Using Author Keywords for Automatic Term Recognition

Masao Utiyama, Masaki Murata, Hitoshi Isahara

Communications Research Laboratory, MPT
2-2-2 Hikaridai Seika-cho, Kyoto 619-0289, Japan
{mutiyama,murata,isahara}@crl.go.jp

Abstract

This paper proposes a method which regards the keywords provided by the authors of technical papers as terms and learns the statistics which distinguish terms from non-terms. Since it uses keywords as training data, it requires no hand-labeled training corpora manually annotated with terms. The proposed method was used to extract terms from the NTCIR morphologically tagged corpus and achieved 0.800 recall and 0.431 precision. The effectiveness of the proposed method has thus been demonstrated.

Keywords

automatic term recognition, supervised learning, statistical measure of termhood, author keywords as training data

1 Introduction

Wide coverage dictionaries are vital for natural language processing. Such dictionaries, however, are difficult to compile manually, especially in technical domains, because new terms are constantly created to represent new concepts. Automatic term recognition is thus necessary to avoid human efforts of compiling dictionaries as much as possible.

Most work on automatic term recognition has taken either unsupervised learning approaches or rule-based approaches (Kageura 1996), i.e., researchers have tried using various statistical measures such as mutual information or tf·idf to characterize terms, or they have written lexical rules to define patterns of terms.

Recently, Demetriou and Gaizauskas (2000) have proposed a bootstrapping approach to automatic term recognition. They used seed knowledge, i.e., a list of terms, in order to discover co-occurrence patterns for the terms in technical texts. Their method identifies patterns and new terms in an iterative manner and has achieved promising results without using hand-labeled training corpora. Other bootstraping approaches are reported in (Enguehard and Pantera 1994; Frantzi and Ananiadou 1999). Bootstraping approaches are used to extend and enhance initial term lists. Initial term lists are either given beforehand (Enguehard and Pantera 1994; Demetriou and Gaizauskas 2000) or acquired automatically (Frantzi and Ananiadou 1999).

We propose a method which regards the keywords provided by the authors of technical papers (author keywords) as terms and learns the statistics which distinguish terms from non-terms. Thus the method requires neither hand-labeled training corpora manually annotated with terms nor a list of seed terms.

We discuss the details of the method in Section 3 and give some experimental results in Section 4. First, however, we describe the morphologically tagged corpus used in the NTCIR workshop

(National Center for Science Information Systems 1999) to set the context for the subsequent discussion.

2 NTCIR corpus

We first describe the NTCIR morphologically tagged corpus and the tagset used in the corpus, and then describe the modification which we made to the original tagset. Next, we discuss the feasibility of using author keywords for automatic term recognition.

2.1 Tagged corpus

We used the morphologically tagged corpus from the NTCIR workshop to evaluate our method. The corpus consists of 1870 Japanese abstracts in the field of artificial intelligence. The abstracts are part of the NACSIS (National Center for Science Information Systems) Academic Conference Database.

The texts in the corpus are morphologically analyzed. The morphemes are tagged for part-of-speech (POS) and type of origin (Kageura, Koyama, and Yoshioka 1999a). For example, a title "類推のための抽象化" "abstraction for analogical reasoning" is morphologically analyzed as follows:

Morpheme	類推	の	ため	の	抽象	化
POS	NS	SCC	NR	SCC	NS	TLNS
Origin	K	W	W	W	K	K

where 'NS' and 'NR' mean nouns, 'SCC' means a conjunction, and 'TLNS' means a suffix. The types 'W' and 'K' are shown in Table 1 along with the other types.

In our experiments, we used simplified POS tags. For example, we unified the noun tags 'NS', 'NN', 'NC', 'NM', 'NP', 'NPH', 'NPG', 'NF', 'NT' and 'NR' into 'N'. We also combined the simplified POS tags and the types of origin into a tagset. Thus, with our tagset, the above example is tagged as shown in Table 2.

Table 1: Types of origin

Types	Tag	Comments	Examples
Original Japanese	W	Wago	仕組み, する, られる
Chinese Origin	K	Kango	組織, 考察, 被害, 者
Non Chinese Origin	G	Gairaigo	システム, オリジナル
Mixed	M	Mixture	同時に, 最後に, 四万七千

This table is based on a table in (Kageura et al. 1999a).

