

Automatic Term Recognition based on Statistics of Compound Noun and its Components

Hiroshi Nakagawa
Information Technology Center,
The University of Tokyo
7-3-1, Bunkyo, Hongo
Tokyo, Japan, 113-0033
nakagawa@r.dl.itc.u-tokyo.ac.jp

Tatsunori Mori
Yokohama National University
79-5, Tokiwadai, Hodogaya
Yokohama, Japan, 240-0085
mori@forest.dnj.ynu.ac.jp

Abstract

In this paper, we propose a new idea of automatically recognizing domain specific terms. Our idea is based on the statistics between a compound noun and its component single-nouns. More precisely, we focus basically on how many nouns adjoin the noun in question to form compound nouns. We propose several scoring methods based on this idea and experimentally evaluate them on NTCIR1 TMREC test collection. The results are very promising especially in low recall area.

Introduction

Automatic term recognition, ATR in short, aims at extracting domain specific terms from a corpus of a certain academic or technical domain. The majority of domain specific terms are compound nouns, in other words, uninterrupted collocations. 85% of domain specific terms are said to be compound nouns. They include single-nouns of the rest 15% very frequently as their components, where “single-noun” means a noun which could not be further divided into several shorter and more basic nouns. In other words, the majority of compound nouns consist of much smaller number of the rest 15% single-noun terms and other single-nouns. In this situation, it is natural to pay our attention to the relation among single-nouns and compound nouns, namely how single-noun terms contribute to make up compound noun terms.

Another important feature of domain specific terms is *termhood* proposed in (Kageura &

Umino 96) where “termhood” refers to the degree that a linguistic unit is related to a domain-specific concept. Thus, what we really have to pursue is an ATR method which directly uses the notion of termhood.

Considering these factors, the way of making up compound nouns must be heavily related to the termhood of the compound nouns. The first reason is that termhood is usually calculated based on term frequency and bias of term frequency like inverse document frequency. Even though these calculations give a good approximation of termhood, still they are not directly related to termhood because these calculations are based on superficial statistics. That means that they are not necessarily meanings in a writer's mind but meanings in actual use. Apparently, termhood is intended to reflect this type of meaning. The second reason is that if a certain single-noun, say N , expresses the key concept of a domain that the document treats, the writer of the document must be using N not only many times but also in various ways. For instance, he/she composes quite a few compound nouns using N and uses these compound nouns in documents he/she writes. Thus, we focus on the relation among single-nouns and compound nouns in pursuing new ATR methods.

The first attempt to make use of this relation has been done by (Nakagawa & Mori 98) through the number of distinct single-nouns that come left or right of a single-noun term to make up compound noun terms. Using this type of number associated with a single-noun term, Nakagawa and Mori proposed the scoring function for term candidates. However their term extraction method is just one example of

making use of the relation among single-nouns and compound nouns. Note that this relation is essentially based on a noun bigram. In this paper, we expand this relation based on noun bigrams which might be the components of longer compound nouns. Then we experimentally evaluate the power of several variations of scoring functions based on this noun bigram relation using the NTCIR1 TMREC test collection. By this experimental clarification, we could conclude that the single-noun term's power of generating compound noun terms is really useful and essential in ATR.

In this paper, section 1 gives the background of ATR methods. Section 2 describes the proposed method of noun bigram based scoring function for term extraction. Section 3 describes the experimental results and discusses them.

1 Background

1.1 Typical Procedures of Automatic Term Recognition

An ATR procedure consists of two procedures in general. The first one is a procedure of extracting term candidates from a corpus. The second procedure is to assign each term candidate extracted by the first procedure a score that indicates how likely the term candidate is a term we would like to recognize. Once we assign a score to each term candidate, they are all ranked according to their scores. The remaining problem is how many terms we have to select from these ranked term candidates. It depends on various factors including how many terms we actually want to extract, who uses these extracted terms for what purpose, and so forth. Thus, in the rest of this subsection, we describe the background of a candidate extraction procedure and a scoring procedure respectively.

