large corpus is required to properly estimate the frequency distribution of nouns. One problem, however, is that it is difficult to obtain accurate frequencies of words from a large corpus, because it is actually impossible to assign morphological tags to it. Therefore, we need another approach for obtaining frequencies of nouns.

Let us look at compound noun segmentation by humans before discussing knowledge acquisition. In many cases, each simple noun in compound nouns can be easily recognized by humans because it has a prominent, frequently-used figure. In fact, the prominent nouns are frequently occurring ones in documents, which we call *distinct nouns* in this paper. Compound nouns usually contain these distinct nouns, which make it easier to segment the compound nouns and identify their constituents. Further, having too many words in the dictionary is not desirable, since it gives bad results due to over-segmentation. Therefore, it is necessary to select distinct nouns, which leads us to use a part of a corpus that consists of frequently used eojeols instead of an entire corpus.

We first examined the distribution of eojeols in a corpus in order to make a subset of the corpus and collect distinct nouns and their lexical frequencies. The notable thing in our experiment was that the fixed number of eojeols always covers around 70% of the whole corpus, although the number of eojeols in the corpus increases in proportion to the size of the corpus. To begin with, figure 1 shows the relation between the number of eojeols and the size of corpora. In figure 1, the number of types of eojeols constantly increases accordingly with the size of a corpus. The main reasons are that (1) a word has numerous morphological variants, since Korean is an agglutinative language where one word has many inflectional forms, and (2) many compound words and unknown words are contained in text. However, despite the increasing number of eojeols, a very small portion of them takes up most parts of the entire corpus. For instance, figure 2 shows that just 60,000 types of eojeol take up almost 70% of three different corpora. That is, about 2.3 million for a 3 million eojeol corpus, 7.5 million for a 10 million eojeol corpus and 20.5 million for a 30 million eojeol corpus belong to the 60,000 eojeols. The lowest frequency of the 60,000 eojeols is 49 in the 30 million eojeol corpus. We decided to analyze 60,000 eojeols which were manually tractable and composed most parts of the corpus (figure 3).

We then made morphological analyses for the 60,000 eojeols using our morphological analyzer and manually corrected them. The morphological analyses often

[3] This is the size of the POS tagged corpus currently publicized by the ETRI (Electronics and Telecommunications Research Institute) project.

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

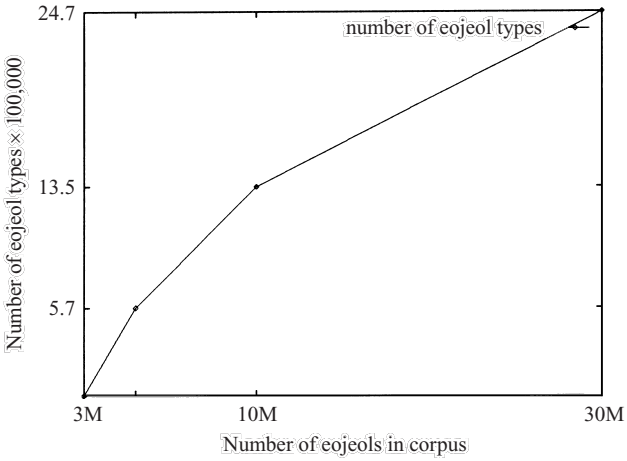



Fig. 1. Number of eojeol types by the corpus size (number of eojeols in a corpus).

contain errors and ambiguities which are due to undergeneration by restricted rules presented for efficiency, and overgeneration by more general rules presented for robustness of the system. With this manual handling, we have a very accurate morphological analyses for the frequently used eojeols.

Something interesting about this process is that many eojeols in Korean are morphologically unambiguous, so that we can easily recognize the constituent morphemes. This is mainly due to the characteristic of syllable distribution in Korean. For example, lexical morphemes and functional morphemes have different distribution, and also inflectional endings for predicates, and postpositions for nominals have quite different distribution for syllables. Such unambiguous morphological analyses make it possible to precisely count the frequencies of morphemes.

As a result of the manual process, we removed all the erroneous results, and only 15% of the 60,000 eojeols remained ambiguous at the mid-level of part of speech classification.[4] We then computed simple nouns and their frequencies with these data.

We obtained the frequency for each simple noun in the following way: if an eojeol has an unambiguous morphological analysis and contains a noun in the lexical part, we add to the lexical dictionary the noun with the frequency of the eojeol. On the other hand, if an eojeol is morphologically ambiguous, we equally distribute its frequency to all the lexical morphemes in the ambiguous eojeol, instead of simply adding the frequency. We then also take nouns, if any, and add them with the frequency to the dictionary. Furthermore, the frequency count is just added if a noun already exists in the dictionary. For instance, *gage* has two possibilities

[4] At the mid-level of part of speech classification, for example, noun, verb, ending and postposition, are represented by one tag without subdivision, e.g. N, V, E and P. To identify the sentential or clausal type (subordinate or declarative) in Korean, the ending should be subclassified for syntactic analysis in more detail, which can be done by a statistical process. This is beyond the scope of this paper.

