Automatic Term Recognition based on Statistics of Compound Nouns

Hiroshi Nakagawa

Abstract
In this paper, we propose a new idea of automatically recognizing domain specific terms of compound noun which is one of several definitions of collocation. Our idea is based on single-noun statistics associated with noun bigrams. 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 high recall area.

1 Introduction
Automatic term recognition, ATR in short, aims at extracting domain specific terms from a monolingual corpus of a certain academic or technical domain. The majority of domain specific terms are compound nouns, in other words, uninterrupted collocations. Of course interrupted collocations are much harder to extract and recognize from corpus. However in documents of a certain academic or technical domain, the domain specific terms we need to extract is uninterrupted collocations. For instance, complex name of bio-chemical substance, names of complex methodology in information science such as “genetic algorithm” are terms to express key concepts of the domain. Apparently an uninterrupted collocation is a compound word. Moreover 85% of domain specific terms are said to be compound nouns. They include single-nouns of the rest 15% as their components, where “single-noun” means a noun which could not be further divided anymore into several shorter and more basic 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 meanings in a writer's mind but meanings in use. Apparently, termhood is intended to reflect the former 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.

To my knowledge, the first attempt to make use of this relation has been done by (Enguehard and Pantera94). (Nakagawa and Mori98) formalizes this idea using 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 proposed the scoring function for term candidates and their experimental results showed high recall and precision in NTCIR1 TMREC task (Kageura et al 1999). 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 single-noun bigram. In this paper, we generalize
this relation based on single-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 2 gives the background of ATR methods. Section 3 describes the proposed method of noun bigram based scoring function for term extraction. Section 4 describes the experimental results and discussion about them. Section 5 is our conclusions.

2 Background

2.1 Standard Procedure of ATR

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 in 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. Considering these factors, at this moment, this selection problem has not yet been scientifically well defined. Thus, in the rest of this subsection, we describe the background of a candidate extraction procedure and a scoring procedure respectively.

2.2 Candidates Extraction

The first thing to do in ATR is to extract term candidates from the given text corpus. We concentrate word-based term candidates. Here we only focus on nouns, more precisely a single-noun or a compound noun, which are exactly the targets of the NTCIR1 TMREC task. To extract compound nouns as promising term candidates and at the same time to exclude undesirable strings such as “is a” or “of the”, the most frequently used method is to filter out the words that are the member of a so called stop-word-list. More complex structures like interrupted collocations become focused on (Smadja and McKeown 1990; Frantzi and Ananiadou 1996, Hisamitsu and Nitta 1996, Shimohata et al. 1997). Either interrupted or uninterrupted collocations are good term candidates in a corpus of a specific domain because of their strong unithood (Kageura and Umino96) which refers to the degree of strength or stability of syntagmatic combinations or collocations. For terms of complex structure like compound nouns or collocations, we make the following basic assumption:

Assumption Terms of 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. Since we focus on uninterrupted collocations, the first task to extract term candidates is a morphological analysis including part of speech (POS) tagging. In English, POS tagging has been one of the main issues of natural language processing, and high quality POS taggers such as (Brill 1994) have already been widely used. In Japanese, which is an agglutinative language, a morphological analyzer segments out words from a sentence and does POS tagging simultaneously (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 and Nitta (1996), Hisamitsu(2000) and Nakagawa (1998) concentrated their efforts on compound nouns. Smadja and McKeown (1990), Daille et al. (1994), Frantzi and Ananiadou (1996) and Shimohata et al. (1997) tried to treat more general structures including interrupted collocations.

2.3 Scoring

Once term candidates have been extracted from a text corpus, the next thing to do is to assign a termhood based score to each term candidate in order to rank them in descending order of termhood. Obviously, terms with high termhood should obtain a high score. However, to directly
measure termhood of the given term candidate is extremely difficult because only the writer of a document could or might know which terms are qualified as necessary and sufficient terms in the given documents. Many researchers have tried to work out the definition of term candidate’s score which approximates termhood. In fact, many of those proposals have been calculated on the basis 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. In this sense, their proposal is basically a unithood-based method. 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. Basically their measures use word co-occurrence, log-likelihood ratio of distribution. 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 termhood as distribution of term candidates in a corpus. On the contrary (Nakagawa and Mori 1998) tried to capture termhood more directly. In this paper, we basically employ and try to generalize the method proposed in (Nakagawa and Mori 1998).