1.2 Candidates Extraction

The first thing to do in ATR is to extract *term candidates* from the given text corpus. Here we only focus on nouns, more precisely a single-noun and a compound noun, which are exactly the targets of the NTCIR1 TMREC task (Kageura et al 1999). To extract compound nouns which are promising term candidates and at the same time to exclude undesirable strings

such as “*is a*” or “*of the*”, the frequently used method is to filter out the words that are the member of a *stop-word-list*. More complex structures like noun phrases, collocations and so on, become focused on (Smadja and McKeown 1990; Frantzi and Ananiadou 1996). All of these are good term candidates in a corpus of a specific domain because all of them have a strong unithood (Kageura&Umino96) which refers to the degree of strength or stability of syntagmatic combinations or collocations. About terms having complex structure like compound nouns or collocations, we put the following basic assumption:

Assumption *Terms having complex structure are to be made of existing simple terms*

The structure of complex terms is another important factor for automatic term candidates extraction. It is expressed syntactically or semantically. As a syntactic structure, dependency structures that are the results of parsing are focused on in many works. Since we focus on these complex structures, the first task in extracting term candidates is a morphological analysis including part of speech (POS) tagging. In Japanese, which is an agglutinative language, a morphological analysis segments out words from a sentence and does POS tagging (Matsumoto et al. 1996).

After POS tagging, the complex structures mentioned above are extracted as term candidates. Previous studies have proposed many promising ways for this purpose, Hisamitsu(2000) and Nakagawa (1998) concentrated their efforts on compound nouns. Smadja and McKeown (1990), and Frantzi and Ananiadou (1996) tried to treat more general structures like collocations.

1.3 Scoring

The next thing to do is to assign a score to each term candidate in order to rank them in descending order of termhood. Many researchers have sought the definition of term candidate's score which approximates termhood. In fact, many of those proposals make use of surface statistics like *tf·idf*. Ananiadou et al. proposed C-value (Frantzi and Ananiadou 1996) and NC-value (Frantzi and Ananiadou 1999) which

count how independently the given compound noun is used in the given corpus. Hisamitsu and Niwa (1999) and Hisamitsu (2000) propose a way to measure termhood which counts how far the given term is different from the distribution of non-domain-specific terms. Kageura et al. (2000) use the frequency with which a Japanese word and its English counterpart co-occur in a bi-lingual document set. All of them tried to capture how important and independent a writer regards and uses individual terms in a corpus

2 Single-Noun Bigrams as Components of Compound Nouns

2.1 Single-Noun Bigrams

Although the relation between a single-noun and complex nouns that include this single-noun is very important. Nevertheless, to my knowledge, this relation has not been paid enough attention so far. Nakagawa and Mori (1998) proposed a term scoring method that utilizes this type of relation. In this paper, we extend their idea comprehensively. Here we focus on compound nouns among the various types of complex terms. In technical documents, the majority of domain-specific terms are noun phrases or compound nouns consisting of small number of single nouns. Considering this observation, we propose a new scoring method that measures the importance of each single-noun. In a nutshell, this scoring method for a single-noun measures how many distinct compound nouns contain a particular single-noun as their part in a given document or corpus. Here, think about the situation where single-noun: N occurs with other single-nouns which might be a part of many compound nouns shown in Figure 1 where [N M] means bigram of noun N and M.

[LN1 N](#L1)	[N RN1](#R1)
[LN2 N](#L2)	[N RN2](#R2)
⋮	⋮
[LNn N](#Ln)	[N RNm](#Rm)

Figure 1. Noun Bigram and their Frequency

In Figure 1, [LN_i N] (i=1,...,n) and [N RN_j] (j=1,...,m) are single-noun bigrams which make

(a part of) compound nouns. #L_i and #R_j (i=1,...,n and j=1,...,m) mean the frequency of the bigram [LN_i N] and [N RN_j] respectively. Note that since we depict only bigrams, compound nouns like [LN_i N RN_j] which contains [LN_i N] and/or [N RN_j] as their parts might actually occur in a corpus. Again this noun trigram might be a part of longer compound nouns. We show an example of noun bigram. Suppose that we extract compound nouns including “trigram” as candidate terms from a corpus shown in the following example.