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

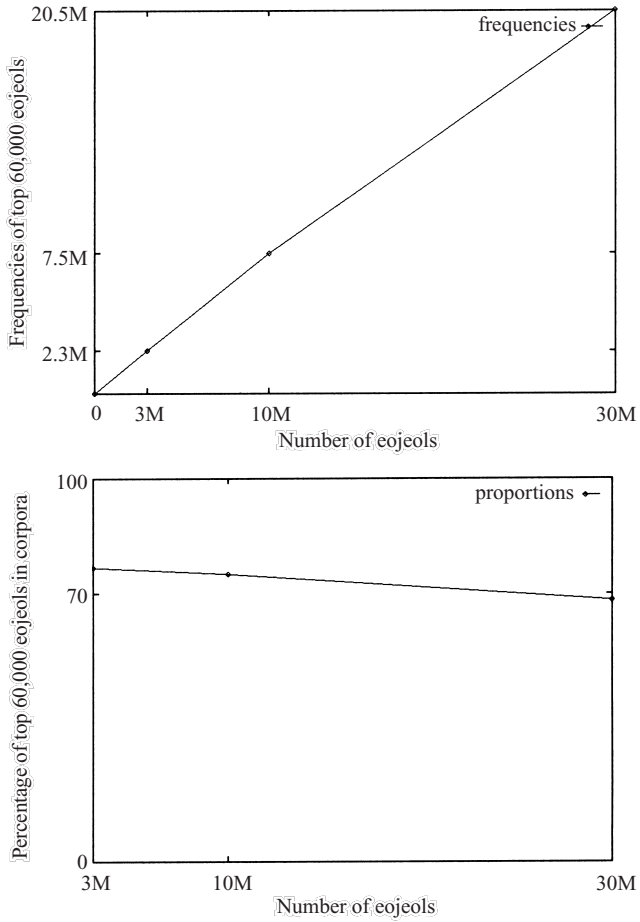



Fig. 2. Distribution of the most frequent 60,000 eojeols selected. In the top figure, the frequencies of the 60,000 eojeols in each corpus takes about 70% of those of all eojeols regardless of the corpus size. The bottom figure shows the proportion is uniform.

of analysis, i.e. '*gage*/N' and '*ga*/V+*ge*/E', and its frequency is 2263, in which the noun '*gage*' is assigned 1132 as its frequency (Table 2). The frequency of the noun '*gage*' in the eojeol '*gagega*' is assigned 165 because the morphological analysis of the eojeol is unambiguous. Moreover, the count is added to *gage* in the dictionary since it exists already. Table 2 shows examples of manually corrected morphological analyses of eojeols containing a noun '*gage*' and their frequencies. In the table, NOM, TOP and ACC represent the nominative marker, the topic marker and the accusative marker, respectively, whose parts of speech are all postposition. We refer to the nouns extracted in such a way as *a set of distinct nouns.*

In addition, we supplemented the dictionary with other necessary nouns which do not appear in the nouns obtained by the method mentioned, above but which are frequently used. First, nouns of more than three syllables are rare in real text in Korean, as shown in Lee and Ahn (1996). Their experiments proved that the syllable-

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

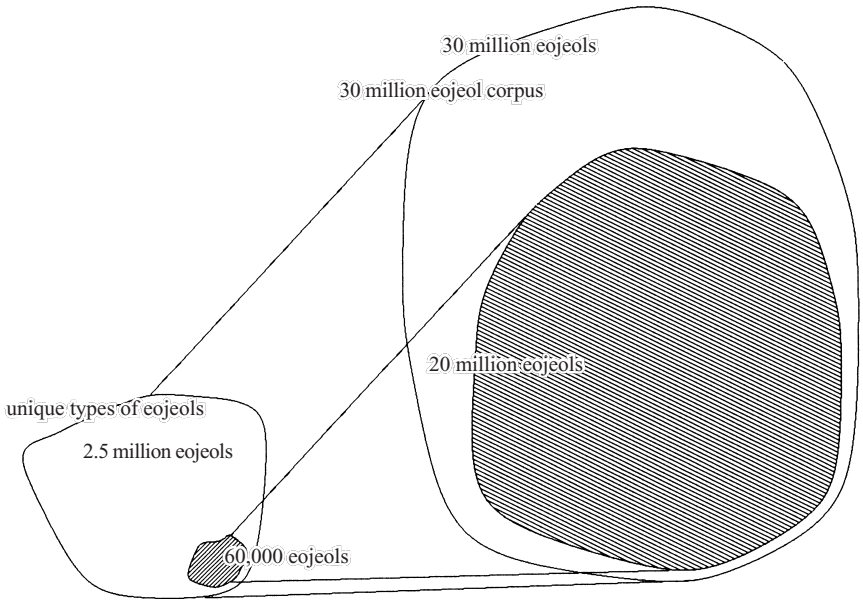



Fig. 3. Distribution of eojeols in a Korean corpus. The figure shows that a small portion of eojeols makes up most of a corpus.

