Compound Noun Segmentation Based on Lexical Data Extracted from Corpus*

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

Abstract

Compound noun analysis is one of the crucial problems in Korean language processing because a series of nouns in Korean may appear without white space in real texts, which makes it difficult to identify the morphological constituents. This paper presents an effective method of Korean compound noun segmentation based on lexical data extracted from corpus. The segmentation is done by two steps: First, it is based on manually constructed built-in dictionary for segmentation whose data were extracted from 30 million word corpus. Second, a segmentation algorithm using statistical data is proposed, where simple nouns and their frequencies are also extracted from corpus. The analysis is executed based on CYK tabular parsing and min-max operation. By experiments, its accuracy is about 97.29%, which turns out to be very effective.

1 Introduction

Morphological analysis is crucial for processing the agglutinative language like Korean since words in such languages have lots of morphological variants. A sentence is represented by a sequence of eojeols which are the syntactic unit delimited by spacing characters in Korean. Unlike in English, an eojeol is not one word but composed of a series of words (content words and functional words). In particular, since an eojeol can often contain more than one noun, we cannot get proper interpretation of the sentence or phrase without its accurate segmentation.

The problem in compound noun segmentation is that it is not possible to register all compound nouns in the dictionary since nouns are in the open set of words as well as the number of them is very large. Thus, they must be treated as unseen words without a segmentation process. Furthermore, accurate compound noun segmentation plays an important role in the application system. Compound noun segmentation is necessarily required for improving recall and precision in Korean information retrieval, and obtaining better translation in machine translation. For example, suppose that a compound noun '*seol'agsan-gugrib-gongwon*(Seol'ag Mountain National Park)' appear in documents. A user might want to retrieve documents about '*seol'agsan*(Seol'ag Mountain)', and then it is likely that the documents with *seol'agsan-gugrib-gongwon*' are also the ones in his interest. Therefore, it should be exactly segmented before indexing in order for the documents to be retrieved with the query '*seol'agsan*'. Also, to translate '*seol'agsan-gugrib-gongwon*' to Seol'ag Mountain National Park, the constituents should be identified first through the process of segmentation.

This paper presents two methods for segmentation of compound nouns. First, we extract compound nouns from a large size of corpus, manually divide them into simple nouns and construct the hand built segmentation dictionary with them. The dictionary includes compound nouns which are frequently used and need exceptional process. The number of data are about 100,000.

Second, the segmentation algorithm is applied if the compound noun does not exist in the built-in dictionary. Basically, the segmenter is based on frequency of individual nouns extracted from corpus. However, the problem is that it is difficult to distinguish proper noun and common noun since there is no clue like capital letters in Korean. Thus, just a large amount of lexical knowledge does not make good results if it contains incorrect data and also it is not appropriate to use frequencies obtained by automatically tagging large corpus. Moreover, sufficient lexical data cannot be acquired from small amounts of tagged corpus.

In this paper, we propose a method to get simple nouns and their frequencies from frequently occurring eojeols using repetitiveness of natural language. The amount of eojeols investigated is manually tractable and frequently used nouns extracted from them are crucial for compound noun segmentation. Furthermore, we propose *min-max* composition to divide a sequence of syllables, which would be proven to be an effective method by experiments.

* This work was supported by a KOSEF's postdoctoral fellowship grant.

To briefly show the reason that we select the operation, let us consider the following example. Suppose that a compound noun be composed of four syllables '$s_1s_2s_3s_4$'. There are several possibilities of segmentation in the sequence of syllables, where we consider the following possibilities ($s_1/s_2s_3s_4$) and (s_1s_2/s_3s_4). Assume that 's_1' is a frequently appearing word in texts whereas '$s_2s_3s_4$' is a rarely occurring sequence of syllables as a word. On the other hand 's_1s_2' and 's_3s_4' occurs frequently but although they don't 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 should not be divided into frequently occurring one and rarely occurring one. In this sense, min-max is the appropriate operation for the selection. In other words, 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 tabular parsing style algorithm.

2 Lexical Data Acquisition

Since the compound noun consists of a series of nouns, the probability model using transition among parts of speech is not helpful, and rather lexical information is required for the compound noun segmentation. Our segmentation algorithm is based on a large collection of lexical information that consists of two kinds of data: One is the hand built segmentation dictionary (**HBSD**) and the other is the simple noun dictionary for segmentation (**SND**).