Example 1.

trigram statistics, word trigram, class trigram, word trigram, trigram acquisition, word trigram statistics, character trigram

Then, noun bigrams consisting of a single-noun “trigram” are shown in the following where the number between (and) shows the frequency.

word trigram (3)	trigram statistics (2)
class trigram (1)	trigram acquisition (1)
character trigram(1)	

Figure 2. An example of noun bigram

We just focus on and utilize single-noun bigrams to define the function on which scoring is based. Note that we are concerned only with single-noun bigrams and not with a single-noun per se. The reason is that we try to sharply focus on the fact that the majority of domain specific terms are compound nouns whose basic statistics is a noun bigram.

2.2 Scoring Function

2.2.1 Direct score of noun bigram

Since a scoring function based on [LN_i N] or [N RN_j] could have infinite number of variations, we here consider the following simple but representative scoring functions.

#LDN(N) and #RDN(N) : These are the number of distinct single-nouns which directly precede or succeed N. These are exactly “n” and “m” in Figure 1. For instance, in an example shown in Figure 2, #LDN(trigram)=3, #RDN(trigram)=2

LN(N,k) and **RN(N,k)**: The general function that takes into account the number of occurrence of each noun bigram is defined for [LN_i N] and [N RN_j] as follows respectively.

$$LN(N,k) = \sum_{i=1}^{\#LDN(N)} (\#Li)^k \quad (1)$$

$$RN(N,k) = \sum_{j=1}^{\#RDN(N)} (\#Rj)^k \quad (2)$$

We find various functions with varying the parameter: k of (1) and (2). For instance, #LDN(N) and #RDN(N) can be defined as **LN(N,0)** and **RN(N,0)**. **LN(N,1)** and **RN(N,1)** are the frequencies of nouns that directly precede or succeed N. For instance, in an example shown in Figure 2, LN(trigram,1)=5, and RN(trigram,1)=3. Now we think about the nature of (1) and (2) with various value of the parameter k. The larger k is, the more we take into account the frequencies of each noun bigram. One extreme is the case k=0, namely LN(N,0) and RN(N,0), where we do not take into account the frequency of each noun bigram at all. LN(N,0) and RN(N,0) describe how linguistically and domain dependently productive the noun : N is in a given corpus. That means whether noun : N presents a key and/or basic concept of the domain treated by the corpus. Other extreme cases are large k, like k=2, 4, etc. In these cases, we rather focus on frequency of each noun bigram. In other words, statistically biased use of noun : N is the main concern. For example, about an example shown in Figure 2, LN(trigram,2)=11, and RN(trigram,2)=5. If k<0, we discount the frequency of each noun bigram. However, this case does not show good results of in our ATR experiment.

2.2.2 Score of compound nouns

The next thing to do is to extend those scoring functions of single-noun to the scoring functions of compound nouns. We adopt very simple method that is a geometric mean. Now think about a compound noun : CN = N₁ N₂...N_L. Then a geometric mean: **GM** of CN is defined as follows.

$$GM(CN,k) = \left(\prod_{i=1}^L (LN(N_i,k) + 1)(RN(N_i,k) + 1) \right)^{1/2L} \quad \text{--- (3)}$$

For instance, if we use LN(N,1) and RN(N,1) in example 1, GM(trigram,1) = $\sqrt{(3+1) \times (5+1)} = 4.90$. In (3), GM does not depend on the length of a compound noun that is the number of single-nouns within the compound noun. This is because since we have not yet had any idea about the relation between the importance of a compound noun and a length of the compound noun, it is fair to treat all compound nouns, including single-nouns, equally no matter how long or short each compound noun is.