Table 2. *Examples of distinct nouns extracted. Here N, V, P and E mean the tags for noun, verb, postposition and ending and ' | ' is used for representation of ambiguous analyses*

| Eojeols | Constituents | Meaning | Frequencies |
|---|---|---|---|
| *gage* | *gage*/N \| *ga*/V+*ge*/E | store \| go | 2263 |
| *gagega* | *gage*/N+*ga*/P | store/NOM | 165 |
| *gageneun* | *gage*/N+*neun*/P \| *ga*/V+*geneun*/E | store/TOP \| go | 113 |
| *gagero* | *gage*/N+*ro*/P | to the store | 166 |
| *gagereul* | *gage*/N+*reul*/P | store/ACC | 535 |
| *gagee* | *gage*/N+*e*/P | in the store | 312 |
| *gageeseo* | *gage*/N+*eseo*/P | in the store | 299 |
| *gageyi* | *gage*/N+*yi*/P | of the store | 132 |
| extracted noun | | | frequency |
| *gage* | | store | 2797 |

based bigram indexing model offers even better results than other n-gram models, such as trigram and quadragram in Korean IR, which shows that two-syllable nouns take an overwhelming majority in nouns. For this reason, we cannot find many such nouns in the set of distinct nouns. In particular, since many nouns of more than three syllables are generated by a word and derivational suffixes, and furthermore, often have syllable features, they are useful for distinguishing the boundaries of constituents in compound nouns. We selected nouns of more than three syllables from the morphological dictionary which is used for basic morphological analysis,

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

and consists of 89,000 words (noun, verb, adverb, etc.). Secondly, simple nouns were extracted from the HBSD. From the HBSD, we selected nouns which do not exist in a set of distinct nouns.

While the supplementary nouns are selected to supplement distinct nouns, the frequency of supplementary nouns does not have much influence on system performance, since our model is based on min-max composition, and most segmentation is dominated by the distinct nouns. Thus, the frequency of the supplementary nouns was assigned equally with some value *fq*. The nouns extracted in this way are referred to as *a set of supplementary nouns*.

To summarize, the SND consists of the following: (1) a set of distinct nouns which are extracted from 60,000 eojeols that frequently occur in a corpus; and (2) a set of supplementary nouns comprising nouns of more than three syllables in length, and simple nouns form the HBSD that come from real compound nouns. The number of simple nouns in the SND for compound noun segmentation is about 50,000.

## 4 Compound word segmentation algorithm

### *4.1 Basic idea*

To simply describe the basic idea of our compound noun segmentation, we first considered the case in which a compound noun is segmented into only two parts. Given a compound noun, it is segmented by the possibility that a sequence of syllables in the compound noun forms a word. The possibility that a sequence of syllables forms a word is measured by the following formula:

$$Word(s_i, \ldots, s_j) = \frac{fq(s_i, \ldots, s_j)}{fq_N} \tag{1}$$

In the formula, $fq(s_i, \ldots, s_j)$ is the frequency of the syllable sequence $s_i, \ldots, s_j$, which is obtained from the SND constructed at the stage of lexical data extraction. The denominator $fq_N$ is the total sum of frequencies of simple nouns. It is a constant for normalizing the possibility value, since the value might be used as weights for applications such as POS tagging. Equation (1) estimates how much the given sequence of syllables is likely to be a word. Naturally, if a sequence of syllables is in the set of distinct nouns, it has a large value of *Word* and is a well known word. In addition, if subsequences of syllables are included in the set of distinct nouns (i.e. frequently occurring words), it is more likely that the entire sequence of syllables is divided around the subsequences. On the other hand, if a subsequence of given syllables is not found in the SND, it is unlikely to be a word. A segmentation candidate containing such a subsequence of syllables would not be the correct result. A suitable operation for this case of estimation is the min-max composition. That is, we first take the minimum value from the function *Word* for each possibility of segmentation, and then we choose the maximum from the selected minimums. Besides, if two minimums are equal, the values of the other sequence of syllables are compared. Whole syllables are preferred if all the values are the same despite the comparison. Also, the argument taking the maximum is selected as the most likely segmentation result.

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

Consider a compound noun '*haggyosaenghwal* (school life)'. If we assume that the syllables are segmented into two, there would be four possibilities of segmentation for the example, as follows:

| | | |
|---|---|---|
| 1. | *hag* | *gyosaenghwal* |
| 2. | *haggyo* | *saenghwal* |
| 3. | *haggyosaeng* | *hwal* |
| 4. | *haggyosaenghwal* | $\phi$ |

In the first of the above example, *hag* is in the SND, but *gyosaenghwal* is not. Actually, *gyosaenghwal* is not a word. On the other hand, both *haggyo* and *saenghwal*, which are frequently occurring syllables, exist in the SND and are all words. In this case, $min(Word(hag), Word(gyosaenghwal)) < min(Word(haggyo), Word(saenghwal))$, and the more probable segmentation is $haggyo + saenghwal$.

### *4.2 Segmentation algorithm*

In this section, we generalize the word segmentation algorithm based on the concept discussed in the previous section. The basic idea is to repetitively apply the min-max operator from every two individual syllables in order to generate a longer sequence of syllables.

For instance, consider a compound word consisting of three syllables $s_1$, $s_2$ and $s_3$. The sequence of syllables $s_1s_2$ is segmented if the minimum between *Words* of two individual syllables is greater than *Word* of the combination of them, i.e. $\min(Word(s_1), Word(s_2)) > Word(s_1s_2)$. Otherwise, it is not segmented. That is, the maximum among minimums and its argument are selected. The most likely segmentation of $s_1s_2$ is then either $s_1s_2$ or $s_1 + s_2$, and similarly that of $s_2s_3$, is either $s_2s_3$ or $s_2 + s_3$. This operation is applied until it is executed for all of the syllables. Then, the most likely segmentation of $s_1s_2s_3$ can be determined by comparing the next segmentation possibilities, i.e. chosen from $S(s_1s_2) + s_3$, $s_1 + S(s_2s_3)$ and $s_1s_2s_3$. Here, $S(s_i, \ldots, s_j)$ is the most likely segmentation of $s_i, \ldots, s_j$. Figure 4 shows this process.

