Information Retrieval, 4, 115–132, 2001

A Corpus-Based Learning Method of Compound Noun Indexing Rules for Korean

JEE-HYUB KIM kjh726@postech.ac.kr
Biological Research Information Center (BRIC), Pohang, South Korea

BYUNG-KWAN KWAK nerguri@postech.ac.kr
SEUNGWOO LEE pinesnow@postech.ac.kr
GEUNBAE LEE gblee@postech.ac.kr
JONG-HYEOK LEE jhlee@postech.ac.kr
Electrical and Computer Engineering Division, Pohang University of Science & Technology (POSTECH), Pohang, South Korea

Received December 27, 2000; Revised February 2, 2001; Accepted February 16, 2001

Abstract. In Korean information retrieval, compound nouns play an important role in improving precision in search experiments. There are two major approaches to compound noun indexing in Korean: statistical and linguistic. Each method, however, has its own shortcomings, such as limitations when indexing diverse types of compound nouns, over-generation of compound nouns, and data sparseness in training. In this paper, we propose a corpus-based learning method, which can index diverse types of compound nouns using rules automatically extracted from a large corpus. The automatic learning method is more portable and requires less human effort, although it exhibits a performance level similar to the manual-linguistic approach. We also present a new filtering method to solve the problems of compound noun over-generation and data sparseness.

Keywords: corpus-based learning, compound noun indexing, filtering, information retrieval, search performance evaluation

1. Introduction

In Korean, nouns play a major role in communicating the content of documents. They are used as index terms. Because compound nouns are more specific and expressive than simple nouns, they are more valuable. For our purposes, such index terms can increase precision in search experiments. There are many definitions for the compound noun: it may occur as a simple continuous noun sequence, or as a continuous noun sequence whose original meaning has been changed into a new one. These various definitions cause ambiguities concerning whether a given continuous noun sequence is a compound noun or not. Therefore, we require a clear definition of compound nouns in information retrieval. In this paper, we define a compound noun as "any continuous noun sequence that appears frequently in documents."[1]

In Korean documents, compound nouns are represented in various forms (shown in Table 1) from various sources (shown in Table 2). This presents a difficulty when indexing all types of compound nouns. Much work has been done on compound noun indexing thus far, but previous studies show limitations when covering all types of compound nouns,

Table 1. Various types of Korean compound nouns with regard to "jeong-bo geom-saeg (information retrieval)".

jeong-bo-geom-saeg (information-retrieval)
jeong-bo-eui geom-saeg (retrieval of information)
jeong-bo geom-saeg (information retrieval)
jeong-bo-leul hyo-yul-jeog-eu-lo geom-saeg-ha-neun (retrieving information efficiently)
jeong-bo-leul geom-saeg-ha-neun (retrieving information)
jeong-bo-geom-saeg si-seu-tem (information-retrieval system)
mun-heon jeong-bo geom-saeg si-seu-tem document information retrieval system)

Table 2. Various sources of Korean compound noun formation.

Compound noun Origin	Examples
Foreign word	*in-teo-ne-tue* tam-saek (internet searching),
	li-col yo-cheong (recall request), ...
Jargon	*deo-saeng* jak-eop (dessin work), ...
Acronym	*jeon-geoyng-leoyn* hoi-gwan (The Federation of Korean Industries Building), ...
Proper noun	*po-hang-gong-dae* ki-suk-sa (the dormitory of POSTECH), ...
Newly-coined word	*no-lae-bang* ju-in (the owner of Karaoke room), ...
Noun combination	*no-lae-bang mun-hwa* (the culture of Karaoke room),
	gyeong-jae beul-leog (the economic block), ...
Derivational noun	*go-hwa-jil* bi-di-o (video with the high quality of a picture), ...

requiring the application of much linguistic knowledge to accomplish the goal. In this paper, we propose a corpus-based learning method for compound noun indexing, which can extract the rules automatically, and with little linguistic knowledge.

Information retrieval systems can be evaluated for their effectiveness in recall and precision, and can also be evaluated for their efficiency in search speed and in storage space for index terms. Until now, many search experiments have been evaluated in terms of effectiveness. However, as the number of documents grows, so does the importance of efficiency. In this paper, we also deal with the efficiency issue in compound noun indexing. To increase efficiency, we focus on reducing the number of indexed spurious compound nouns. We perform experiments on several filtering methods to locate an algorithm that can most efficiently reduce spurious compound nouns.

The remainder of this paper is organized as follows. Section 2 describes previous compound noun indexing methods for Korean and compound noun filtering methods. We show the overall compound noun indexing system architecture in Section 3, and explain each

module of the system in Section 4 and 5 in detail. We evaluate our method by using standard Korean test collections in Section 6. Finally, concluding remarks are given in Section 7.