2.1 Hand-Built Segmentation Dictionary

The first phase of compound noun segmentation uses the built-in dictionary (HBSD). The advantage of using the built-in dictionary is that the segmentation could (1) be very accurate by hand-made data and (2) become more efficient. In Korean compound noun, one syllable noun is sometimes highly ambiguous between suffix and noun, but human can easily identify them using semantic knowledge. For example, one syllable noun '*ssi*' in Korean might be used either as a suffix or as a noun which means 'Mr/Ms' or '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 the number of compound nouns containing a meaning of 'seed' is even smaller. Besides, we can treat general spacing errors using the dictionary. By the spacing rule for Korean, there should be one content word except noun in an eojeol, but it turns out that one or more content words of short length sometimes appear without space in real texts, which causes the lexical ambiguities. It makes the system inefficient to deal with all these words on the phase of basic morphological analysis.

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(graph seed)	podo(grape)+*n*ssi(seed)
chuggutim(football team)	chuggu(football)+tim(team)

Table 1: Examples of compound noun and analysis information in built-in dictionary

To construct the dictionary, compound nouns are extracted from corpus and manually elaborated. First, the morphological analyzer analyzes 30 million eojeol corpus using only simple noun dictionary, and the failed results are candidates for compound noun. After postpositions, if any, are removed from the compound noun candidates of the failure eojeols, the candidates are modified and analyzed by hand. In addition, a collection of compound nouns of KAIST (Korea Advanced Institute of Science & Technology) is added to the dictionary in order to supplement them. The number of entries contained in the built-in dictionary is about 100,000. Table 1 shows some examples in the built-in dictionary. The italic characters such as '*n*' or '*x*' in analysis information (right column) of the table is used to make distinction between noun and suffix.

2.2 Extraction of Lexical Information for Segmentation from Corpus

As we said earlier, it is impossible for all compound nouns to be registered in the dictionary, and thus the built-in dictionary cannot cover all compound nouns even though it gives more accurate results. We need some good segmentation model for compound noun, therefore.

In compound noun segmentation, the thing that we pay attention to was that lexical information 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 no use. Namely, the frequency of each noun plays highly important role in compound noun segmentation. Besides, since the parameter space is huge, we cannot extract enough lexical information from hundreds of thousands of POS tagged corpus[1] even if accurate lexical information can be extracted from annotated corpus. Thus, a large size of corpus should be used to extract proper frequencies of nouns. However, it is difficult to look at a large size of corpus and to assign analyses to it, which makes it difficult to estimate the frequency distribution of words. Therefore, we need another approach for obtaining frequencies of nouns.

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

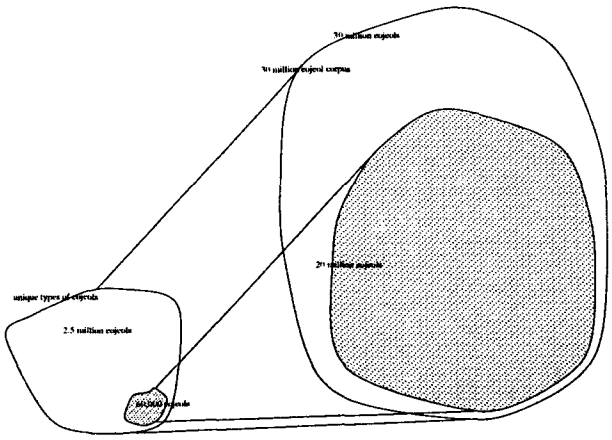

Figure 1: Distribution of eojeols in Korean corpus

It must be noted here that each noun in compound nouns could be easily segmented by human in many cases because it has a prominent figure in the sense that it is a frequently used word and so familiar with him. In other words, nouns prominent in documents can be defined as frequently occurred ones, which we call *distinct nouns*. Compound nouns contains these distinct nouns in many cases, which makes it easier to segment them and to identify their constituents. Empirically, it is well-known that too many words in the dictionary have a bad influence on morphological analysis in Korean. It is because rarely used nouns result in oversegmentation if they are included in compound noun segmentation dictionary. Therefore, it is necessary to select distinct nouns, which leads us to use a part of corpus instead of entire corpus that consists of frequently used ones in the corpus.