Moreover, the previous results such as $S(s_1s_2)$ and $S(s_2s_3)$ do not need to be re-estimated for longer syllables once they are made. For this case, we can effectively implement the segmentation algorithm by borrowing the CYK parsing method, which is dynamic programming. Using the CYK method, the execution looks like a composition rather than segmentation, since we use the bottom-up strategy. The final result is put in the top of the table after all possible segmentations of syllables are checked. When a compound noun is composed of $n$ syllables $s_1s_2, \ldots, s_n$, the composition is started from each $s_i (i = 1, \ldots, n)$. Thus, the possibility that the individual syllable forms a word is recorded in the cell of the first row, as shown in Figure 5.

In Figure 5, $C_{i,j}$ is an element of the CYK table where the segmentation result of the syllables $s_{j,\ldots,j+i-1}$ is stored. For instance, the segmentation result of $s_1, \ldots, s_i$

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

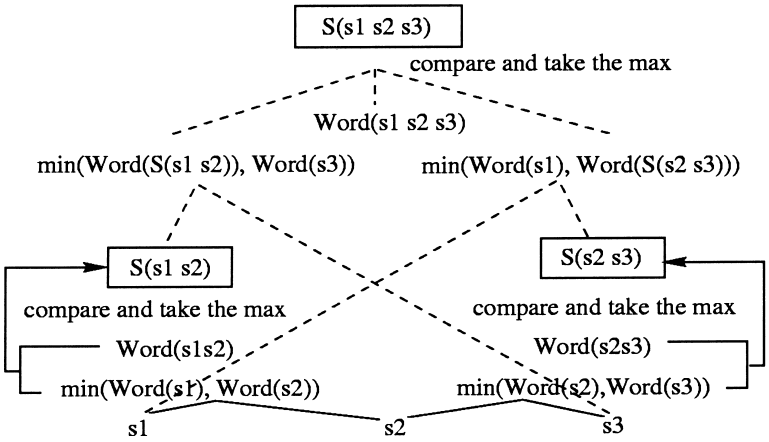



Fig. 4. Segmentation of a compound noun composed of three syllables.

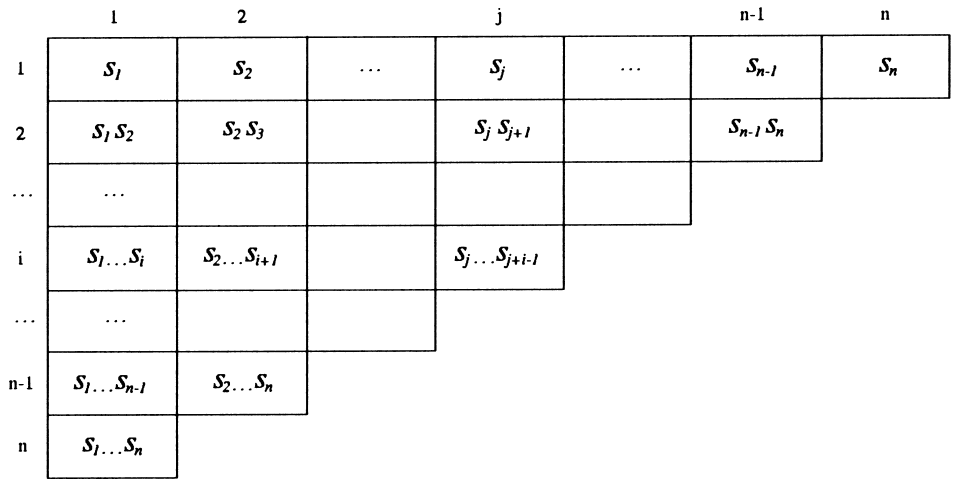



Fig. 5. Composition table.

is contained in $C_{i,1}$. In addition, the segmentation result of $s_2, \ldots, s_{i+1}$ is contained in $C_{i,2}$. Moreover, each cell $C_{i,j}$ has the most probable segmentation result for a series of syllables $s_{j,\ldots,j+i-1}$, because the procedure in CYK tabular parsing naturally follows the dynamic programming. For example, $C_{2,1}$ and $C_{2,3}$ have the most likely segmentation of $s_1 s_2$ and $s_3 s_4$, respectively. In addition, the most likely segmentation of $s_1 s_2 s_3$ is made in $C_{3,1}$ based on the previous results such as segmentations of $s_1 s_2$ and $s_2 s_3$. When the segmentation of $s_1 s_2 s_3$ is about to be checked, only $\min(value(C_{2,1}), value(C_{1,3}))$, $\min(value(C_{1,1}), value(C_{2,2}))$ and $Word(s_1 s_2 s_3)$ are compared to determine the segmentation for the syllables, because each $C_{i,j}$ already has the most likely segmentation. Here, $value(C_{i,j})$ represents the possibility value of $C_{i,j}$.