Table 2: Tagged morphemes

Morpheme	類推	の	ため	の	抽象	化
Tag	N/K	SCC/W	N/W	SCC/W	N/K	TLN/K

2.2 Author keywords as terms

We automatically extracted 4223 author keywords from the keyword field in the corpus. Of these, 3059 keywords occurred in the abstract field. The other keywords were listed in the keyword field but did not appear in the abstract field. These 3059 keywords are henceforth referred to as Author-Keywords.

The recall and precision of Author-Keywords vis-a-vis Manual-Candidates, i.e., the 8834 terms provided by the NTCIR TMREC group, were:

$$\text{Recall} = \frac{\text{number of matched terms}}{\text{number of terms in Manual-Candidates}} = \frac{1820}{8834} = 0.206,$$

$$\text{Precision} = \frac{\text{number of matched terms}}{\text{number of terms in Author-Keywords}} = \frac{1820}{3059} = 0.595.$$

The recall is low, but the precision is high. This means that many terms are not included in Author-Keywords but author keywords are likely to be terms. Since author keywords are likely to be terms, we should be able to extract the characteristics of terms from Author-Keywords.

3 Automatic term recognition

Before describing the details of our method, we present our assumption about terms. We assume that the origin of a term X could be an author keyword. When X first appears as an author keyword, X might not be a term because it is not familiar in the community. Gradually, X becomes popular, then it is regarded as a term by the community. Under this assumption, the high precision shown in Section 2.2 is not a coincidence but a necessary consequence. Therefore we use author keywords as training data for automatic term recognition.

Below, we first describe the format of training data used in our method and then describe a statistical measure which distinguishes terms from non-terms. The measure needs a threshold for the discrimination. The threshold is set automatically, as discussed in Section 3.3.

3.1 Format of training data

The morphemes in a sequence of morphemes are tagged with '1' if the sequence matches an author keyword, otherwise they are tagged with '0'. We put the special symbol 'EOS', which is always tagged with '0', at the beginning and at the end of a sentence so that the measure defined in Equation (1) is properly calculated for ordinary words. For example, "類推のための抽象化" "abstraction for analogical reasoning" is tagged as shown in Table 3 when "類推" "analogical reasoning" and "抽象化" "abstraction" are author keywords.

Table 3: Tagged morphemes

Morpheme	EOS	類推	の	ため
Tag	$\langle$EOS, 0$\rangle$	$\langle$N/K, 1$\rangle$	$\langle$SCC/W, 0$\rangle$	$\langle$N/W, 0$\rangle$
	の	抽象	化	EOS
	$\langle$SCC/W, 0$\rangle$	$\langle$N/K, 1$\rangle$	$\langle$TLN/K, 1$\rangle$	$\langle$EOS, 0$\rangle$

3.2 Statistical measure of termhood

We first define the termhood of a morpheme and then define the termhood of a term, which is calculated from the termhoods of the constituent morphemes.

Let w_{i-1}, w_i, w_{i+1} be a sequence of morphemes and t_{i-1}, t_i, t_{i+1} be their tags. Then, the termhood of w_i is defined by:

$$K_i = \log \frac{\Pr(\langle t_{i-1}, * \rangle, \langle t_i, 1 \rangle, \langle t_{i+1}, * \rangle)}{\Pr(\langle t_{i-1}, * \rangle, \langle t_i, 0 \rangle, \langle t_{i+1}, * \rangle)} \tag{1}$$

where $\Pr(event)$ is the probability of *event* in the training data and '$*$' matches either 1 or 0. For example, $\Pr(\langle \mathrm{EOS}, 0 \rangle, \langle \mathrm{N/K}, 1 \rangle, \langle \mathrm{SCC/W}, 0 \rangle)$ is the probability of the sequence "$\langle \mathrm{EOS}, 0 \rangle, \langle \mathrm{N/K}, 1 \rangle, \langle \mathrm{SCC/W}, 0 \rangle$" in the training data. We call K_i the K-measure of word i.

Equation (1) takes a high value when the ratio of $\Pr(\langle t_{i-1}, * \rangle, \langle t_i, 1 \rangle, \langle t_{i+1}, * \rangle)$ to $\Pr(\langle t_{i-1}, * \rangle, \langle t_i, 0 \rangle, \langle t_{i+1}, * \rangle)$ is high. This means that it takes a high value when w_i is likely to be tagged with $\langle t_i, 1 \rangle$, which means that w_i is likely to be a part of a token of a term. Note that we do not use words in Equation (1). This is because we want to acquire general patterns of terms.