First, we examined distribution of eojeols in corpus in order to make the subset of corpus to extract lexical frequencies of nouns. The notable thing in our experiment is that the number of eojeols in corpus is increased in proportion to the size of corpus, but a small portion of eojeols takes most parts of the whole corpus. For instance, 70% of the corpus consists of just 60 thousand types of eojeols which take 7.5 million of frequency from 10 million eojeol corpus and 20.5 million from 30 million eojeols. The lowest frequency of the 60,000 eojeols is 49 in 30 million eojeol corpus. We decided to take 60,000 eojeols which are manually tractable and compose most parts of corpus (Figure 1).

Second, we made morphological analyses for the 60,000 eojeols by hand. Since Korean is an agglutinative language, an eojeol is represented by a sequence of content words and functional words as mentioned before. Especially, content words and functional words often have different distribution of syllables. In addition, inflectional endings for predicate and postpositions for nominals also have quite different distribution for syllables. Hence we can distinguish the constituents of eojeols in many cases. Of course, there are also many cases in which the result of morphological analysis has ambiguities. For example, an eojeol '*na-neun*' in Korean has ambiguity of '*na*/N+*neun*/P', '*na*/PN+*neun*/P' and '*nal*/V+*neun*/E'. In this example, the parts of speech N, PN, P, V and E mean noun, pronoun, postposition, verb and ending, respectively. On the other hand, many eojeols which are analyzed as having ambiguities by a morphological analyzer are actually not ambiguous. For instance, '*ga-geora*' (go/imperative) has ambiguities by most morphological analyzer among '*ga*/V+*geora*/E' and '*ga*/N+*i*/C+*geora*/E' (C is copula), but it is actually not ambiguous. Such morphological ambiguity is caused by overgeneration of the morphological analyzer since the analyzer uses less detailed rules for robustness of the system. Therefore, if we examine and correct the results scrupulously, many ambiguities can be removed through the process.

As the result of the manual process, only 15% of 60,000 eojeols remain ambiguous at the mid-level of part of speech classification[2]. Then, we extracted simple nouns and their frequencies from the data. Despite of manual correction, there must be ambiguities left for the reason mentioned above. There may be some methods to distribute frequencies in case of ambiguous words, but we simply assign the equal distribution to them. For instance, *gage* has two possibilities 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 shows examples of manually corrected morphological analyses of eojeols containing a noun '*gage*' and their frequencies. We call the nouns extracted in such a way *a set of distinct nouns.*

In addition, we supplement the dictionary with other nouns not appeared in the words obtained by the method mentioned above. First, nouns of more than three syllables are rare in real texts in Korean, as shown in Lee and Ahn (1996). Their experiments proved that syllable based bigram indexing model makes much better result than other n-gram model such as trigram and quadragram in Korean IR. It follows that two syllable nouns take an overwhelming majority in nouns. Thus, there are not many such nouns in the simple nouns extracted by the manually corrected nouns (a set of distinct nouns). In particular, since many nouns of more

[2]At the mid-level of part of speech classification, for example, endings and postpositions are represented just by one tag e.g. E and P. To identify the sentential or clausal type (subordinate or declarative) in Korean, the ending should be subclassified for syntactic analysis more detail which can be done by statistical process. It is beyond the subject of this paper.

eojeols	constituents	meaning	frequencies
gage	gage/N@ga/V+ge/E	store@go	2263
gage-ga	gage/N+ga/P	store/SUBJ	165
gage-neun	gage/N+neun/P@ga/V+geneun/E	store/TOP@go	113
gage-ro	gage/N+ro/P	to the store	166
gage-reul	gage/N+reul/P	store/OBJ	535
gage-e	gage/N+e/P	in the store	312
gage-eseo	gage/N+eseo/P	in the store	299
gage-yi	gage/N+yi/P	of the store	132
extracted noun			frequency
gage		store	2797

Table 2: Example of extraction of distinct nouns. Here N, V, P and E mean tag for noun, verb, postposition and ending and '@' is marked for representation of ambiguous analysis

than three syllables are derived by a word and suffixes and have some syllable features, they are useful for distinguishing the boundaries of constituents in compound nouns. We select nouns of more than three syllables from morphological dictionary which is used for basic morphological analysis and consists of 89,000 words (noun, verb, adverb etc). Second, simple nouns are extracted from hand-built segmentation dictionary. We selected nouns which do not exist in a set of distinct nouns.