2.2.3 Combining Compound Noun Frequency

The information we do not use in the bigram based methods described in 2.2.1 and 2.2.2 is the frequency of single-nouns and compound-nouns that occur independently, namely left and right adjacent words not being nouns. For instance, “word patterns” occurs independently in “... use the word patterns occurring in ...”. Since the scoring functions proposed in 2.2.1 is noun bigram statistics, the number of this kind of independent occurrences of nouns themselves are not used. If we take this information into account, the new type of information is used and the better results are expected. In this paper, we take very simple method for this. Namely, if a single-noun or a compound noun occurs independently, the score of the noun is multiplied by the number of its independent occurrences. Then GM(CN,k) of the formula (3) is revised as follows. We call this new GM as **FGM(CN,k)** which is defined as follows.

if N occurs independently

$$\text{then } FGM(CN,k) = GM(CN,k) \times f(CN)$$

where f(CN) means the number of independent occurrences of noun CN (4)

For instance, in example 1, if we find independent “trigram” three times in the corpus, FGM(trigram,1) = $3 \times \sqrt{(3+1) \times (5+1)} = 14.70$

2.2.4 Modified C-value

We compare our methods with C-value based method(Frantzi and Ananiadou 1996) because

1) their method is very powerful to extract and properly score compound nouns., and 2) their method is basically based on unithood. On the contrary, our scoring functions proposed in 2.2.1 try to capture termhood. However their original definition of C-value can not score a single-noun because the important part of the definition C-value is:

$$C\text{-value}(a) = (\text{length}(a) - 1)(n(a) - \frac{t(a)}{c(a)}) \quad (5)$$

where a is compound noun,

$\text{length}(a)$ is the number of single-nouns that consist of a ,

$n(a)$ is the total frequency of occurrence of a on the corpus,

$t(a)$ is the frequency of occurrence of a in longer candidate terms,

$c(a)$ is the number of those candidate terms.

As known from (5), all single-noun's C-value come to be 0. The reason why the first term of right hand side is $(\text{length}(a)-1)$ is that C-value originally seemed to capture how much computational effort is to be made to recognize important part of the term. Thus, if the $\text{length}(a)$ is 1, we do not need any effort to recognize its part because the term a is a single-word and does not have its part. But we intend to capture how important the term is for the writer or reader, namely its termhood. In order to C-value be suitable to capture termhood, we modify (5) as follows.

$$MC\text{-value}(a) = \text{length}(a)(n(a) - \frac{t(a)}{c(a)}) \quad (6)$$

Where “MC-value” means “Modified C-value.”

MC-value(trigram) of Example 1 is

$$1 \cdot (8 - 8/5) = 6.4$$

3 Experimental Evaluation

3.1 Experiment

In our experiment, we use the NTCIR1 TMREC test collection (Kageura et al 1999). As an activity of TMREC, they have provided us with the Japanese test collection of the term recognition task. The goal of this task is to automatically recognize and extract terms from the text corpus which contains 1,870 abstracts gathered from the computer science and communication engineering domain corpora of

NACSIS Academic Conference Database, and 8,834 manually collected correct terms. The TMREC text corpus is morphologically analyzed and POS tagged by hand. From this POS tagged text, we extract uninterrupted noun sequences as term candidates. Actually 16,708 term candidates are extracted and several scoring methods are applied to them. All the extracted term candidates: CNs are ranked according to the descending order of $GM(CN, k)$, $FGM(CN, k)$ and $MC\text{-value}(CN)$. As for parameter k of (1) and (2), we choose $k=1$ because its performance is the best among various value of k between 0 and 4. Thus, henceforth, we omit k from GM and FGM , like $GM(CN)$ and $FGM(CN)$. We use $GM(CN)$ as the baseline.

In evaluation, we conduct the experiments where we pick up the highest ranked term candidate down to the PNth highest ranked term candidate by these three scoring methods, and evaluate the set of selected terms with the number of correct terms, we call it CT, within it. In the following figures, we only show CT because recall is $CT/8834$, where 8834 is the number of all correct terms, precision is CT/PN . Another measure NTCIR1 provides us with is the terms which include the correct term as its part. We call it “longer term” or LT. They are sometimes valued terms and also indicate in what context the correct terms are used. Then we also use the number of longer terms in our evaluation.