2. Previous research

2.1. Compound noun indexing

Two standard methods used in compound noun indexing are the statistical and the linguistic. With the statistical method, Fagan (1989) indexed phrases using six different parameters, including information on co-occurrence of phrase elements, the relative location of phrase elements, etc., and achieved a reasonable performance when measured for effectiveness. However, he did not achieve consistent substantial improvements in five experimental document collections. Zhai (1997) used a noun phrase parser for his linguistic method and performed extensive experiments for phrase indexing. Experimental results showed that linguistic phrase indexing consistently and significantly improved retrieval performance. Strzalkowski et al. (1996) and Evans and Zhai (1996) indexed subcompounds from complex noun phrases using noun-phrase analysis. These methods must locate head-modifier relations in noun phrases, so require difficult syntactic parsing in Korean.

For linguistic methods, Kim (1994) used five manually chosen compound noun indexing rule patterns based on linguistic knowledge. However, this method cannot index diverse types of compound nouns. Won et al. (2000) used a full parser to increase precision in search experiments. However, this linguistic method cannot be applied to unrestricted texts in a robust manner.

In summary, previous methods, whether statistical or linguistic, have their own shortcomings. Statistical methods require significant amounts of co-occurrence information to ensure a reasonable performance, and cannot index diverse types of compound nouns. Linguistic methods require compound noun indexing rules to be manually inserted. Further, such manual insertion cannot prevent the generation of meaningless compound nouns as the result. This decreases the performance of information retrieval systems. In addition, linguistic methods cannot cover various types of compound nouns because of the limitations of current linguistic knowledge.

In this paper, we present a hybrid method that uses linguistic rules, but these rules are automatically acquired from a large corpus through statistical learning. Our method generates more diverse compound noun indexing rule patterns than previous standard methods (Kim 1994, Lee et al. 1997) because previous methods used only the most general rule patterns (shown in Table 3) and were based solely on the current state of linguistic knowledge.

Kim et al. (1998) added ten additional hand-written rules to cover a wider range of compound nouns, but their method, has low accuracy, and is also incapable of indexing diverse types of compound nouns. It follows that simply adding more linguistic rules will not work. Therefore, a new statistical approach is required to extract compound noun indexing rules to cover all kinds of compound noun types in unrestricted texts. For example, Table 4 shows some examples of meaningful compound nouns that cannot be indexed by the previous methods.

Table 3. Typical hand-written compound noun indexing rule patterns for Korean.

Noun without case makers/Noun
Noun with a genitive case maker/Noun
Noun with a nominal case maker or an accusative case maker /Verbal common noun or adjectival common noun
Noun with an adnominal ending/Noun
Noun within predicate particle phrase/Noun
(The two nouns before and after a slash (/) in the pattern can form a single compound noun.)

Table 4. New forms of compound nouns which need to be indexed.

chim-sig-eu-lo i-lu-eo-jin pyeong-ya (plane made by erosion) → chim-sig/pyeong-ya (erosion plane)
sa-gwa-ga yeol-lin na-mu (tree bearing apple) → sa-gwa/na-mu(apple tree)
yu-li-lo man-deun keob (cup made of glass) → yu-li/keob (glass cup)

2.2. *Compound noun filtering*

Both statistical and linguistic compound noun indexing methods tend, in actual application, to generate spurious compound nouns. Since an information retrieval system can be evaluated both by effectiveness and by efficiency (van Rijsbergen 1979), spurious compound nouns should be efficiently filtered. Kando et al. (1998) insisted that, for Japanese, a smaller number of index terms should result in a superior performance of the information retrieval system. For Korean, Won et al. (2000) showed that segmentation of compound nouns is more efficient than compound noun synthesis in search performance.

Much work has been done on compound noun filtering methods; Kim (1994) used mutual information only, and Yun et al. (1997) used mutual information and relative frequency of POS (Part-Of-Speech) pairs together. Lee et al. (1997) used manually-constructed stop-word dictionaries. Most previous methods for compound noun filtering utilize only one consistent method for generated compound nouns, irrespective of the different origin of compound noun indexing rules. Such methods cause many problems due to data sparseness in both dictionaries and training data which are constructed manually. Our approach solves the data sparseness problem by applying co-occurrence information to automatically extracted compound noun elements, along with a statistical precision measure which best fits each rule.

3. Overall system architecture

The compound noun indexing system proposed in this paper consists of two major modules: one for automatically extracting compound noun indexing rules, and the other for indexing documents, filtering the automatically generated compound nouns, and weighting the indexed compound nouns.