Now we describe the generalized segmentation algorithm on the basis of the above discussion. When the segmentation process is about to make the segmentation of syllables $s_i, \ldots, s_j$, the most likely segmentations for every subsequence of $s_i, \ldots, s_j$

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

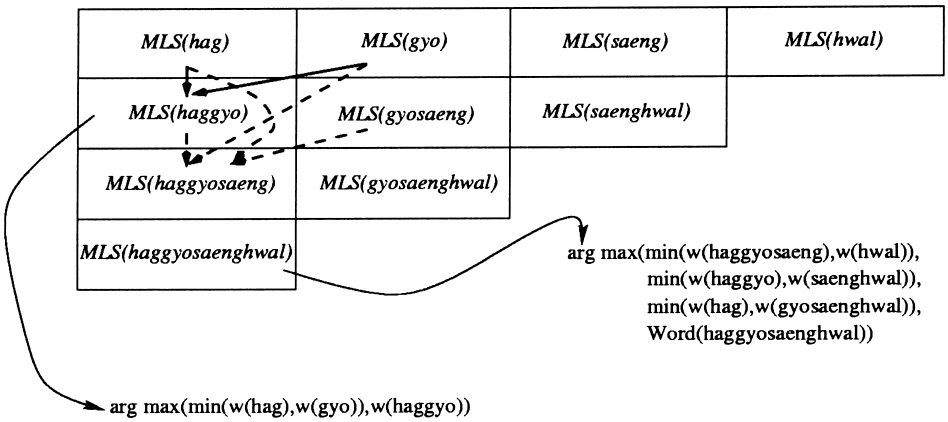



Fig. 6. State of table when analyzing '*haggyosaenghwal*'. $w(s_i,\ldots,s_j) = value(C_{i,j})$. *MLS* means the most likely segmentation.

such as $s_is_{i+1}$, $s_{i+2},\ldots,s_j$, etc. would already be stored in the table. In other words, when we consider one possibility $s_is_{i+1} + s_{i+2},\ldots,s_j$, in order to compute the most likely segmentation of $s_i,\ldots,s_j$, the segmentation of each subsequence is already known. Therefore, what we need to do is compute and compare the word generation possibilities for every two sequences of syllables $s_i,\ldots,s_k$ and $s_{k+1},\ldots,s_j (i < k < j)$ in $s_i,\ldots,s_j$.

To illustrate the execution of the algorithm, we again take the example compound noun '*haggyosaenghwal*' (school life), which is segmented into '*haggyo*' (school) and '*saenghwal*' (life) (Figure 6). When it comes up to the cell $C_{4,1}$, we have to make the most probable segmentation for '*haggyosaenghwal*', i.e. $s_1s_2s_3s_4$. There are three prefixes of sequences of syllables, i.e. $s_1$ in $C_{1,1}$, $s_1s_2$ in $C_{2,1}$ and $s_1s_2s_3$ in $C_{3,1}$, that can construct the word consisting of $s_1s_2s_3s_4$ whose segmentation would be put in $C_{4,1}$. For instance, the word $s_1s_2s_3s_4$ (*haggyosaenghwal*) is made with $s_1$ (*hag*) combined with $s_2s_3s_4$ (*gyosaenghwal*). Likewise, it might be made by $s_1s_2$ combined with $s_3s_4$ and $s_1s_2s_3$ combined with $s_4$. Since each eall has the most probable result and its value, it is simple to find the best segmentation for each syllable. In addition to the three cases, the whole sequences of syllables is compared with them to make segmentation of $s_1s_2s_3s_4$ as follows:

1. $\min(value(C_{3,1}), value(C_{1,4}))$
2. $\min(value(C_{2,1}), value(C_{2,3}))$
3. $\min(value(C_{1,1}), value(C_{3,2}))$
4. $Word(s_1s_2s_3s_4) = Word(haggyosaenghwal)$

Again, the most likely segmentation result then is put in $C_{4,1}$ with the likelihood value for its segmentation. We call it *MLS* (Most Likely Segmentation), which is found in the following way:

$$MLS(C_{4,1}) = arg\ \max(\min(value(C_{3,1}), value(C_{3,4})), \min(value(C_{2,1}), value(C_{2,3})), \min(value(C_{1,1}), value(C_{3,2})), Word(s_1s_2s_3s_4))$$

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

```
/* initialization step */
for i=1 to n do
   value(C_{1,i}) = Word(s_i);
   MLS(C_{1,i}) = s_i

for i=2 to n do
  for j=1 to n-i+1 do
   value(C_{i,j}) = max(min(value(C_{i-1,j}), value(C_{1,j+i-1})),
                        min(value(C_{i-2,j}), value(C_{2,j+i-2})),
                            ...
                        min(value(C_{1,j}), value(C_{i-1,j+1})),
                        Word(s_j ... s_{i+j}))
   MLS(C_{i,j}) = arg max(min(value(C_{i-1,j}), value(C_{1,j+i-1})),
                          min(value(C_{i-2,j}), value(C_{2,j+i-2})),
                              ...
                          min(value(C_{1,j}), value(C_{i-1,j+1})),
                          Word(s_j ... s_{i+j}))
```

Fig. 7. The segmentation algorithm.