Equation (1) can be decomposed as follows if we assume $\langle t_{i-1}, * \rangle$ and $\langle t_{i+1}, * \rangle$ are statistically independent of each other.

$$K_i = \log \frac{\Pr(\langle t_i, 1 \rangle | \langle t_{i-1}, * \rangle) \Pr(\langle t_{i+1}, * \rangle | \langle t_i, 1 \rangle)}{\Pr(\langle t_i, 0 \rangle | \langle t_{i-1}, * \rangle) \Pr(\langle t_{i+1}, * \rangle | \langle t_i, 0 \rangle)} \tag{2}$$

where

$$\begin{aligned}
\Pr(\langle t_i, 1 \rangle | \langle t_{i-1}, * \rangle) &= \frac{\mathrm{Freq}(\langle t_{i-1}, 1 \rangle, \langle t_i, 1 \rangle) + \mathrm{Freq}(\langle t_{i-1}, 0 \rangle, \langle t_i, 1 \rangle) + 0.5}{\mathrm{Freq}(\langle t_{i-1}, 1 \rangle) + \mathrm{Freq}(\langle t_{i-1}, 0 \rangle) + 0.5}, \\
\Pr(\langle t_{i+1}, * \rangle | \langle t_i, 1 \rangle) &= \frac{\mathrm{Freq}(\langle t_i, 1 \rangle, \langle t_{i+1}, 1 \rangle) + \mathrm{Freq}(\langle t_i, 1 \rangle, \langle t_{i+1}, 0 \rangle) + 0.5}{\mathrm{Freq}(\langle t_i, 1 \rangle) + 0.5}, \\
\Pr(\langle t_i, 0 \rangle | \langle t_{i-1}, * \rangle) &= \frac{\mathrm{Freq}(\langle t_{i-1}, 1 \rangle, \langle t_i, 0 \rangle) + \mathrm{Freq}(\langle t_{i-1}, 0 \rangle, \langle t_i, 0 \rangle) + 0.5}{\mathrm{Freq}(\langle t_{i-1}, 1 \rangle) + \mathrm{Freq}(\langle t_{i-1}, 0 \rangle) + 0.5}, \\
\Pr(\langle t_{i+1}, * \rangle | \langle t_i, 0 \rangle) &= \frac{\mathrm{Freq}(\langle t_i, 0 \rangle, \langle t_{i+1}, 1 \rangle) + \mathrm{Freq}(\langle t_i, 0 \rangle, \langle t_{i+1}, 0 \rangle) + 0.5}{\mathrm{Freq}(\langle t_i, 0 \rangle) + 0.5},
\end{aligned}$$

where Freq(*event*) is the frequency of *event* in the training data and 0.5 is used for smoothing probability estimation. For example, Freq($\langle$N/K, 1$\rangle$, $\langle$SCC/W, 0$\rangle$) is the number of occurrences of the sequence "$\langle$N/K, 1$\rangle$, $\langle$SCC/W, 0$\rangle$" in the training data.

3.2.1 Term extraction

A term is defined to be a maximum length morpheme sequence whose morphemes have K-values which are greater than a threshold.

Given "類推のための抽象化" "abstraction for analogical reasoning," for example, we can calculate the K-measures of the morphemes as shown in Figure 1 and properly extract "類推" "analogical reasoning" and "抽象化" "abstraction" as the terms.