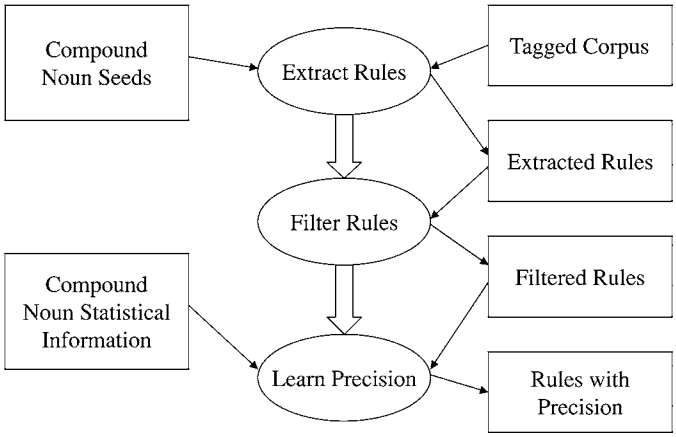

Figure 1. Compound noun indexing-rule extraction module (control flow ⇒, data flow →).

The compound noun indexing-rule extraction module (figure 1) learns indexing rules automatically from a large POS tagged corpus, with frequently used compound nouns as initial seeds. Extracted rules are filtered to select the proper rules, and the precision of the selected rules is learned by expanding the statistical information of the compound nouns. The results of the learning process are rules with learned precision, which will be applied to the subsequent compound noun indexing process.

The module for indexing, filtering, and weighing compound nouns (figure 2) indexes documents using previously extracted compound noun indexing rules. The first step is to segment words into morphemes and tag parts-of-speech for the documents. To accomplish

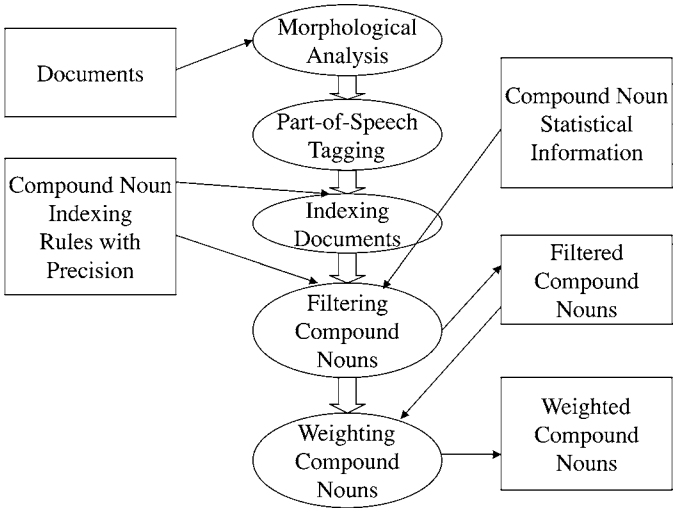

Figure 2. Compound noun indexing, filtering, and weighting module (control flow ⇒, data flow →).

this, we employ a natural language processing engine, named SKOPE (Standard KOrean Processing Engine) (Cha et al. 1998). The second step is to index the tagged documents using automatically extracted rules. The third step is to filter the indexed compound nouns to overcome a compound noun over-generation problem. Finally, we weight the indexed compound nouns. The weighted compound nouns will be used as final index terms.

4. Automatic extraction of compound noun indexing rules

There are three major steps in automatically extracting compound noun indexing rules. The first step is to collect compound noun statistical information, and the second step is to extract the rules from a large tagged corpus, using the statistical information we collected. The final step is to establish the precision of each rule.

4.1. Collecting compound noun statistics

We first collect initial compound noun seeds from various types of well-balanced documents, such as the ETRI Kemong encyclopaedia,[2] as well as a number of dictionaries on the Internet. In Korean, many compound nouns consist of two or three single nouns. We collected 10,368 seeds, as shown in Table 5. The small number of seeds are bootstrapped to extract the compound noun indexing rules for various corpora.

In order to gain more practical statistics on the compound nouns, we use the concept of "complete compound nouns." A complete compound noun is a continuous noun sequence, composed of at least two nouns, on the condition that both the preceding and the following POS of the sequence are not nouns (Yoon et al. 1998). These complete compound nouns are candidates for inclusion into final compound noun index terms. The complete compound noun statistics will be used to decide whether or not a given continuous noun sequence is a compound noun, which delivers precision information to the extracted compound noun indexing rules. To collect statistics, we constructed a 1,000,000 eojeol[3] tagged corpus for learning and a 100,000 eojeol tagged corpus for a compound noun indexing experiment from a large document set (Korean Information Base). We collected complete compound nouns composed of 2–3 nouns from the tagged training corpus (Table 6).

Table 5. Collected compound noun seeds.

Number of component elements	2	3	Total
ETRI Kemong encyclomedia	5,100	2,088	7,188
Internet dictionaries	2,071	1,109	3,180

Table 6. Statistics for complete compound nouns.