3.2 Results

In Figure 3 through 5, PN of X-axis means PN.

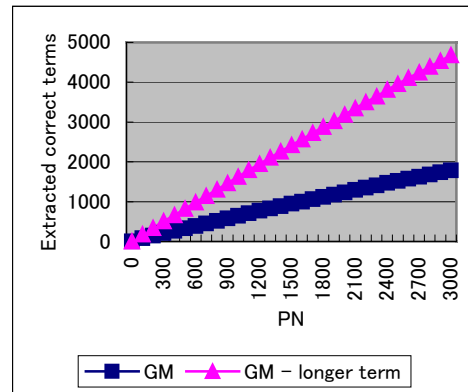


Figure 3. CT and LT of GM(CN) for each PN

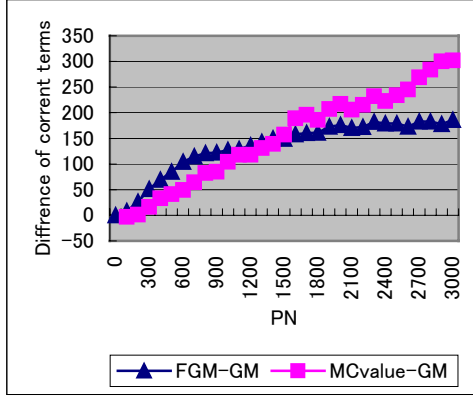


Figure 4. CT of FGM(CN) minus CT of GM(CN), and CT of MC-value(CN) minus CT of GM(CN) for each PN

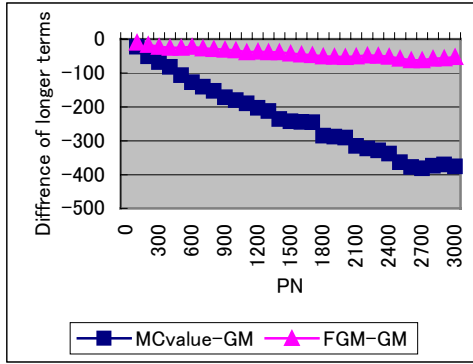


Figure 5. LT of GM(CN) minus LT of FGM(CN), and LT of GM(CN) minus LT of MC-value(CN) for each PN

In Figure 3, the Y-axis means CT meaning the number of correct terms picked up by GM(CN) and the number of longer terms picked up by GM(CN) for each PN. They are our baseline. The Figure 4 shows the difference between CT of FGM(CN) and CT of GM(CN) and the difference between CT of MC-value(CN) and CT of GM(CN) for each PN. Figure 5 shows the difference between LT of GM(CN) and LT of FGM(CN) or LT of MC-value(CN) for each PN. As known from Figure 4, FGM based method outperforms MC-value up to 1,400 highest ranked terms. Since in the domains of TMREC task that are computer science and communication engineering, 1,400 technical terms are important core terms, FGM method we propose is very promising to extract and recognize the domain specific terms. We also

show CT of each method for larger PN, say, from 3000 up to 15000 in Table 1 and 2.

Table 1. CT of each ranking method for PN larger than 3000

PN	GM	FGM	MC-value
3000	1784	1970	2111
6000	3286	3456	3671
9000	4744	4866	4930
12000	6009	6090	6046
15000	7042	7081	7068

Table 2. LT of each ranking method for PN larger than 3000

PN	GM	FGM	MC-Value
3000	2893	2840	2531
6000	5644	5576	5011
9000	8218	8152	7578
12000	10523	10488	9852
15000	12174	12186	12070

As seen on these figures and tables, if we want more terms about these domains, MC-value is more powerful, but when PN is larger than 12,000, again FGM outperforms. As for recognizing longer terms, GM(CN), which is the baseline, performs best for every PN. MC-value is the worst. From this observation we come to know that MC-value tends to assign higher score to shorter terms than GM or FGM. We are also interested in what kind of term is favored by each method. For this, we show the average length of the highest PN ranked terms of each method in Figure 6 where length of CN means the number of single-words CN consists of. Clearly, GM prefers longer terms. So does FGM. On the contrary, MC-value prefers shorter terms. However, as shown in Figure 6, average length of MC-value is more fluctuating. That means GM and FGM have more consistent tendency in ranking compound nouns. Finally we compare our results with NTCIR1 results (Kageura et al 1999). Unfortunately since (Kageura et al 1999) did not provide recall, precision and F-measure of top 3,000 to 15,000 term candidates, we could not directly compare our results with other