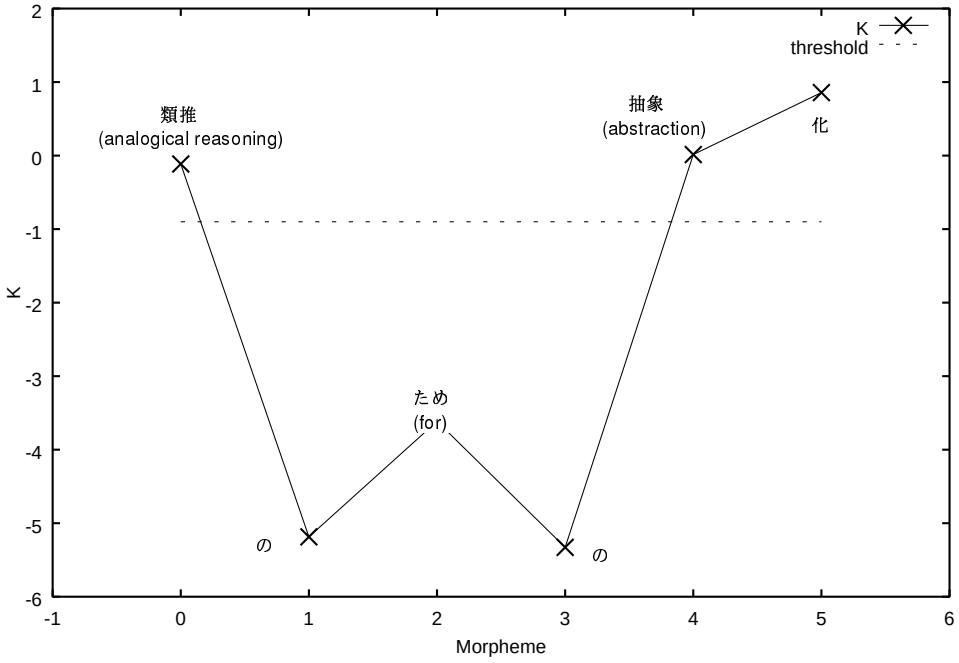

Figure 1: K-measure

The termhood of a token of a term which occurs as a morpheme sequence $w_i, w_{i+1}, ..., w_j$ is

defined by:

$$K_i^j = \sum_{k=i}^{j} K_k. \tag{3}$$

The termhood of a type is the sum of the termhoods of the tokens of the type. If the tokens of term X occur at positions $i_1...j_1$, $i_2...j_2$, ... and $i_n...j_n$ then its termhood is defined by:

$$K(X) = \sum_{k=1}^{n} K_{i_k}^{j_k}. \tag{4}$$

$K(X)$ is used to rank types of terms and is discussed in Section 4.

The accuracy of automatic term extraction and the ranking of terms crucially depend on the threshold to be used in separating morphemes into two sets; one set consisting of the morphemes included in terms and the other set consisting of the morphemes not included in terms.

3.3 Distribution of the K-measure

Let W_1 be the set of morphemes included in Author-Keywords and let W_0 be the set of morphemes not included in Author-Keywords. Let K_1 and K_0 be the set of K-measures for the morphemes in W_1 and W_0, respectively.

Figure 2 shows the distributions of K_1 and K_0 calculated by using the data obtained from the NTCIR tagged corpus. The horizontal axis indicates the K-measure and the vertical axis indicates the frequency of morphemes. The Figure also shows the distributions of the K-measures over T_1 and T_0, where T_1 is the set of K-measures for the morphemes which are included in Manual-Candidates and T_0 is the set of K-measures for the morphemes not included in Manual-Candidates. Thus T_0 and T_1 are the ideal cases of K_0 and K_1.

The distributions of K_* and T_* are similar. The distributions of K_1 and T_1 have clear peaks around $K = 0.5$. The distributions of K_0 and T_0, on the other hand, have several peaks. But K_1 and K_0 (T_1 and T_0) are reasonably apart. So, if we set an appropriate threshold, they should be distinguished from each other with high precision. Note that the peak of T_1 is higher than that

