A Study on Terminology Extraction

Based on Classified Corpora

Chen Yirong[1], Lu Qin[1], Li Wenjie[1], Sui Zhifang[2],Ji Luning[1]

[1]Department of Computing, Hong Kong Polytechnic University
[2]Institute of Computational Linguistics, Peking University, China
E-mail: csyrchen@comp.polyu.edu.hk, csluqin@comp.polyu.edu.hk,
cswjli@comp.polyu.edu.hk,szf@pku.edu.cn,cslji@comp.polyu.edu.hk

Abstract

Algorithms for automatic term extraction in a specific domain should consider at least two issues, namely *Unithood* and *Termhood*(Kageura,1996). *Unithood* refers to the degree of a string to occur as a word or a phrase. *Termhood* (Chen Yirong, 2005) refers to the degree of a word or a phrase to occur as a domain specific concept. Unlike *unithood*, study on *termhood* is not yet widely reported. In classified corpora, the class information provides the cue to the nature of data and can be used in *termhood* calculation. Three algorithms are provided and evaluated to investigate *termhood* based on classified corpora. The three algorithms are based on lexicon set computing, term frequency and document frequency, and the strength of the relation between a term and its document class respectively. Our objective is to investigate the effects of these different *termhood* measurement features. After evaluation, we can find which features are more effective and also, how we can improve these different features to achieve the best performance. Preliminary results show that the first measure can effectively filter out independent terms or terms of general use.

1. Introduction

A term is a lexical unit to effectively represent a domain concept (Zhifang SUI, 2002). Terminology extraction is extracting terminology from a set of documents. Manual terminology extraction is most likely unaffordable in terms of both time and cost. So it is desirable to extract terminology automatically. Automatic terminology exaction is a major topic in natural language processing and has a wide variety of applications such as dictionary generation, and keyword extraction for information retrieval.

Algorithms for automatic term extraction in a specific domain should consider at least two issues, namely *Unithood* and *Termhood* (Kageura, 1996). Unithood refers to the degree of a string to occur as a word or a phrase. Termhood (Chen Yirong, 2005) refers to the degree of a word or a phrase to occur as a domain specific term representing certain concepts in that domain.

Many works have been done on unithood calculation. For instance, mutual information [Church et al. 1990] and *log*-likelihood ratio [Dunning 1993; Cohen 1995] have been widely used for extracting word bi-grams.

Relatively speaking, less study is done on termhood. The distribution of a term occurring within a domain or across domains provides cues to the nature of the term. We have made three observations which are useful in termhood calculation using term distribution information. Firstly, since a term is used to represent domain concept, it is obvious that a term is mostly like to occur in the specific domain and not so often in the general domain. Thus with the availability of classified corpora, we can extract term candidates from different corpora. By examining the intersections and differences of these corpora, we can find those domain specific terms.

Secondly, even within a specific domain, the distribution of a term in documents contains information of the relativity between the terms and the domain. A typical phenomenon is that a term is more likely to occur many times in just several documents where the related subjects are discussed. While general words are distributed more evenly in documents.

Thirdly, the distribution of a term in different domains can also indicate the relationship between the term and its domain. Naturally, a term candidate is more likely to be a terminology in one domain if it occurs much more frequent than in other domains.

Our algorithms are based on the above three observations. The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the design of these algorithms. Section 4 gives performance evaluation. Section 5 is the conclusion.

2. Related works

Many researches have been done on term extraction in a specific domain (terminology extraction). Various methods for measuring the domain specificity of a word have been proposed in term extraction.

[Nakagawa, 2002] made an assumption for automatic recognition of domain specific terms, that is, “terms having complex structure are to be made of existing simple terms”. So he only focused on the relation between single-noun and compound noun. The compound nouns can be determined as domain specific by measuring and scoring each single-noun as their part in a given document or corpus. But this method cannot deal with non-compound terminology.

The most commonly used measurement for termhood measurement is Term Frequency Inverse Document Frequency (TFIDF). It calculates the termhood by combining word frequency with a document and word occurrence within a set of documents.

[E. Frank, 1999] focused on domain-specific key phrase extraction. He considered only two attributes for discriminating between key phrases and non-key phrases—the TF-IDF score of a phrase, and the position of the phrase’s first appearance in the whole document. However, classical measures such as TF-IDF are so sensitive to term frequencies that they fail to avoid very frequent non-informative words. And since Frank mainly

focused on key phrases of a document, the second feature may not help much in extracting terminology.