From these four cases, the maximum value and the segmentation result are selected and recorded in $C_{4,1}$. To generalize, the algorithm is described as shown in Figure 7.

The algorithm is straightforward. Let *Word* and *MLS* be the likelihood of being a noun and the most likely segmentation for a sequence of syllables. In the initialization step, each cell in the first row of the table is assigned a *Word* value for each individual syllable $s_i (1 \leq i \leq n)$ using its frequency if it is found in SND. Then, the segmenter compares possible analyses to make a longer word, as shown in Figure 7. At each step, $value(C_{i,j})$ is estimated from the maximum value of segmentation for $s_j, \ldots, s_{j+i-1}$, which is propagated from the previous step. For every possible analysis that can be compared in a cell, the segmentation having the maximum value is put in the cell as the most likely segmentation. When all syllables from the first to -*n*th syllable are processed, $C_{n,1}$ has the segmentation result.

In addition, the algorithm compares $n$ times in order to make the most likely segmentation of $n$ syllables in each second *for* loop. As shown before, four possibilities were computed to compute $MLS(C_{4,1})$ in the case of *haggyosaenghwal*. Therefore, the overall complexity of the algorithm follows that of CYK parsing, $O(n^3)$.

### *4.3 Default analysis and tuning*

For the final result, we should take into consideration several issues which are related to the syllables that were left unsegmented. There are several reasons why the given string remains unsegmented:

1. The first is a case where the string consists of several nouns, but one of them is an unregistered word. A compound noun '*geonchugsasiheom*' is composed of '*geonchugsa*' and '*siheom*', which have the meanings of *authorized architect* and *examination*. In this case, the unknown noun is caused by the suffix

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

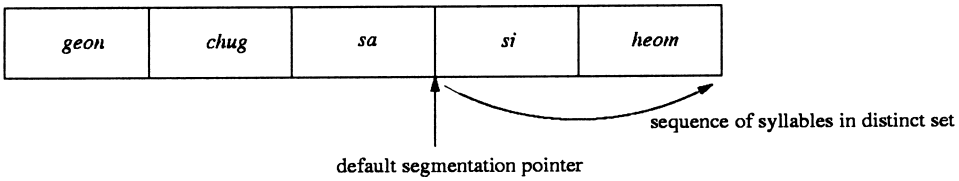



Fig. 8. Default segmentation pointer for '*geonchugsasiheom*' where '*siheom*' is a very frequently used noun.

such as '*sa*', because the suffix derives many words. However, it is known that it is very difficult to treat the kinds of suffixes since the suffix like '*sa*' is a very frequently used character in Korean, and thus prone to causing over-segmentation if included in basic morphological analysis.

2. The string might also consist of a proper noun and a noun representing a position or geometric information. For instance, a compound noun '*kim-daejungdaetongryeong*' is composed of '*kimdoejung*' and '*daetongryeong*', which mean person name and president, respectively.
3. Finally, the string might be proper noun itself. For example, '*willi'amseu*' is a transliterated word for a foreign name 'Williams', and '*honggildong*' is a person name in Korean. Generally, since it has a different sequence of syllables from general Korean words, it often remains unsegmented.

If the basic segmentation fails, the default segmentation procedures would be executed for solving the three aforementioned problems in the following way.

For the first issue, we use the *set of distinct nouns*. In the initialization step, we store, together with a frequency, the pointer to the sequence of syllables which belongs to the set of distinct nouns. Attention should be paid to infrequent sequences of syllables (those in the set of supplementary nouns) in the default segmentation, because it could be found in any proper noun like a person name, a place name, etc., or in transliterated words. It is known that the performance drops if all the nouns in the compound noun segmentation dictionary are considered for default segmentation. We save the pointer to the boundary only when a noun in a distinct set appears. For the above example, '*geonchugsasiheom*', it would be recorded that the default segmentation would be '*geonchugsa*' and '*siheom*', since '*siheom*' is in the set of distinct nouns (Figure 8).

If this procedure fails, the sequence of syllables is checked to see whether it might be a proper noun or not. Since a proper noun in Korean could have a kind of nominal suffix such as '*daetongryeong* (president)' or '*ssi* (Mr/Ms)', as mentioned above, it could identified by segmenting it. If there does not exist any nominal suffix, then the whole syllable would be regarded just as a transliterated foreign word, or a proper noun like a person or place name.

### *4.4 Further issues*

Our system has some interesting features for the application system based on morphological analysis.

First, our system can present an improved model of a spelling checker. Most current spelling checkers embedded in Korean word processors use morphological

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

analysis to judge the spelling error. If the morphological analysis of an eojeol results in a failure, namely, non-existence in the dictionary, then the eojeol is regarded as having a spelling error. The problem here comes from the dictionary, which consists of simple words. Since the number of compound nouns is not finite, they are not registered in the dictionary. That is, compound nouns which occur in an eojeol must be segmented for their correctness to be checked. However, the system performance would be dropped due to an overgeneration problem if all nouns in the dictionary are taken into consideration. Thus, most systems would generally require an error correction for compound nouns written without white space. We expect to be able to present an improved model for the spelling checker, using the idea of the distinct noun dictionary for compound noun analysis.