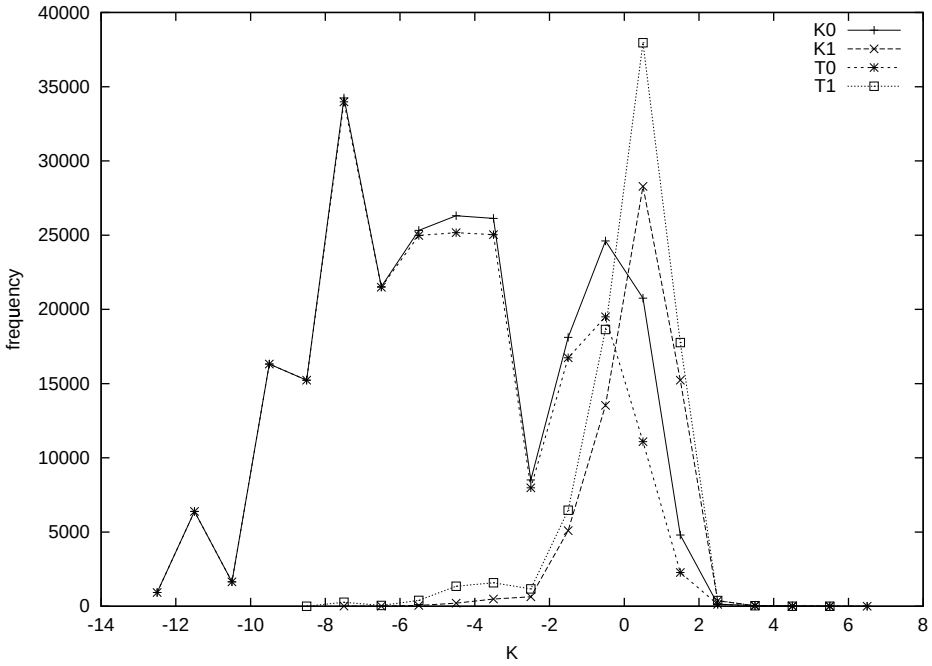

Figure 2: Distributions of K_1 and K_0

of K_1. The difference is the number of morphemes which are included in Manual-Candidates but not included in Author-Keywords.

3.4 Threshold

The threshold for the K-measure is obtained by applying linear discriminant analysis(Tanaka and Wakimoto 1983).

Let μ_1 and μ_0 be the means of K_1 and K_0, respectively, and let σ_1 and σ_0 be the standard deviations of K_1 and K_0, respectively. Then, the threshold ϕ is defined by:

$$\phi = \frac{\mu_1\sigma_0 + \mu_0\sigma_1}{\sigma_1 + \sigma_0}.$$

The value of ϕ was -0.9047 when we applied linear discriminant analysis to the data obtained from the NTCIR tagged corpus. The F-measure ($= \frac{2\times\text{Precision}\times\text{Recall}}{\text{Recall}+\text{Precision}}$) of the terms extracted from the NTCIR tagged corpus vis-a-vis Manual-Candidates was 0.5358 when we used $\phi = -0.9047$ as the threshold.

This F-measure is nearly optimum, as shown in Figure 3, which shows the change in the F-measure when we varied the threshold from -4 to 2 in steps of 0.1. The best F-measure 0.5351 was obtained with a threshold of -0.9. The best F-measure was actually lower than the F-measure obtained by setting the threshold to -0.9047, which demonstrates the effectiveness of applying linear discriminant analysis to obtain the threshold for the K-measure.

4 Experiment and evaluation

As was described in Section 2, we extracted terms from the NTCIR tagged corpus and investigated the performance of our method vis-a-vis Manual-Candidates.

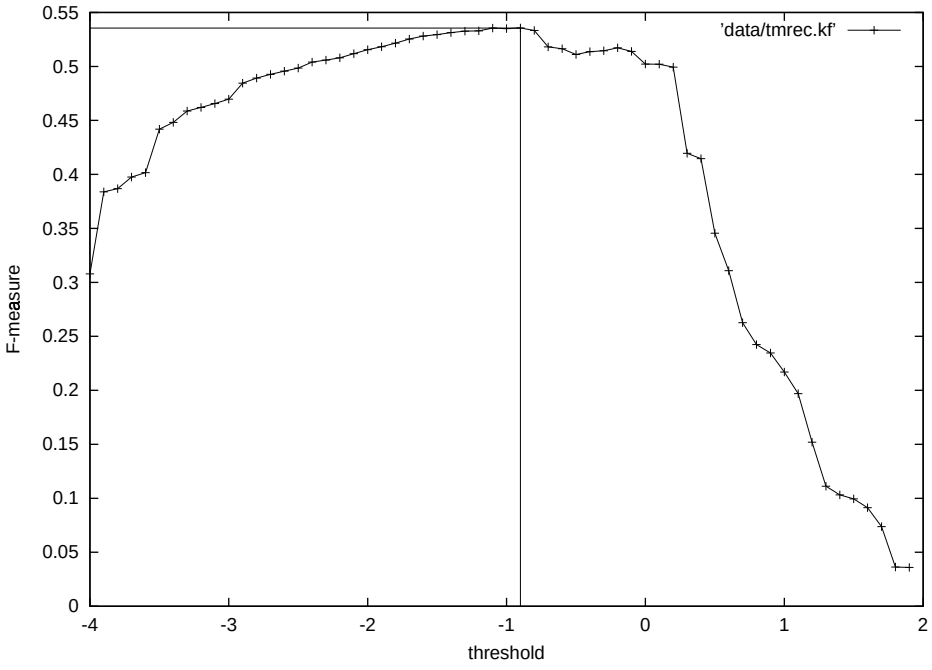

Figure 3: Threshold for the K-measure vs. F-measure

4.1 Term recognition without filtering

First, we evaluated the method explained in Section 3 without modification. We used the NTCIR tagged corpus to estimate the probabilities used in Equation (1). The method extracted 17600 terms from the NTCIR tagged corpus. The recall, precision and F-measure of the extracted terms were:

$$\text{Recall} = \frac{\text{number of matched terms}}{\text{number of terms in Manual-Candidates}} = \frac{7082}{8834} = 0.802,$$

$$\text{Precision} = \frac{\text{number of matched terms}}{\text{number of extracted terms}} = \frac{7082}{17600} = 0.402,$$

$$\text{F-Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Recall} + \text{Precision}} = 0.536.$$

4.2 Term recognition with simple filtering

Next, we selected from the extracted terms those noun phrases whose length (number of characters) was greater than one, where a noun phrase was defined as an extracted term ending with a noun or suffix. We then discarded *hiragana* and numeric phrases. The number of terms remaining was 16382 and the performance was:

Recall	Precision	F-measure
0.800	0.431	0.560

This simple filtering improved precision significantly, with a slight degradation of recall. We henceforth refer to these 16382 terms as Candidates.

4.2.1 Recall vs. precision curve

We sorted the terms in Candidates in descending order of the K-measure defined in Equation (4). We also sorted the terms in descending order of their frequencies. Figure 4 shows the recall vs. precision curves of these ranking methods.

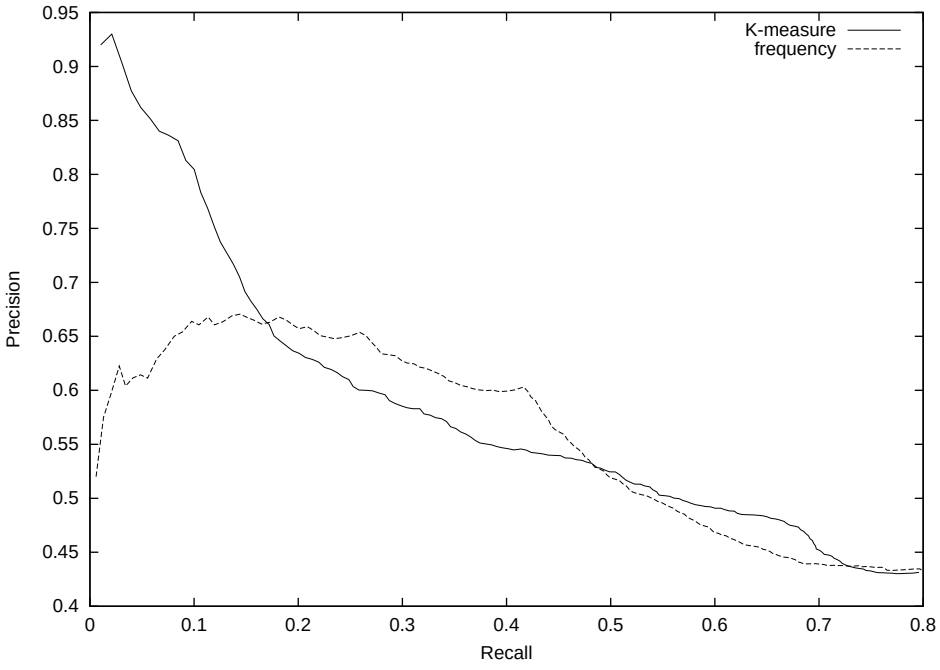

Figure 4: Recall vs. Precision

Figure 4 clearly shows that the terms ranked high according to K-measure are indeed terms vis-a-vis Manual-Candidates. On the other hand, the terms ranked high according to frequency are not necessarily terms. On the contrary, precision for those higher ranked terms is relatively low.

Thus, Figure 4 suggests that the K-measure defined in Equation (4) provides a very good reflection of the characteristics of Manual-Candidates.

4.3 Extracted terms

A good method for automatic term recognition should give us some insight into the nature of terms. This section demonstrates that the K-measure indeed gives us such an insight.