The frequency is assigned equally with some value fq. Since the model is based on min-max composition and the nouns extracted in the first phase are most important, the value does not take an effect on the system performance.

The nouns extracted in this way are referred to as *a set of supplementary nouns*. And the SND for compound noun segmentation is composed of a set of distinct nouns and a set of supplementary nouns. The number of simple nouns for compound noun segmentation is about 50,000.

3 Compound Word Segmentation Algorithm

3.1 Basic Idea

To simply describe the basic idea of our compound noun segmentation, we first consider a compound noun to be segmented into only two nouns. Given a compound noun, it is segmented by the possibility that a sequence of syllables inside it forms a word. The possibility that a sequence of syllables forms a word is measured by the following formula.

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

In the formula, $fq(s_i, \ldots s_j)$ is the frequency of the syllable $s_i \ldots s_j$, which is obtained from SND constructed on the stages of lexical data extraction. And, fq_N is the total sum of frequencies of simple nouns. Colloquially, the equation (1) estimates how much the given sequence of syllables are likely to be word. If a sequence of syllables in the set of distinct nouns is included in a compound noun, it is more probable that it is divided around the syllables. If a compound noun consists of, for any combination of syllables, sequences of syllables in the set of supplementary nouns, the boundary of segmentation is somewhat fuzzy. Besides, if a given sequence of syllables is not found in SND, it is not probable that it is a noun.

Consider a compound noun '*hag-gyo-saeng-hwal*(school life)'. In case that segmentation of syllables is made into two, there would be four possibilities of segmentation for the example as follows:

1.	*hag*	*gyo-saeng-hwal*
2.	*hag-gyo*	*saeng-hwal*
3.	*hag-gyo-saeng*	*hwal*
4.	*hag-gyo-saeng-hwal*	ϕ

As we mentioned earlier, it is desirable that the eojeol is segmented in the position where each sequence of syllables to be divided occurs frequently enough in training data. As the length of a sequence of syllables is shorter in Korean, it occurs more frequently. That is, the shorter part usually have higher frequency than the other (longer) part when we divide syllables into two. Moreover, if the other part is the syllables that we rarely see in texts, then the part would not be a word. In the first of the above example, *hag* is a sequence of syllable appearing frequently, but *gyo-saeng-hwal* is not. Actually, *gyo-saeng-hwal* is not a word. On the other hand, both *hag-gyo* and *saeng-hwal* are frequently occurring syllables, and actually they are all words. Put another way, if it is unlikely that one sequence of syllables is a word, then it is more likely that the entire syllables are not segmented. The min-max composition is a suitable operation for this case. Therefore, 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. Also, the argument taking the maximum is selected as the most likely segmentation result.

Here, $Word(s_i \ldots s_j)$ is assigned the frequency of the syllables $s_i \ldots s_j$ from the dictionary SND. Besides, if two minimums are equal, the entire syllable such as *hag-gyo-saeng-hwal*, if compared, is preferred, the values of the other sequence of syllables are compared or the dominant pattern has the priority.

3.2 Segmentation Algorithm

In this section, we generalize the word segmentation algorithm based on data obtained by the training method described in the previous section. The basic idea is to apply min-max operation to each syllable in a compound noun by the bottom-up strategy. That is, if the minimum between *Word*s of two sequences of syllables is greater than $Word$ of the combination of them, the syllables should be segmented. For instance, let us suppose a compound noun consist of two syllable s_1 and s_2. If $\min(Word(s_1), Word(s_2)) > Word(s_1 s_2)$, then the compound noun is segmented into s_1 and s_2. It is not segmented, otherwise. That is, we take the maximum among minimums. For example, '*hag*' is a frequently occurring word, but '*gyo*' is not in Korean. In this case, we can hardly regard the sequence of syllable '*hag-gyo*' as the combination of two words '*hag*' and '*gyo*'. The algorithm can be applied recursively from individual syllable to the entire syllable of the compound noun.

The segmentation algorithm is effectively implemented by borrowing the CYK parsing method. Since we use the bottom-up strategy, the execution looks like composition rather than segmentation. After all possible segmentation of syllables being checked, the final result is put in the top of the table. When a compound noun is composed of n syllables, i.e. $s_1 s_2 \ldots s_n$, 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.