Number of component elements	1	2	3
Vocabulary	264,359	200,455	63,790

4.2. Extracting indexing rules

We define a template to extract the compound noun indexing rules from a POS tagged corpus as follows:

front-condition-tag | sub-string-tags (tag 1 tag 2 . . . tag n-1 tag n)
| rear-condition-tag | synthesis locations (x y) → lexicon x / lexicon y
(for 3-noun compounds, synthesis locations (x y z) → lexicon x / lexicon y / lexicon z)

The template means that if a front-condition-tag, a rear-condition-tag, and sub-string-tags are coincident with input sentence tags, the lexical item in the synthesis position of the sentence can be indexed as a compound noun as "x / y (for 3-noun compounds, x / y / z)". The tags used in the template are POS (Part-Of-Speech) tags. We use the POSTAG set (Appendix A). The POS tags appropriate for synthesis positions are the common noun (MC), the proper noun (MP), and the numeral (S) (i.e. these tags can be used as components of compound nouns). If the tags in sub-string-tags are action verbs (D), state verbs (H), existential verbs (E), assignment verbs (I), ending (e), or case markers (j), we add this lexical information to the rules to make them more specific.

The following is an algorithm to extract compound noun indexing rules from a large tagged corpus, using the two-noun compound seeds and the template defined above. The scope of rule extraction is limited to the end of a sentence, or, if there is a conjunctive ending (eCC) in the sentence, limited only to the conjunctive ending of the sentence. A rule extraction example is shown in figure 3.

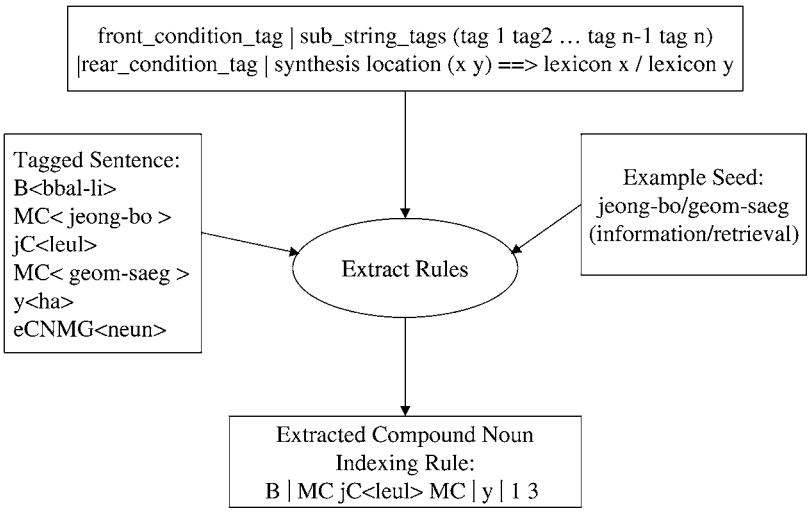

Figure 3. Rule extraction process example.

Algorithm 1: Extracting compound noun indexing rules (for 2-noun compounds)

```
Read Template
Read Seed (Consist of Constituent 1 / Constituent 2)
Tokenize Seed into Constituents
Put Constituent 1 into Key1 and Constituent 2 into Key2
While (Not(End of Documents))
 {
   Read Initial Tag of Sentence
   While (Not(End of Sentence or eCC))
   {
      Read Next Tag of Sentence
      If (Read Tag == Key1)
      {
         While (Not(End of Sentence or eCC))
         {
            Read Next Tag of Sentence
            If (Current Tag == Key2)
               Write Rule according to the Template
         }
      }
   }
 }
```

The next step is to refine the extracted rules to select the proper ones. We used a rule filtering algorithm(Algorithm 2) using frequency, together with the heuristic that rules with negative lexical items (shown in Table 7) will make spurious compound nouns.

Algorithm 2: Filtering extracted rules using frequency and heuristics

1. For each compound noun seed, select the rules whose frequency is greater than 2.
2. Among rules selected by step 1, select only rules that are extracted by at least 2 seeds.
3. Discard rules which contain negative lexical items.

We automatically extracted and filtered out 2,036 rules from the large tagged corpus (Korean Information Base, 1,000,000 eojeol) using Algorithm 2 (see above). Among the

Table 7. Negative lexical item examples.

Negative items (tags)	Example phrases
je-oe (MC) (exclude)	no-jo-leul je-oe-han hoe-eui (meeting excluding union)
eobs (E) (not exist)	sa-gwa-ga eobs-neun na-mu (tree without apple)
mos-ha (D) (can not)	dog-lib-eul mos-han gug-ga (country that can not be liberated)

Table 8. Distribution of extracted rules by number of elements in sub-string-tags.

Number	Distribution	Example
2 tags	79.6%	MC MC
3 tags	12.6%	MC jO⟨eui⟩ MC
4 tags	4.7%	MC y eCNMG MC
5 tags	1.5%	MC MC jO⟨e⟩ DI⟨sog-ha-neun⟩ MC
Over 6 tags	1.6%	MC jO⟨leul⟩ MC y⟨ha⟩ eCNMG ⟨neun⟩ MC

Table 9. Comparison between the automatically extracted rules and manual rules.