First, we show the frequency of tag patterns. Then, we describe the growth curves of tag patterns, which is calculated from the K-measure. The growth curves reveal productive trends in term formulation which may not be discovered without using the K-measure.

4.3.1 Frequency of tag patterns

The frequencies and ratios of the patterns in the tag sequences whose ratios are greater than 0.05 are listed in Table 4. The most frequent pattern in Candidates is "N/K N/K", two nouns of Chinese origin. Table 4 shows static statistics for the tag patterns in Candidates.

Table 4: Major patterns

Frequency	Ratio	Pattern	Terms
4199	0.256	N/K N/K	問題解決 (problem solving) 知識獲得 (knowledge acquisition)
1776	0.108	N/K	知識 (knowledge) 問題 (problem) 学習 (learning)
1144	0.070	N/G	システム (system) モデル (model) ユーザ (user)
1093	0.067	N/K N/K N/K	自然言語処理 (natural language processing)
991	0.060	N/K N/G	知識ベース (knowledge base) 対象モデル (object model)
849	0.052	N/G N/K	オブジェクト指向 (object oriented) フレーム問題 (frame problem)

4.3.2 Growth curves of tag patterns

Dynamic statistics are shown in Figure 5, which shows the growth curves of the patterns in Table 4. The horizontal axis indicates the ranks of the terms in Candidates sorted according to K-measure, and the vertical axis indicates the cumulative ratios of the patterns. The cumulative ratio of pattern X at rank R, $C(X, R)$, is defined by:

$$C(X, R) = \frac{\text{number of pattern } X \text{ terms ranked within the top } R \text{ terms}}{\text{number of pattern } X \text{ terms in Candidates}}$$

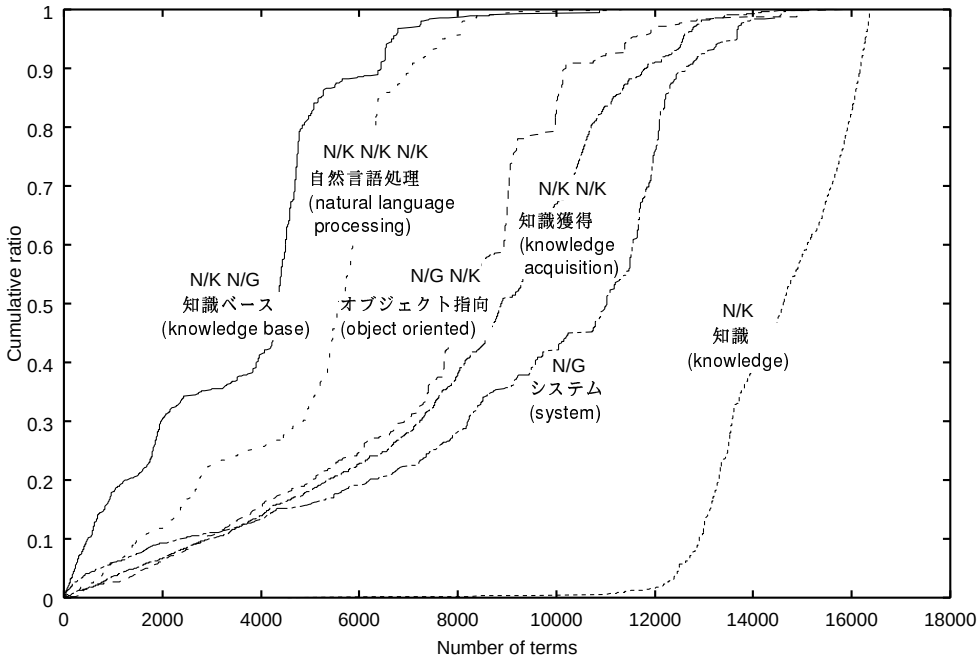

Figure 5: Growth curves of tag patterns

Figure 5 shows that most terms of pattern "N/K N/G" are ranked high. On the other hand, most terms of pattern "N/K" are ranked low. The figure suggests the termhoods of the patters are ordered as

"N/K N/G" > "N/K N/K N/K" > "N/G N/K" > "N/K N/K" > "N/G" > "N/K".

This order agrees with our intuition.

First, in general, the termhood of non-Chinese origin words (G, *gairaigo* or *katakana words*) is greater than that of Chinese origin words (K or *kango*). This is because *katakana* is used to represent relatively new words borrowed from foreign languages, especially from English, while *kango* is used to represent common words. The order "N/G" > "N/K" directly follows this observation.