Here, $C_{i,j}$ is an element of CYK table where the segment result of the syllables $s_{j,\ldots,j+i-1}$ is stored (Figure 2). For instance, the segmentation result such that $arg\max(\min(Word(s_1), Word(s_2)), Word(s_1 s_2))$ is stored in $C_{1,2}$. What is interesting here is that the procedure follows the dynamic programming. Thus, each cell $C_{i,j}$ has the most probable segmentation result for a series of syllables $s_{j,\ldots,j+i-1}$. Namely, $C_{1,2}$ and $C_{2,3}$ have the most likely segmentation of $s_1 s_2$ and $s_2 s_3$ respectively. When the segmentation of $s_1 s_2 s_3$ is about to be checked, $\min(value(C_{2,1}), value(C_{1,3}))$, $\min(value(C_{1,1}), value(C_{2,2}))$ and $Word(s_1 s_2 s_3)$ are compared to determine the segmentation for the syllables, because all $C_{i,j}$ have the most likely segmentation. Here, $value(C_{i,j})$ represents the possibility value of $C_{i,j}$.

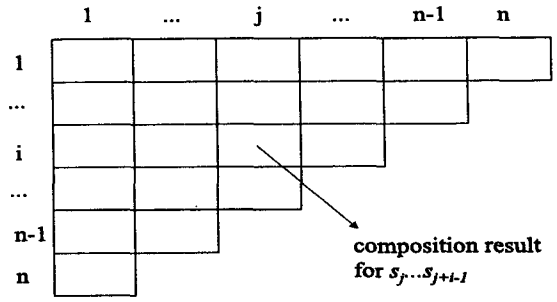

Figure 2: Composition Table

Then, we can describe the segmentation algorithm as follows:

When it is about to make the segmentation of syllables $s_i \ldots s_j$, the segmentation results of less length of syllables like $s_i \ldots s_{j-1}$, $s_{i+1} \ldots s_j$ and so forth would be already stored in the table. In order to make analysis of $s_i \ldots s_j$, we combine two shorter length of analyses and the word generation possibilities are computed and checked.

To make it easy to explain the algorithm, let us take an example compound noun '*hag-gyo-saeng-hwal*' (school life) which is segmented with '*haggyo*' (school) and '*saenghwal*' (life) (Figure 3). When it comes up to cell $C_{4,1}$, we have to make the most probable segmentation for '*hag-gyo-saeng-hwal*' i.e. $s_1 s_2 s_3 s_4$. There are three kinds of sequences of syllables, i.e. s_1 in $C_{1,1}$, $s_1 s_2$ in $C_{2,1}$ and $s_1 s_2 s_3$ in $C_{3,1}$ that can construct the word consisting of $s_1 s_2 s_3 s_4$ which would be put in $C_{4,1}$. For instance, the word $s_1 s_2 s_3 s_4$ (*hag-gyo-saeng-hwal*) is made with s_1 (*hag*) combined with $s_2 s_3 s_4$ (*gyo-saeng-hwal*). Likewise, it might be made by $s_1 s_2$ combined with $s_3 s_4$ and $s_1 s_2 s_3$ combined with s_4. Since each cell has the most probable result and its value, it is simple to find the best segmentation for each syllables. In addition, four cases, including the whole sequences of syllables, are compared to make segmentation of $s_1 s_2 s_3 s_4$ as follows:

1. $\min(value(C_{3,1}), value(C_{3,4}))$
2. $\min(value(C_{2,1}), value(C_{2,3}))$
3. $\min(value(C_{1,1}), value(C_{3,2}))$
4. $Word(s_1 s_2 s_3 s_4) = Word(\textit{hag-gyo-saeng-hwal})$

Again, the most probable segmentation result is put in $C_{4,1}$ with the likelihood value for its segmentation. We call it MLS (Most Likely Segmentation)