Secondly, the compound noun analysis would make the Korean IR (Information Retrieval) and IE (Information Extraction) systems more effective. It is well known that accurate compound noun analysis is very helpful for enhancing the IR system (Yoon, Kang and Choi 1999). Therefore, our system would be useful for this kind of information system. In addition, it could contribute to the improvement of an application system like MT (Machine Translation) since the constituents of compound nouns should be identified in order to be translated precisely. For '*haggyosaenghwal*' to be translated to *school life*, it should first be segmented '*haggyo*' and '*saenghwal*'. The compound noun segmenter is actually used for the application, as embedded in our morphological analyzer and tagger.

Another issue related to compound noun segmentation is about unknown words. There are two cases, excepting spelling errors, when a word is not found in the dictionary: (1) a compound noun; and (2) an unknown word such as proper noun or transliterated foreign word. Thus, the morphological analyzer should judge, prior to compound noun segmentation, whether a word is a compound noun or an unknown word. Currently, if a word does not exist in the dictionary, our morphological analyzer tries to judge whether it is a proper noun or a transliterated foreign word. Then, it if fails, it goes through compound noun segmentation. If this also fails in segmentation, the word is regarded as an unknown word.

## 5 Experimental results

For the test of compound noun segmentation, we first extracted compound nouns from the ETRI POS tagged corpus. From this processing, 1774 types of compound nouns were extracted, which were then used as a gold standard test set.

We evaluated our system using two methods: (1) the precision and recall rate, and (2) the segmentation accuracy per compound noun, which we refer to as SA. They are defined respectively as follows:

$$\textbf{Precision} = \frac{number\ of\ correct\ constituents\ in\ proposed\ segment\ results}{total\ number\ of\ constituents\ in\ proposed\ segment\ results} \times 100$$

$$\textbf{Recall} = \frac{number\ of\ correct\ constituents\ in\ proposed\ segment\ results}{total\ number\ of\ constituents\ in\ compound\ nouns} \times 100$$

$$\textbf{SA} = \frac{number\ of\ correctly\ segmented\ compound\ nouns}{total\ number\ of\ compound\ nouns} \times 100$$

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

Table 3. *Result 1: Precision and recall rate*

| | Precision | Recall |
|---|---|---|
| Number of correct constituents | 3553/3628 | 3553/3637 |
| Rate | 98.04 | 97.80 |

Table 4. *Result 2: Segmentation accuracy for compound noun*

| | SA | |
|---|---|---|
| | Whole System | Baseline |
| Number of correct constituents | 1726/1774 | 1673/1774 |
| Rate | 97.29 | 94.30 |

The important influence on the Korean IR system is whether words are appropriately segmented or not. The precision and recall estimates how appropriate the segmentation results are. Each is 98.04% and 97.80%, respectively, which shows that our algorithm is very effective (Table 3).

Furthermore, SA reflects how accurate the segmentation is for a compound noun. We compared two methods: (1) using only the segmentation algorithm which is the baseline of our system to estimate the accuracy of the algorithm; and (2) using both the HBSD and segmentation algorithm, which reflects system accuracy as a whole. As shown in Table 4, the baseline performance using only distinct nouns and the algorithm is about 94.3%, which is fairly good. From the results, we find that distinct nouns have a great impact on compound noun segmentation. Also, the overall segmentation accuracy for the gold standard is about 97.29%, which is a very good result for the application system. In addition, it shows that the built-in dictionary supplements the algorithm, which results in better segmentation.

Finally, we compared our system with the previous work by Yun, Cho and Rim (1997). It is impossible to directly compare our results with those, since the test set is different. It was reported that the accuracy given in the paper is about 95.6%. When comparing the performance only with the accuracy, our system outperforms theirs.

Analyzing the erroneous results, we found some interesting facts concerning the quality of the segmentation results. The segmentation errors were sometimes really bad, but sometimes tolerable. For example, '*gwanggaetowang* (King gwanggaeto)' is a proper noun of a compound noun, in which '*gwanggaeto* (person name)' and '*wang* (king)' are combined. But the compound noun is hardly ever segmented; it is used as one simple word, since '*wang*' (which means *king*) is not omitted when a king's name is referred to in Korean. Moreover, the word '*gwanggaeto*' is not a name used in present Korean, and thus an ancient king's name usually appears with the word '*wang*', representing *king*. That is, most queries may be input not as *gwanggaeto*, but as *gwanggaetowang* in IR systems, which means that it is sometimes acceptable whether the compound noun is segmented into *gwanggaeto* and *wang* or not.

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

Table 5. *Result 4: Agreement of people's segmentation and segmentation results for errors*

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| People agreed | 0 | — | 6 | 14 | 15 | 13 |
| Agreement of segmentation results | 21 | 14 | 8 | 3 | 1 | 1 |

The reason is that there are many words in Korean used in a rigid way, since highly related words are written without white space. In particular, one syllable morphemes originating from Chinese characters have a meaning by themselves, but are usually not used independently. Thus, the meaningful units that all people think of are not agreed upon, even though they know what morphemes the word is composed of. This has been discussed by (Sproat *et al.* 1994).