Second, the termhood of a term strongly depends on the termhood of the head of the term. The head of a Japanese term is the last word of the term. Thus the head of "N/K N/G" is "N/G" and the head of "N/G N/K" and "N/K N/K" is "N/K." So, the termhood of "N/K N/G" depends on that of "N/G" and the termhood of "N/G N/K" and "N/K N/K" depends on that of "N/K." This observation, together with the first observation, explains the order "N/K N/G" > "N/G N/K" > "N/K N/K".

Finally, the number of component words of a term is usually two or three (Kageura, Yoshioka, Tsuji, Yoshikane, Takeuchi, and Koyama 1999b). This observation agrees with the fact that the termhood of two or three word terms is greater than that of one word terms.

Moreover, the curves in the Figure depict productive trends in term formulation, which enhances our understanding of the quantitative structures of terms.

4.4 Upper bound and baseline

We estimated the upper bound of the performance of our method by using Manual-Candidates instead of Author-Keywords to train our method. We extracted terms by using the method described in Section 3 and then filtered the extracted terms according to the method described in Section 4.2. The number of remaining terms was 17756 and the performance was:

Recall	Precision	F-measure
0.871	0.433	0.579

We also tried a baseline method. We defined a candidate term as a sequence of nouns, prefixes, or suffixes and then extracted all the candidate terms in the corpus. The number of extracted candidate terms was 24339 and the performance was:

Recall	Precision	F-measure
0.829	0.301	0.441

Table 5 compares our method with the upper bound and the baseline method. The precision of our method is comparable to that of the upper bound and outperforms that of the baseline method. This demonstrates the effectiveness of our method, a method which uses author keywords as training data. On the other hand, the recall of our method is relatively low compared with the upper bound. This means that Author-Keywords showed little variation in the types of keywords, with the result that our method could not extract the types of keywords which did not appear in Author-Keywords.

Table 5: Comparison of performances

Method	Recall	Precision	F-measure
Upper bound	0.871	0.433	0.579
Our method	0.800	0.431	0.560
baseline method	0.829	0.301	0.441

5 Conclusion

We have proposed a method which uses author keywords as training data for automatic term recognition. The idea of using author keywords as training data is applicable to many other academic domains since abstracts of academic papers usually have keywords.

The proposed method achieved 0.800 recall and 0.431 precision. Though the recall was somewhat lower than the recall obtained when we used Manual-Candidates as training data, the precision was comparable to that of using Manual-Candidates as training data. Characteristic

patterns of terms were also extracted. Thus our method is effective both for automatic term recognition and for revealing productive trends in term formulation.

The features used for estimating probabilities in our method are rather simple; we used only part-of-speech and type of origin. More specific features such as suffixes, prefixes, or words may be useful for estimating precise probabilities. It is also possible to use more sophisticated methods for estimating probabilities (Manning and Schütze 1999).

Reference

Demetriou, G., and Gaizauskas, R. (2000). "Automatically Augmenting Terminological Lexicons from Untagged Text." In *LREC'2000*, pp. 861–867.

Enguehard, C., and Pantera, L. (1994). "Automatic Natural Acquisition of a Terminology." *Journal of Quantitative Linguistics*, *2*(1), 27–32.

Frantzi, K. T., and Ananiadou, S. (1999). "The *C-value/NC-value* domain independent method for multi-word term extraction." *Journal of Natural Language Processing*, *6*(3), 145–179.

Kageura, K. (1996). "Methods of Automatic Term Recognition." *Terminology*, *3*(2), 259–289.

Kageura, K., Koyama, T., and Yoshioka, M. (1999a). *Table for POS, Inflection and Types of Origin*. http://www.rd.nacsis.ac.jp/~ntcadm/workshop/sampleatr-d-t-eng.html.

Kageura, K., Yoshioka, M., Tsuji, K., Yoshikane, F., Takeuchi, K., and Koyama, T. (1999b). "Evaluation of the Term Recognition Task." In *NTCIR Workshop 1*, pp. 417–434.

Manning, C. D., and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press.

National Center for Science Information Systems (1999). *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition.* National Center for Science Information Systems.

Tanaka, Y., and Wakimoto, K. (1983). *Methods of Multivariate Statistical Analysis.* Gendai Suugaku-sya.