3 Single-Noun Bigrams as Components of Compound Nouns

3.1 Single-Noun Bigrams

As said in section 1, the relation between a single-noun and complex nouns those include this single-noun is very important. 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 documents of technical domain, the majority of domain-specific terms are compound nouns. In spite of the huge number of technical terms consisting of compound nouns, very few single-nouns contribute to make these compound nouns. This is a basic observation we rely on. 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 corpus of specific domain. Here, think about the situation where single-noun: N occurs with other single-nouns which might be a part of many compound nouns such as shown in Figure 1 where [N M] means bigram of noun N and M.

\[
\begin{align*}
[LN1 N] (#L1) & \quad [N RN1](#R1) \\
[LN2 N] (#L2) & \quad [N RN2](#R2) \\
& \vdots \\
[LNn N](#Ln) & \quad [N RNm](#Rm)
\end{align*}
\]

Figure 1. Noun Bigram and their Frequency

In Figure 1, \([LNi N] (i=1,..,n)\) and \([N RNj] (j=1,...,m)\) are single-noun bigrams which make a part of compound nouns. \#Li and \#Rj \((i=1,..,n\) and \(j=1,...,m)\) mean the frequency of the bigram \([LNi N] \) and \([N RNj]\), respectively. Note that since we depict only bigrams, compound nouns like \([LNi N RNj]\) which contains \([LNi N] \) and/or \([N RNj]\) 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 as follows.

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.

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

Figure 2. An example of noun bigram

We 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.
3.2 Scoring Function

3.2.1 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 focus on 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, with the notation of Figure 1.

\[
\#LDN(N) = \sum_{i=1}^{k} (#Li)^k
\]
\[
\#RDN(N) = \sum_{j=1}^{k} (#Rj)^k
\]

We find various functions with varying the parameter: k of (1) and (2). For instance, \#LDN(N,0) and \#RDN(N,0) can be defined as \#LN(N,0) and \#RN(N,0). \#LN(N,1) and \#RN(N,1) are the number of total occurrences of nouns that directly precede or succeed N. For instance, about 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 single-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 single-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 single-noun bigram. However, this case does not show good results of in our ATR experiment.

3.2.2 Entropy of noun

The most traditional way to measure the amount of information carried by single-noun bigram is an entropy of single-noun: N in the whole single-noun bigrams including N. The definition of left entropy of N: \(H_L(N)\) and right entropy of N: \(H_R(N)\) are defined by the following formulae.

\[
M_L = \sum_{i=1}^{#LDN(N)} #Li, \quad M_R = \sum_{j=1}^{#RDN(N)} #Rj
\]

\[
H_L(N) = -\sum_{i=1}^{#LDN(N)} \frac{#Li}{M_L} \log \frac{#Li}{M_L}
\]

\[
H_R(N) = -\sum_{j=1}^{#RDN(N)} \frac{#Rj}{M_R} \log \frac{#Rj}{M_R}
\]

These are information theoretical measures of N.

3.2.3 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 3.2.1 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:

\[
c(a) = (\text{length}(a) - 1)(\text{n}(a) - \frac{t(a)}{c(a)})
\]

where a is compound noun, 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(i) is the number of those candidate terms. As known from (5), all single-noun’s C-value come to be 0. Thus we can not score each single-noun properly by (5). Then, in order to C-value be able to score single-nouns, we modify (6) as follows.

\[
MC\text{-}value(a) = \text{length}(a)(\text{n}(a) - \frac{t(a)}{c(a)})
\]

Where “MC-value” means “Modified C-value.” MC-value(trigram) of Example 1 is \((7 - 7/5) = 5.6\)
3.2.4 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. First of all, for the direct scores described in 3.2.1 and the entropy described in 3.2.2, we use the symbol $FL(N)$ to represent one of $#LDN(N)$, $#LN(N,k)$, or $H_2(N)$ and $FR(N)$ to represent one of $#RDN(N)$, $#RN(N,k)$ or $H_8(N)$. Now think about a compound noun : $CN = N_1 N_2 \ldots N_L$. Then a geometric mean: $GM$ of $CN$ is defined as follows.

$$GM(N) = \left( \prod_{i=1}^{L} (FL(N_i)+1)(FR(N_i)+1) \right)^{1/L}$$

(7)

For instance, if we use $#LN(N,1)$ and $#RN(N,1)$ in example 1, $GM(\text{trigram}) = \sqrt[3]{(3+1)(5+1)} = 4.90$. In (7), $GM$ 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 the compound noun is.

On the other hand, $MC$-value itself gives the score of compound noun as known in (6). Thus in our experiment, we directly use $MC$-value($CN$) of (6) as a score.