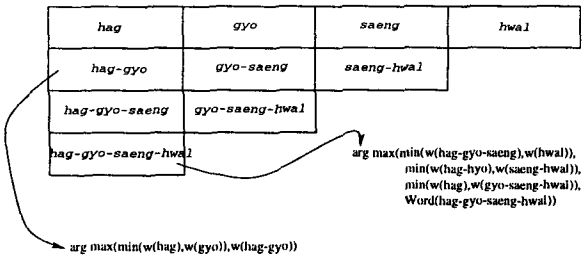

Figure 3: State of table when analyzing 'hag-gyo-saeng-hwal'. Here, $w(s_i \ldots s_j) = value(C_{i,j})$

which is found in the following way:

$$MLS(C_{4,1}) = \arg\max(\min(value(C_{3,1}), value(C_{3,4})), \min(value(C_{2,1}), value(C_{2,3})), \min(value(C_{1,1}), value(C_{3,2})), Word(s_1 s_2 s_3 s_4))$$

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

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 of the table is assigned $Word$ value for a sequence of syllables $s_j \ldots s_{j+i+1}$ using its frequency if it is found in SND. In other words, if the value of $Word$ for the sequence in each cell is greater than zero, the syllables might be as a noun a part of a compound noun and so the value is recorded as MLS. It could be substituted by more likely one in the segmentation process.

In order to make it efficient, the segmentation result is put as MLS instead of the syllables in case the sequence of syllables exists in the HBND. The minimum of each $Word$ for constituents of the result as $Word$ is recorded.

Then, the segmenter compares possible analyses to make a larger one as shown in Figure 4. Whenever $Word$ of the entire syllables is less than that of segmented one, the syllables and value are replaced with the segmented result and its value. For instance, $s_1 + s_2$ and its likelihood substitutes $C_{2,1}$ if $\min(Word(s_1), Word(s_2)) > Word(s_1 s_2)$. When the entire syllables from the first to nth syllable are processed, $C_{n,1}$ has the segmentation result.

The overall complexity of the algorithm follows that of CYK parsing, $O(n^3)$.

3.3 Default Analysis and Tuning

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

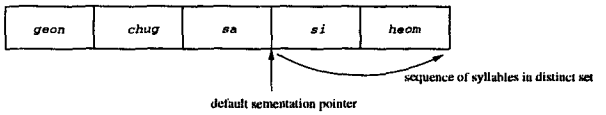

Figure 5: Default segmentation pointer for '*geon-chug-sa-si-heom*' where '*si-heom*' is a very frequently used noun.

1. The first one is a case where the string consists of several nouns but one of them is a unregistered word. A compound noun '*geon-chug-sa-si-heom*' is composed of '*geon-chug-sa*' and '*si-heom*', which have the meanings of *authorized architect* and *examination*. In this case, the unknown noun is caused by the suffix 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 make oversegmentation if included in basic morphological analysis.

2. The string might consist of a proper noun and a noun representing a position or geometric information. For instance, a compound noun '*kim-dae-jung-dae-tong-ryeong*' is composed of '*kim-dae-jung*' and '*dae-tong-ryeong*' where the former is personal name and the latter means president respectively.

3. The string might be a proper noun itself. For example, '*willi'amseu*' is a transliterated word for foreign name 'Williams' and '*hong-gil-dong*' is a personal name in Korean. Generally, since it has a different sequence of syllables from in a general Korean word, it often remains unsegmented.

If the basic segmentation is failed, three procedures would be executed for solving three problems above. For the first issue, we use the *set of distinct nouns*. That is, the offset pointer is stored in the initialization step as well as frequency of each noun in compound noun is recorded in the table. Attention should be paid to non-frequent sequence of syllables (ones in the set of supplementary nouns) in the default segmentation because it could be found in any proper noun such as personal names, place names, etc or transliterated words. It is known that the performance drops if all nouns in the compound noun segmentation dictionary are considered for default segmentation. We save the pointer to the boundary only when a noun in distinct set appears. For the above example '*geon-chug-sa-si-heom*', the default segmentation would be '*geon-chug-sa*' and '*si-heom*' since '*si-heom*' is in the set of distinct nouns and the pointer is set before '*si-heom*' (Figure 5).

/* initialization step */
for i=1 to n do
 for j=1 to n-i+1 do
  $value(C_{i,j}) = Word(s_j \ldots s_{j+i-1})$;
  $MLS(C_{i,j}) = s_j \ldots s_{j+i-1}$; if $value(C_{i,j}) > 0$
  ϕ; otherwise

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

Figure 4: The segmentation algorithm

If this procedure is failed, the sequence of syllables is checked whether it might be proper noun or not. Since proper noun in Korean could have a kind of nominal suffix such as '*daetongryeong*(president)' or '*ssi*(Mr/Ms)' as mentioned above, we can identify it by detaching the nominal suffixes. If there does not exist any nominal suffix, then the entire syllables would be regarded just as the transliterated foreign word or a proper noun like personal or place name.