NTCIR1 participants. They only provide the number of the whole extracted terms and also the number of the whole extracted correct terms. Then, what is important is the fact that we extracted 7,082 correct terms from top 15,000 term candidates with FGM methods. This fact is indicating that our methods show the highest performance among all other participants of NTCIR1 TMREC task because 1) the highest number of terms within the top 16,000 term candidates is 6,536 among all the participants of NTCIR1 TMREC task, and 2) the highest number of terms in all the participants of NTCIR1 TMREC task is 7,944, but they are extracted from top 23,270 term candidates, which means extremely low precision.

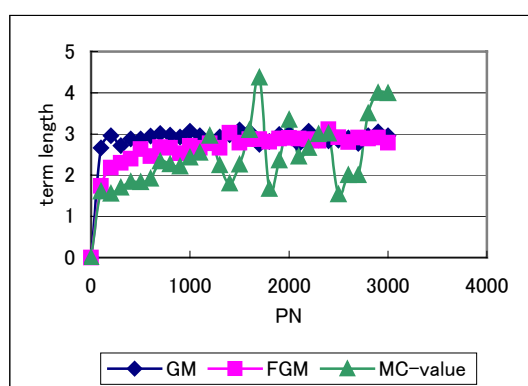


Figure 6. The average length of extracted terms by GM(CN), FGM(CN) and MC-value(CN) for each PN

Conclusion

In this paper, we introduce single-noun bigram based statistical methods for ATR, which are basically how many nouns adjoin the single-noun in question to form compound nouns. Through experimental evaluation using NTCIR1 TMREC test collection, the method FGM we propose shows best performance in selecting up to 1,400 domain specific terms. The proposed idea has potentially more variations that we have to pursue as our future problem.

Acknowledgements

This research is funded by the ministry of Education Science and Academic, Japan.

References

- Frantzi, T.K. and Ananiadou, S. 1996. "Extracting nested collocations". In *Proceedings of 16th International Conference on Computational Linguistics*, 41-46.
- Frantzi, T.K. and Ananiadou, S. 1999. "The c-value/nc-value method for atr". *Journal of Natural Language Processing* 6(3), 145-179.
- Hisamitsu, T. and Niwa, Y. 1999. "Term Extraction Using a New Measure of Term Representativeness". In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, 475-484.
- Hisamitsu, T. 2000. "A Method of Measuring Term Representativeness". In *Proceedings of 18th International Conference on Computational Linguistics*, 320-326.
- Kageura, K. and Umino, B. 1996. "Methods of automatic term recognition: a review". *Terminology* 3(2), 259-289.
- Kageura, K. et al. 1999. "TMREC Task: Overview and Evaluation". In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, 411-440.
- Kageura, K., Tsuji, K. and Aizawa, A. 2000. "Automatic Thesaurus Generation through Multiple Filtering". In *Proceedings of 18th International Conference on Computational Linguistics*, 397-403.
- Matsumoto, Y., Kurohashi, S., Yamaji, O., Taeki, H. and Nagao, M. 1996. *Instruction Manual of Japanese Morphological Analyzer JUMAN3.1*. Nagao Lab. at Kyoto University.
- Nakagawa, H. and Mori, T. 1998. "Nested Collocation and Compound Noun for Term Recognition". In *Proceedings of the First Workshop on Computational Terminology COMPTERM'98*, 64-70.
- Smadja, F.A. and McKeown, K.R. 1990. "Automatically extracting and representing collocations for language generation". In *Proceedings of the 28th Annual Meetings of the Association for Computational Linguistics*, 252-259.