[Hisamitsu, 2000] used the baseline method for defining the representative-ness of a term. The document set which contains all the documents is labeled as D_0. Documents that contains the term T is labeled as D(T) . If a term is topic specific, all the terms in D(T) should probably have different distribution in D_0. They developed a method called the baseline method to compute the difference between the two distributions. Baseline method cannot handle some "background noise"—words which are irrelevant to term T and simply happen to occur in D(T) (documents containing T). This is the part to be offset by the baseline function. Based on the idea of the baseline method, [Hisamitsu, 2002] used another approach to measure the bias of word occurrences. The number of words whose saliency over a threshold value is taken as the degree of bias of word occurrences in D(T). The algorithm has good performance, but its running time is quite long.

[Jing-Shin Chang, 2005] proposed a statistical model for finding domain specific words (DSW). He defined Inter-Domain Entropy (IDE) by acquiring normalized relative frequencies of occurrence of terms in various domains. Terms whose IDE are above a threshold are unlikely to be associated with any certain domain. Then the top-k% candidates can be determined as the domain specific words of the domain.

The above two works used only one of the three observations in the introduction, respectively. In the next section we will present our algorithms which use the three observations, respectively.

3. Algorithm design

In classified corpora, class information provides cues to the data and can be used in termhood calculation. Therefore, if a new term is extracted from set of corpora of the same domain, it is most likely that it belongs to that classified domain. In other words, if we are trying to find new software related terms, we can first try to extract them from a software related corpus. On the other hand, if you have a new term candidate, we can also try to verify its domain through examination of corpora of different domains. Generally speaking, suppose we have a general corpus in Chinese, labeled *G*. and a domain specific corpus, labeled *S*. Then, the lexicon set obtained from S - G contains the terms used in the domain of software.

Suppose the IT domain is formed by both the domain of software and communication and the corpus in communication is labeled C. Then, the lexicon set obtained from $S \cap C - G$ contains the common terms in domain of information technology, and *(S-C)-G* represents the terms in the domain of software. Verification of terms in the domain of IT and Software can be carried out respectively against the proposed lexicons. Moreover, by using documents with labels of the hierarchical categories, a lexicon tree may also be achieved [Jing-Shin Chang,2004]. The algorithm based on this lexicon sets of different domains is called the *Lexicon Set Algorithm(LSA)*.

The second algorithm, referred to as the *Document Crossing Algorithm(DCA)*, is based on term frequency and document frequency. It uses the so called TFIDF (term frequency inverse document frequency) method. Term frequency (TF) is the frequency of terms appeared in the corpus. Document frequency (DF) represents the number of documents in which the term occurs. Since a terminology is domain specific, it is more likely to occur many times in just some specific documents. So the value of TF divided by DF (thus called TFIDF), should be related to termhood closely. Based on the TDIDF formula used in information retrieval, below is the formula used for termhood estimation:

$$Termhood(w) = 1 - \frac{DF(w)}{f(w)}$$

where *w* is a term, *f(w)* refers to the frequency of *w*, *DF(w)* refers to the document frequency of *w*, and *Termhood(w)* refers to the termhood estimation value of *w*. *Termhood(w)* is a value between 0 to 1 and the larger the value is, the more likely a term is a terminology.

The last algorithm, referred to as the *Domain Relativity Algorithm(DRA)*, is based on the strength of the relation between a term and its document class. We assume that if a term belongs to a domain, more people in this domain will use it than in other domains. In other words, they would appear more frequent in the documents of certain domain/class. Consequently, the frequency of a term appearing in different documents of a particular domain would be a good measure. Based on this observation, we use the strength of the association of a term *w* with a specific domain *d*, denoted by *Association(d, w)*, as the estimation of termhood and the measure is calculated as following:

$$Association(d,w) = \frac{p(d,w) - p(d)p(w)}{p(d,w)}$$

where *p(d,w)* is the probability of a term *w* belong to domain *d*, *p(d)* and *p(w)* are the independent probabilities of domain *d* and word string *w*. It should be noted that *p(w)* equals to *DF(w)* divided by the total number of documents in the document set of the corpora and *p(d)* equals to the number of documents belong to domain d. *Association(d,w)* is a value between 0 to 1 and the larger the value is, the more likely a term is a terminology.