Method	Number of general rule patterns	Number of lexical terms used in rule patterns
Manual linguistic method	5	16
Our method	23	78

Table 10. Examples of newly added rule patterns.

Rule	Example
Noun+bound noun/Noun	jeon-jaeng jung go-a → jeon-jaeng/go-a
Noun+suffix/Noun	dong-seo-gan naeng-jeon → dong-seo/naeng-jeon
Noun+suffix+assignment verb +adnominal ending/Noun	u-ho-jeog-in gug-ga → u-ho/gug-ga

filtered rules, there are 19 rules with negative lexical items. Finally, we filtered out 2,017 rules. Table 8 shows a distribution of the final rules according to the number of elements in their sub-string-tags. The table shows that 98.4% of the rules consist of two to five elements in the sub-string-tags, so most of the compound nouns can be covered by using only up to 5 tag rules.

The automatically extracted rules contain more rule patterns and lexical items than man-made rules, so are able to cover more diverse types of compound nouns (Table 9). When we cross-checked the overlap between the two rule collections, we discovered that the manual linguistic rules are a subset of our automatically generated statistical rules. Table 10 shows some of the example rules newly generated from our extraction algorithm, which were previously missing from the manual rule patterns.

4.3. Learning the precision of extracted rules

The precision of a rule can be defined by counting the number of indexed compound noun candidates generated by the rule which are actual compound nouns:

$$Prec(rule) = \frac{N_{actual}}{N_{candidate}}$$

where $Prec(rule)$ is the precision of a rule, N_{actual} is the number of actual compound nouns, and $N_{candidate}$ is the number of compound noun candidates generated by the automatic indexing rules.

To calculate with precision, we require a defining measurement for compound noun identification. Su et al. (1994) showed that the average mutual information value of a compound noun tends to be higher than that of a non-compound noun. This led us to use mutual information as the criterion for identifying compound nouns. If the mutual information value of the compound noun candidate is higher than the average mutual information value of the compound noun seeds, we conclude that it is a compound noun. For mutual information (MI), we use two different equations: one for two-element compound nouns (Church and Hanks 1990) and the other for three-element compound nouns (Su et al. 1994). The equations are as follows:

$$I(x; y) \equiv \log_2 \frac{P(x, y)}{P(x) \times P(y)}$$

where x and y are two words in the corpus, and $I(x; y)$ represents the mutual information of these two words (in this order),

$$I(x; y; z) \equiv \log_2 \frac{P_D(x, y, z)}{P_I(x, y, z)}$$

where $P_D(x, y, z)$ is the probability for x, y and z to occur jointly (dependently), and $P_I(x, y, z)$ is the probability for x, y and z to occur by chance (independently), i.e., $P_I(x, y, z) \equiv P(x) \times P(y) \times P(z) + P(x) \times P(y, z) + P(x, y) \times P(z)$. Table 11 shows the average MI value of the two- and three-element compound noun seeds.

The MI was calculated from the statistics of the complete compound nouns collected from the tagged training corpus (see Section 4.1). However, complete compound nouns are continuous noun sequences, which cause a data sparseness problem. Therefore, we need to expand the statistics. Figure 4 shows the architecture of the precision learning module which expands the statistics of the complete compound nouns, along with an algorithmic explanation (Algorithm 3) of the process. Table 12 shows the improvement in the average precision during the repetitive execution of this learning process.

Table 11. Average value of the mutual information (MI) of compound noun seeds.

Number of elements	2	3
Average MI	3.56	3.62

Table 12. Improvement in the average precision of rules.

Learning cycles	1	2	3	4	5	6
Average precision of rules	0.19	0.23	0.39	0.44	0.45	0.45

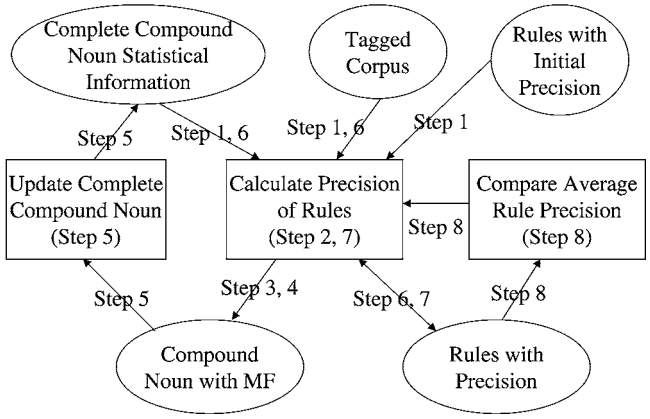

Figure 4. Learning the precision of the compound noun indexing rules (The steps are shown in Algorithm 3).