Therefore, we compared the system output with the results produced by humans for 48 of the erroneous results. The five people examined, who are all native speakers and involved in linguistics and computational linguistics, were asked to segment the compound nouns into meaningful units into which they thought the compounds should be segmented. According to the experiments, we have classified the erroneous results into two groups: (1) tolerable ones which are acceptable in the application system; and (2) bad ones whose segmentation results did not correspond with any of the human segmentations. Out of 48 erroneous results, 21 belong to the second case, which can be classified as bad results. Actually, the tolerable results are wrong in the gold standard, but they still have an important meaning in the application, for example in that they can appear in the query for information retrieval. From these results, we claim that the system makes 98.8% meaningful segmentation. This result clearly shows that our system would be more effective in an application system such as the IR system, which does not need to construct a dictionary.

It turned out that errors mainly occur in cases when unknown words are included in compound nouns. The system uses frequencies extracted from a corpus, but the segmentation does not follow the frequency in this case. For example, '*namyangjugun*' should be segmented into '*namyangju* (a place name in Korea)' and '*gun* (country)', but the segmentation results in *namyang* and *jugun*, since the place name *namyangju* is not registered in the segmentation dictionary but the two nouns *namyang* and *jugun* are. Besides, some parts of errors like this need semantic information, and some parts may be solved by combining this algorithm with the tagging stage.

## 6 Conclusions

In this paper, we have presented a new method for Korean compound noun segmentation. First, we proposed the lexical acquisition for compound noun analysis, which consists of the manually constructed segmentation dictionary (HBSD) and the dictionary for applying the segmentation algorithm (SND). The hand-built segmentation dictionary was made manually for compound nouns extracted from a corpus. The simple noun dictionary is based on very frequently occurring nouns,

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press

which are called distinct nouns because they are clues for identifying the constituents of compound nouns. Secondly, the compound noun was segmented based on the modification of CYK tabular parsing and min-max composition, which was proved to be a very effective method from our experiments. The bottom up approach using the min-max operation guarantees the most likely segmentation, as it is applied in the same way as dynamic programming.

It turns out that the results are very accurate, and the segmentation accuracy is 97.29%. Consequently, this methodology is promising, and the segmentation system would be helpful for application systems such as machine translation and information retrieval.

## References

Cha, J., Lee, G. and Lee, J. H. (1998) Generalized unknown morpheme guessing for hybrid POS tagging of Korean. *Proceedings of the 6th Workshop on Very Large Corpora*, pp. 85–93.

Chen, K.-J. and Liu, S.-H. (1992) Large-corpus-based methods for Chinese personal name recognition. *J. Chinese Infor. Process.* **6**(3): 7–15.

Elmi, M. A. and Evens, M. (1998) Spelling correction using context. *Proceedings of the 17th Intl. Conf. on Computational Linguistics and the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 360–364.

Fukami, S., Mizumoto, M. and Tanaka, K. (1980) Some considerations on fuzzy conditional inference. *Fuzzy Sets and Systems*, **4**: 243–273.

Hopcroft, J. E. and Ullman, J. D. (1979) *Introduction to Automata Theory, Languages, and Computation.*

Jin, W. and Chen, L. (1995) Identifying unknown words in Chinese corpora. *Proceedings of the 3rd Natural Language Processing Pacific Rim Symposium*, pp. 234–239.

Kasami, T. and Torii, K. (1965) A syntax analysis procedure for unambiguous context-free grammars. *J. Association for Computing Machinery*, **16**(3): 423–431.

Lee, J. H. and Ahn, J. S. (1996) Using n-grams for Korean text retrieval. *Proceedings 19th Annual International Association for Computing Machinery SIGIR Conference on Research and Development in Information Retrieval*, pp. 216–224.

Li, J. and Wang, K. (1995) Study and implementation of nondictionary Chinese segmentation. *Proceedings of the 3rd Natural Language Processing Pacific Rim Symposium*, pp. 266–271.

Nagao, M. and Mori, S. (1994) A new method of N-gram statistics for large number of N and automatic extraction of words and phrases from large text data of Japanese. *Proceedings of the 15th Intl. Conf. on Computational Linguistics*, pp. 611–615.

Park, B.-R., Hwang, Y.-S. and Rim, H.-C. (1997) Recognizing Korean unknown words by comparatively analyzing example words. *Proceedings of Intl. Conf. on Computer Processing of Oriental Languages*, pp. 127–132.

Sproat, R., Shih, C., Gale, W. and Chang, N. (1994) A stochastic finite-state word-segmentation algorithm for Chinese. *Proceedings 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 66–72.

Yoon, J., Kang, B.-J. and Choi, K.-S. (1999) Information retrieval based on compound noun analysis for exact term extraction. *J. Computer Processing of Oriental Language* (submitted).

Yoon, J. Lee, W.-J. and Choi, K.-S. (1999). Word Segmentation Based on Estimation of Words from Examples. Manuscript.

Younger, D. H. (1967) Recognition and parsing of context-free languages in time $n^3$. *Information and Control* **10**(2): 189–208.

Yun, B.-H. Cho, M.-J. and Rim, H.-C. (1997) Segmenting Korean compound nouns using statistical information and a preference rules. *Proceedings of Pacific Association for Computational Linguistics*, pp. 45–350.

Zadeh, L. A. (1965) Fuzzy sets. *Information Control* **8**: 338–353.

https://doi.org/10.1017/S1351324901002637 Published online by Cambridge University Press