3.2.5 Frequency of Independent Occurrence of Single-Noun

The information we do not use in the bigram based methods described in 3.2.1 and 3.2.2 is the frequency of single-nouns that occur in a corpus not as a part of compound nouns. We call this occurrence as “independent occurrence.” For instance, in “… use the patterns occurring in …”, the single-noun “patterns” occurs independently. Since the scoring functions proposed in 3.2.3 is single-noun bigram statistics, the number of this kind of independent occurrences of single-nouns themselves are not used. If we take this information into account, the new information is added and the better results are expected. In this paper, we take very simple method for this. Namely, if a single-noun occurs independently, the score of the single-noun is multiplied by the number of its independent occurrences. Then $GM(N)$ of the formula (7), in this case $L=1$, namely $CN$ is a single-noun, is revised. We call this new GM as $GM_1(N)$ which is defined as follows.

$$GM_1(N) = f(N) \times GM(N)$$

(8)

For instance, in example 1, if we find independent occurrence of “trigram” three times in the corpus, $GM_1(\text{trigram}) = 3 \times \sqrt[3]{(3+1)(5+1)} = 14.70$

4 Experimental Evaluation

4.1 Experiment

In our experiment, we use the NTCIR1 TMREC test collection (Kageura et al 1999). The NTCIR1 TMREC group called for participation of the term recognition task that is a part of NTCIR1 held in 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 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 are ranked according to the descending order of $GM(CN)$ for each single-noun bigram based method, $GM_1(CN)$ for $#LN(N,1)$ and $#RN(N,1)$, or $MC$-value($CN$). For evaluation, we use five predetermined number: PN, namely 3000, 6000, 9000, 12000 and 15000, for selecting term candidates as correct terms of each scoring function. More precisely speaking, we conduct the experiments where we pick up the top PN term candidates and evaluate these picked up terms with recall, precision and F-measure. As scoring methods of compound nouns, we adopt $GM(CN)$ based on single-noun scoring functions described in 3.2.1, 3.2.2, and
3.2.3, GM1(CN) described in 3.2.5, and MC-value based method.

4.2 Bigram Based Methods

Firstly we examine the methods proposed in 3.2.1 that we call bigram based methods for k=0, 0.25, 0.5, 1, 2 and 4 of formula (1) and (2). This is because we want to know the effect of the value of parameter k in (1) and (2). The scores of compound nouns : GM(CN) of (7) are shown in Table 1.

Table 1. The number of correct terms and F-measure of GM based on LN(N,k),RN(N,k)

<table>
<thead>
<tr>
<th>PN</th>
<th>0</th>
<th>0.25</th>
<th>0.5</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>1746</td>
<td>1743</td>
<td>1772</td>
<td>1784</td>
<td>1778</td>
<td>1769</td>
</tr>
<tr>
<td>9000</td>
<td>4713</td>
<td>4694</td>
<td>4704</td>
<td>4744</td>
<td>4714</td>
<td>4711</td>
</tr>
<tr>
<td>15000</td>
<td>7036</td>
<td>7037</td>
<td>7035</td>
<td>7042</td>
<td>7048</td>
<td>7045</td>
</tr>
</tbody>
</table>

In each cell, the upper low shows the number of extracted terms and the lower low shows F-measure.

This results indicates that k=1 of (1) and (2) is the best performance especially in low PN area. That means that since the method which most directly uses the statistics of left and right nouns of a single-noun bigram is showing the best performance. However, the differences between the results with other values of k of (1) and (2) are not significant. Considering these results, the performance of these bigram based method essentially owe to the idea of single-noun bigrams stated in 3.1.

4.3 Comparison of Results

Now we compare the methods proposed in 3.2.1, 2,3,4 and 5. The results of five scoring methods are shown in Table 2 and 3. In Table 2, the number of correct terms given by the NTCIR1 TMREC test collection within top PN term candidates are shown. In Table 3, recall, precision and F-measure of top PN term candidates each of five method are shown. Among bigram based methods, we apply multiplying f(N) of the formula (8) proposed in 3.2.5 to the best one, namely the GM based on #LDN(N,1) and #RDN(N,1) and it is symbolized as GM1 in Table 2 and 3. In Table 2 and 3, LDN, RN(1), Entropy and MC-value means the scoring method based on #LDN and #RDN, #LN(N,1) and #RN(N,1), H_L and H_R , and MC-value, respectively.

Table 2. The number of correct terms of five methods for each PN

<table>
<thead>
<tr>
<th>PN</th>
<th>LDN</th>
<th>RN(1)</th>
<th>GM1</th>
<th>Entropy</th>
<th>MC-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>1746</td>
<td>1784</td>
<td>1789</td>
<td>1755</td>
<td>2111</td>
</tr>
<tr>
<td>6000</td>
<td>3270</td>
<td>3286</td>
<td>3293</td>
<td>3306</td>
<td>3671</td>
</tr>
<tr>
<td>9000</td>
<td>4713</td>
<td>4747</td>
<td>4761</td>
<td>4661</td>
<td>4930</td>
</tr>
<tr>
<td>12000</td>
<td>5974</td>
<td>6009</td>
<td>6025</td>
<td>5979</td>
<td>6046</td>
</tr>
<tr>
<td>15000</td>
<td>7036</td>
<td>7042</td>
<td>7082</td>
<td>7035</td>
<td>7068</td>
</tr>
</tbody>
</table>