Algorithm 3:

1. Calculate all rules' initial precision using initial complete compound noun statistical information.
2. Calculate the average precision of the rules.
3. Multiply a rule's precision by the frequency of the compound noun made by the rule. We call this value the modified frequency (MF).
4. Collect the same compound nouns, and sum all the modified frequencies for each compound noun.
5. If the summed modified frequency is greater than a threshold, add this compound noun to the complete compound noun statistical information.
6. Re-calculate the precision of all rules, using the changed complete compound noun statistical information.
7. Calculate the average precision of the rules.
8. If the average precision of the rules is equal to the previous average precision, stop. Otherwise, go to step 2.

5. Compound noun indexing, filtering, and weighting

In this section, we explain how to use automatically extracted rules to index compound nouns, and describe how to filter and weight the indexed compound nouns.

5.1. Compound noun indexing

To index compound nouns from documents, we use a natural language processing engine, named SKOPE (Standard KOrean Processing Engine) (Cha et al. 1998), which processes documents by analyzing words into morphemes and tagging part-of-speeches. The tagging

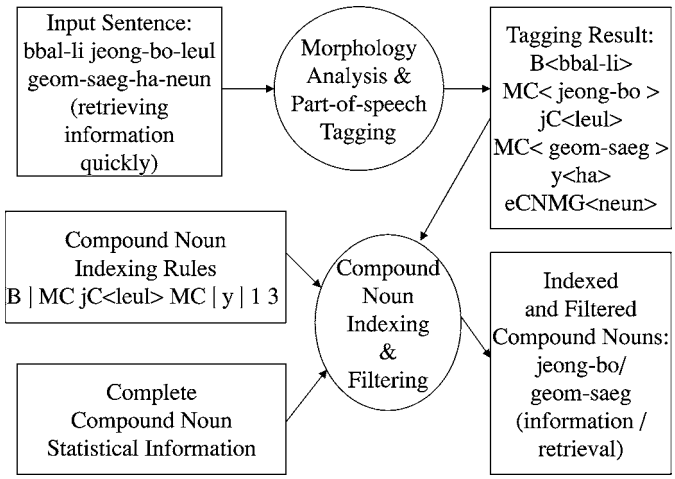

Figure 5. Compound noun indexing process.

results are compared with the automatically learned compound noun indexing rules and, if they are coincident with each other, index them as compound nouns. Figure 5 shows the process of compound noun indexing, with an example.

5.2. Compound noun filtering

Among the indexed compound nouns mentioned above, meaningless compound nouns can occur, which increase the number of index terms and the search speed. To solve this compound noun over-generation problem, we experiment with seven different filtering methods (shown in Table 13) by analyzing their relative effectiveness and efficiency, as shown in Table 19. These methods can be divided into three categories: the first, using MI, the second, using the frequency of the compound nouns (FC), and the last, using the frequency of the compound noun elements (FE). MI is a measure of word association,

Table 13. Seven different filtering methods.

(MI) A. Mutual information of compound noun elements (threshold: 0)
(MI) B. Mutual information of compound noun elements (threshold: average of MI of compound noun seeds)
(FC) C. Frequency of compound nouns in the training corpus (threshold: 4)
(FC) D. Frequency of compound nouns in the test corpus (threshold: 2)
(FE) E. Frequency of compound noun heads in the training corpus (threshold: 5)
(FE) F. Frequency of compound noun modifiers in the training corpus (threshold: 5)
G. No filtering

Table 14. Three weighting methods for compound nouns.

A. Statistical method (atn.ntc)
B. Average of the components of a compound noun
C. Sum of the components of a compound noun

which is used under the assumption that a highly associated word n-gram is more likely to be a compound noun. FC is used with the assumption that a frequently encountered word n-gram is more likely to be a compound than a rarely encountered n-gram. FE is used with the assumption that a word n-gram with a frequently-encountered specific element is more likely to be a compound. In the method of C, D, E, and F, each threshold was decided by calculating the average number of compound nouns that result from each method.

Among these methods, method B generated the smallest number of compound nouns and showed reasonable effectiveness (Table 19). On the basis of this filtering method, we develop a smoothing method to solve the data sparseness problem. We combine the precision of rules with the mutual information of compound noun elements, and propose our final filtering method (H) as follows:

$$T(x, y) = \log_2 \frac{P(x, y)}{P(x) \times P(y)} + \alpha \times \textit{Precision}(\textit{the applied rule})$$

where α is a weighting coefficient. For the three-element compound nouns, the MI part is replaced by a three-element MI equation (see Section 4.3).

5.3. Compound noun weighting

We considered three weighting methods for compound nouns, which are shown in Table 14; one is a statistical method, used for single term weighting; and the others are methods which use the weight of the components of compound nouns. Experiments (in Section 6.1) showed that the statistical weighting scheme which was normally used for single term weighting is also the most effective for compound noun weighting.