4 Experimental Results

For the test of compound noun segmentation, we first extracted compound noun from ETRI POS tagged corpus[3]. By the processing, 1774 types of compound nouns were extracted, which was used as a gold standard test set.

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

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

$$\textbf{Recall} = \frac{number\ of\ correct\ constituents\ in\ proposed\ segment\ results}{total\ number\ of\ constituents\ in\ compoundnouns}$$

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

[3]The corpus was constructed by the ETRI (Electronics and Telecommunications Research Institute) project for standardization of natural language processing technology and the corpus presented consists of about 270,000 eojeols at present.

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

SA reflects how accurate the segmentation is for a compound noun at all. We compared two methods: (1) using only the segmentation algorithm with default analysis which is a baseline of our system and so is needed to estimate the accuracy of the algorithm. (2) using both the built-in dictionary and the 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% and fairly good. From the results, we can find that the distinct nouns has 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.

Lastly, we compare our system with the previous work by (Yun *et al.* , 1997). It is impossible that we directly compare our result with theirs, 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 in terms of the accuracy, our system outperforms theirs.

Embeded in the morphological analyzer, the compound noun segmentater is currently being used for some projects on MT and IE which are worked in several institutes and it turns out that the system is very effective.

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

Table 3: Result 1: Precision and recall rate

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

Table 4: Result 2: Segmentation accuracy for Compound Noun

5 Conclusions

In this paper, we presented the 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 corpus. The simple noun dictionary is based on very frequently occurring nouns which are called distinct nouns because they are clues for identifying constituents of compound nouns. Second, the compound noun was segmented based on the modification of CYK tabular parsing and min-max composition, which was proven to be the very effective method by experiments. The bottom up approach using min-max operation guarantees the most likely segmentation, being applied in the same way as dynamic programming.

With our new method, the result for segmentation is as accurate as 97.29%. Especially, the algorithm made results good enough and the built-in dictionary supplemented the algorithm. Consequently, the methodology is promising and the segmentation system would be helpful for the application system such as machine translation and information retrieval.

6 Acknowledgement

We thank Prof. Mansuk Song at Yonsei Univ. and Prof. Key-Sun Choi at KAIST to provide data for experiments.

References

Cha, J., Lee, G. and Lee, J. 1998. Generalized Unknown Morpheme Guessing for Hybrid POS Tagging of Korean. In *Proceedings of the 6th Workshop on Very Large Corpora.*

Choi, K. S., Han, Y. S., Han, Y. G., and Kwon, O. W. 1994. KAIST Tree Bank Project for Korean: Present and Future Development. In *Proceedings of the International Workshop on Sharable Natural Language Resources.*

Elmi, M. A. and Evens, M. 1998. Spelling Correction Using Context. In *Proceedings of COLING/ACL 98*

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 In *Proceedings of NLPRS 95*

Lee, J. H. and Ahn, J. S. 1996. Using n-grams for Korean Text Retrieval. In *Proceedings of 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*

Li, J. and Wang, K. 1995. Study and Implementation of Nondictionary Chinese Segmentation. In *Proceedings of NLPRS 95*

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. In *Proceedings of COLING 94*

Park, B, R., Hwang, Y. S. and Rim, H. C. 1997. Recognizing Korean Unknown Words by Comparatively Analyzing Example Words. In *Proceedings of ICCPOL 97*

Sproat, R. W., Shih, W., Gale, W. and Chang, N. 1994. A Stochastic Finite-State Word-segmentation Algorithm for Chinese. In *Proceedings of the 32nd Annual Meeting of ACL*

Yoon, J., Kang, B. and Choi, K. S. 1999. Information Retrieval Based on Compound Noun Analysis for Exact Term Extraction. Submitted in Journal of Computer Processing of Orientla Language.

Yoon, J., Lee, W. and Choi, K. S. 1999. Word Segmentation Based on Estimation of Words from Examples. Technical Report.

Yun, B. H., Cho, M. C. and Rim, H. C. 1997. Segmenting Korean Compound Nouns Using Statistical Information and a Preference Rules. In *Proceedings of PACLING.*