Table 3. Recall, Precision, and F-measure of five methods for each PN

<table>
<thead>
<tr>
<th>PN</th>
<th>LDN</th>
<th>RN(1)</th>
<th>GM1</th>
<th>Entropy</th>
<th>MC-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>6000</td>
<td>.370</td>
<td>.372</td>
<td>.372</td>
<td>.374</td>
<td>.415</td>
</tr>
<tr>
<td>9000</td>
<td>.533</td>
<td>.536</td>
<td>.537</td>
<td>.527</td>
<td>.557</td>
</tr>
<tr>
<td>12000</td>
<td>.676</td>
<td>.680</td>
<td>.681</td>
<td>.676</td>
<td>.684</td>
</tr>
<tr>
<td>15000</td>
<td>.796</td>
<td>.796</td>
<td>.801</td>
<td>.796</td>
<td>.799</td>
</tr>
</tbody>
</table>

In each cell of this table, from top to bottom, recall, precision and F-measure are shown in this order.

We also conducted experiments where we use an arithmetic mean in stead of a geometric mean for scoring compound nouns. However, the results are almost the same or little worse. Then we only showed the results of a geometric mean.

4.4 Discussions

As seen from Table 2 and 3, all of the bigram based methods show almost the same results except for GM1 based method. Among bigram
based methods, entropy based method is inferior to others. That can be interpreted as follows: The importance as term is not directly counted as the amount of mathematical information, but rather counted as the number of occurrences in the corpus. In other words, termhood is not calculated with entropy.

Next, we compare MC-value based method with other methods. As seen in Table 2 and 3, MC-value based method is clearly superior to other methods especially in low PN area. However, in high PN area, say the case where 15,000 terms are selected, the scoring method based of GM1 outperforms MC-value based method.

Now we compare the results from different point of view, namely the length of extracted compound terms. The length of an extracted compound noun, including single-nouns, is counted by the number of single-nouns that consist of the compound nouns. Of course, the length of a single-noun term is 1. The average length of compound nouns for intervals of 1 – 3000, 3001-6000, … and 12001-15000 terms are shown in Table 4.

<table>
<thead>
<tr>
<th>PN</th>
<th>LDN RDN</th>
<th>LN (1) RDN</th>
<th>LN (1) RN</th>
<th>GM1 Entropy</th>
<th>MC-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1~3000</td>
<td>2.78</td>
<td>2.91</td>
<td>2.79</td>
<td>2.68</td>
<td>2.48</td>
</tr>
<tr>
<td>3001~6000</td>
<td>2.88</td>
<td>2.88</td>
<td>2.88</td>
<td>2.76</td>
<td>2.79</td>
</tr>
<tr>
<td>6001~9000</td>
<td>2.72</td>
<td>2.69</td>
<td>2.70</td>
<td>2.79</td>
<td>3.05</td>
</tr>
<tr>
<td>9001~12000</td>
<td>2.46</td>
<td>2.45</td>
<td>2.46</td>
<td>2.75</td>
<td>2.42</td>
</tr>
<tr>
<td>12001~15000</td>
<td>1.94</td>
<td>2.94</td>
<td>1.99</td>
<td>2.06</td>
<td>2.00</td>
</tr>
</tbody>
</table>

As seen in Table 4, clearly the average length of extracted terms within 1 – 3000 interval of MC-value is much shorter than that of other methods. And in this interval, MC-value results in high recall and precision. That means that correct terms are averagely short and MC-value can assign high value to shorter terms. However, intervals of 3001 – 6000 up to 12001 – 15000 the average length of terms scored by MC-value are not that shorter or even longer than other methods. Parallel to this tendency, extracted correct terms, recall and precision of other methods and those of MC-value become closer. And finally GM1 overrides MC-value when top 15000 terms are selected. However this tendency should not be regarded as the matter of good or bad. Rather it is a character of each method that means the preference for short terms or long terms. This character is to be used as a guideline for selecting scoring method where the purpose of ATR is given.

In sum, taking into account these strong points and weak points of each method, if we want small number of high quality terms or short terms, MC-value based method is recommended. But if we want a list of terms as comprehensive as possible, and this is the case where automatic term extraction system is useful, then bigram based methods, especially GM1 based method is recommended to use.

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 bigram based GM1 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 or 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.

5 Conclusions

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 with NTCIR1 TMREC test collection, the proposed idea which has many
variations including single-noun’s entropy is essentially showing high performance in ATR. MC-value which is the modified C-value is superior in low PN area, but the noun bigram based method: GM1 which takes into account the number of independent occurrence of single-nouns outperforms in high PN area. The proposed idea has potentially more variations that we have to pursue as our future problem.

References