6. Experiment results

In this section, we perform several retrieval experiments. To calculate the similarity between a document and a query, we use the p-norm retrieval model (Fox 1983) and use 2.0 as the p-value. We also use the component nouns in a compound as separate indexing terms (Strzalkowski et al. 1996). For single index terms, we use the weighting method atn.ntc (Lee 1995). We follow the standard TREC evaluation schemes for our evaluation (Salton and Buckley 1991).

Table 15. Performance of three weighting schemes.

	Scheme A	Scheme B	Scheme C
Recall	82.49	82.49	82.49
11-point average	38.25	36.82	35.82
Precision		(−3.74%)	(−6.27%)

6.1. Retrieval experiments using three compound noun weighting schemes

We performed experiments to determine the best weighting scheme (See Section 5.3) for compound nouns. We used KTSET1.0[4] as the test set (Table 15). Table 15 shows that Scheme A (statistical method) has the best precision. Scheme A also has the advantage of convenience in calculating the compound noun's weight. As a result, we will use scheme A for the other experiments.

6.2. Compound Noun Indexing Experiments

This experiment evaluates how well the proposed method can index diverse types of compound nouns compared with previous popular methods, which use human-generated compound noun indexing rules (Kim 1994, Lee et al. 1997). For simplicity, we filtered the generated compound nouns using the mutual information of compound noun elements at a threshold of zero (method A in Table 13).

Table 16 shows that the terms indexed by the linguistic approach are a subset of those generated by the statistical approach. It follows that the proposed method can cover more diverse compound nouns than the manual linguistic rule method. We also perform a retrieval experiment to evaluate our automatically extracted rules. Table 17 and Table 18 show that our method has as good a recall and 11-point average precision as the manual linguistic rule method.

6.3. Retrieval experiments using various filtering methods

In this experiment, we compare seven filtering methods to discover which one is best in terms of effectiveness and efficiency. We employed our automatic rules for compound

Table 16. Compound noun indexing coverage experiment (with a 200,000 eojeol Korean Information Base).

	With manual linguistic rule patterns	With our automatic rule patterns
Number of generated actual compound nouns	22,276	30,168 (+35.4%)
Number of generated actual compound nouns without overlap	0	7,892

Table 17. Compound noun indexing effectiveness experiment I.

	With manual linguistic rule patterns	With our automatic rule patterns
Average recall	82.66	83.62 (+1.16%)
11-pt. avg. precision	42.24	42.33 (+0.21%)
Number of index terms	504,040	515,801 (+2.33%)

(With KTSET2.0[5] test collection)

Table 18. Compound noun indexing effectiveness experiment II.

	With manual linguistic rule patterns	With our automatic rule patterns
Average recall	86.32	87.50 (+1.35 %)
11-pt. avg. precision	34.33	34.54 (+0.61 %)
Number of index terms	1,242,458	1,282,818 (+3.15 %)

(With KRIST[6] test collection)

Table 19. Retrieval experiment results of various filtering methods.

	A	B	C	D	E	F	G	H
Avg. rec.	83.62	83.62	83.62	83.62	83.62	83.62	84.32	**84.32**
		(+0.00)	(+0.00)	(+0.00)	(+0.00)	(+0.00)	(+0.84)	**(+0.84)**
11-pt. avg. pre.	42.45	42.42	42.49	42.55	42.72	42.48	42.48	**42.75**
		(−0.07)	(+0.09)	(+0.24)	(+0.64)	(+0.07)	(+0.07)	**(+0.71)**
Pre. at 10 Docs.	52.11	52.44	52.07	52.80	52.26	51.89	52.81	**52.98**
# of index terms	515,801	508,197	514,537	547,266	572,360	574,035	705,975	**509,895**
		(−1.47)	(−0.25)	(+6.10)	(+10.97)	(+11.29)	(+36.87)	**(−1.15)**

noun indexing, with the test collection KTSET2.0. We used recall and 11-point average precision to evaluate effectiveness, and used the number of index terms to measure efficiency.

Table 19 shows the results of the various filtering experiments. Method B generates the smallest number of compound nouns (best efficiency) and method H (our final proposing method) shows the best recall and precision (effectiveness) with the reasonable number of compound nouns (efficiency). We conclude that filtering method H is the best, in terms of both effectiveness and efficiency.

7. Conclusion

In this paper, we presented a method to extract compound noun indexing rules automatically from a large tagged corpus, and showed that this method can index compound

nouns appearing in diverse types of documents. For effectiveness, this method is as good as previous linguistic approaches but requires no human intervention. The proposed method also uses no parser or man-made rules, so can be applied to unrestricted texts very robustly and with a high domain portability. We also presented a filtering method to solve the compound noun over-generation problem. Our proposed filtering method (H) displays a fine retrieval performance both in terms of effectiveness and efficiency.

In future, we need to perform experiments on much larger commercial databases to test the practicality of our method. Although our filtering method efficiently decreases many spurious compound nouns, some compound nouns cannot be filtered out by using only syntactic and lexical information. To solve this problem, we need semantic information for the words in order to index compound nouns more efficiently. Finally, our method does not require language-dependent knowledge, so further research is needed to verify whether it can be easily applied to other languages.

Appendix

A: The POS (Part-Of-Speech) set of POSTAG

Tag	Description	Tag	Description
MC	common noun	MP	proper noun
MD	bound noun	T	pronoun
G	adnoun	S	numeral
B	adverb	K	interjection
DR	regular verb	DI	irregular verb
HR	regular adjective	HI	irregular adjective
I	assignment verb	E	existential predicate
jC	case particle	jS	auxiliary particle
jO	other particle	eGE	final ending
eGS	prefinal ending	eCNDI	aux conj ending
eCNDC	quote conj ending	eCNMM	nominal ending
eCNMG	adnominal ending	eCNB	adverbial ending
eCC	conjunctive ending	y	predicative particle
b	auxiliary verb	+	prefix
-	suffix	su	unit symbol
so	other symbol	s‘	left parenthesis
s’	right parenthesis	s.	sentence closer
s-	sentence connection	s,	sentence comma
sf	foreign word	sh	Chinese character

Acknowledgments

This research was partly supported by Ministry of Education through BK21 program toward Electrical and Computer Engineering Division at POSTECH and also by KOSEF special purpose basic research (1997.9–2000.8 #970-1020-301-3).

Notes

1. The frequency threshold can be adjusted according to the application system.
2. Courteously provided by ETRI, Korea.
3. Korean spacing unit which corresponds to an English word or phrase.
4. Courteously provided by KT, Korea (1,000 documents and 30 queries).
5. Courteously provided by KT, Korea (4,410 documents and 50 queries).
6. Courteously provided by KORDIC, Korea (13,514 documents and 30 queries).

References

Cha J, Lee G and Lee J-H (1998) Generalized unknown morpheme guessing for hybrid POS tagging of Korean. In: Proceedings of Sixth Workshop on Very Large Corpora in Coling-ACL 98.

Church KW and Hanks P (1990) Word association norms, mutual information, and lexicography. Computational Linguistics, 16(1):22–29.

Evans DA and Zhai C (1996) Noun-phrase analysis in unrestricted text for information retrieval. In: Proceeding of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA, pp. 17–24.

Fagan JL (1989) The effectiveness of a non-syntactic approach to automatic phrase indexing for document retrieval. JASIS, 40(2):115–132.

Fox EA (1983) Extending the Boolean and vector space models of information retrieval with p-norm queries and multiple concept types. Ph.D. Thesis, Cornell University.

Kando N, Kageura K, Yoshoka M and Oyama K (1998) Phrase processing methods for Japanase text retrieval. SIGIR Forum, 32(2):23–28.

Kim MJ, Park M, Chang H, Choi J and Lee SJ (1998) The generation methods of compound noun for efficient index term extraction. In: Proceedings of the 10th Conference of Korean and Korean Information Processing, pp. 121–129.

Kim PK (1994) The automatic indexing of compound words from Korean text based on mutual information. Journal of KISS, 21(7):1333–1340.

Lee H-A, Lee J-H and Lee G (1997) Noun phrase indexing using clausal segmentation. Journal of KISS, 24(3):302–311.

Lee JH (1995) Combining multiple evidence from different properties of weighting schemes. In: SIGIR'95, pp. 180–188.

Salton G and Buckley C (1991) Text REtrieval conferences evaluation program. In: ftp://ftp.cs.cornell.edu/pub/smart/.trec_eval.7.0beta.tar.gz.

Strzalkowski T, Guthrie L, Karlgren J, Leistensnider J, Lin F, Perez-Carballo J, Straszheim T, Wang J and Wilding J. (1996) Natural language information retrieval: TREC-5 report. In: The Fifth Text REtrieval Conference (TREC-5), NIST Special Publication. pp. 500–238.

Su K-Y, Wu M-W and Chang J-S (1994) A corpus-based approach to automatic compound extraction. In: Proceedings of ACL 94, pp. 242–247.

van Rijsbergen CJ (1979) Information Retrieval. Butterworths, London.

Won H, Park M and Lee G (2000) Integrated multi-level indexing method for compound noun processing. Journal of KISS, 27(1):84–95.

Yoon J-T, Jong E-S and Song M (1998) Analysis of Korean compound noun indexing using lexical information between nouns. Journal of KISS, 25(11):1716–1725.

Yun B-H, Kwak Y-J and Rim H-C (1997) A Korean information retrieval model alleviating syntactic term mismatches. In: Proceedings of the Natural Language Processing Pacific Rim Symposium, pp. 107–112.

Zhai C (1997) Fast statistical parsing of noun phrases for document indexing. In: Fifth Conference on Applied Natural Language Processing, pp. 312–319.