4. Implementation and Evaluation

Our objective is to investigate the effects of these different termhood measurement features. Using precision and recall as the performance measurement, we can find which features are more effective.

The evaluation procedure for each of the measures given in Section 3 involves six steps listed below:

1. Data Collection: the "People's Daily" (in January 1998) is used as a general corpus ***G*** and the "Semiconductor Optoelectronics" journal as the domain corpus ***SO***.
2. Word Segmentation: the segmentation tool developed at the lab in PolyU [Lu Qin, 2003] is used to segment the above corpora.
3. Counting and acquiring frequency tuples: Count all the unigram and bi-gram in the segmented corpora to get tuples <gram, frequency of gram in the domain, number of documents which contains the gram in the domain, frequency of a gram within a documents in the domain>

4. Unithood Computing and Filtering: Compute unithood of each bi-gram and filter out bi-grams unithood of which is less than a certain value.
5. Using three algorithm respectively to re-rank and filter the term lists and output lists
6. Evaluating the lists.

Unithood identification in Step 4 is the algorithm to determine whether a string of words(or segmented units) as a term candidate should qualify as a term. The unithood identification for a term *w* involves both the external measure and the internal measure. The external measurement looks into the relationship of *w* with respect to its neighbors so as to judge whether it is used independently or as a substring of some longer terms. The internal measurement looks into the relationship of *w* with respect to the substrings(or called elements as they are segmented units) within *w*. The external measure is based on two rates, the left dependent rate(*LD*) and the right dependent rate(*RD*), whose formulas are listed below

$$LD(w) = \frac{\max_{a \in A} f(aw)}{f(w)}$$

$$RD(w) = \frac{\max_{b \in B} f(wb)}{f(w)}$$

where *f(w)* is the frequency of a string *w*, *A* is the full set of all the left neighbor elements of *w*, *B* is the full set of all right neighbor elements of *w.* It can be seen that *LD(w)* and *RD(w)* are statistical measures of the degree of independence of *w* to its neighbors. The more different neighbors *w* has, the lower the values of *LD(w)* and *RD(w)*. The combined consideration of *w* to its neighbors on both sides, denoted as *IR* (independent rate), are calculated using the geometric measure of square root as shown below.

$$IR(w) = (1 - 1/f(w)) * \sqrt{(1 - LD(w)) * (1 - RD(w))}$$

It can be seen that *IR(w)* is a consolidated external measure taking into account of both the left dependent rate and the right dependent rate.

The internal measure will first compute the connective rate for the internal adjacent elements in a term candidate *w.* Suppose a *w* is formed by a series of segmented units where $w = w_1 w_2, \ldots, w_n$. the CR value of any two neighboring elements $w_i w_{i+1}$ ($i = 1$ to $n-1$) in *w*, denoted as $CR(w_i w_{i+1})$, is given as

$$CR(w_i\, w_{i+1}) = \frac{p(w_i\, w_{i+1}) - p(w_i)\, p(w_{i+1})}{p(w_i\, w_{i+1})} * \sqrt{LD(w_i\, w_{i+1}) * RD(w_i\, w_{i+1})}$$

where *p(w)* is the probability of a string *w*, $CR(w_i w_{i+1})$ represents the ratio of two elements of the string $w_i w_{i+1}$ will be connected. The $w_i w_{i+1}$ with the lowest $CR(w_i w_{i+1})$ value indicates that w should be separated between w_i and w_{i+1} if *w* should in deed be considered as two terms as $w_i w_{i+1}$ has the weakest connectivity. The break point is then determined by $CR_{Min}(w)$ given in the following formula.

$$CR_{\min}(w_1 \cdots w_n) = \min_{1 \le i \le n-1} (CR(w_i\, w_{i+1}))$$

The final formula to measure unithood, denoted as *Unithood(w)*, takes into consideration of both external measure of IR(w) and internal measure $CR_{Min}(w)$ as shown below:

$$Unithood(w) = IR(w) * CR_{\min}(w)$$

In the implementation, a experimentally determined threshold value, U_{cut} is used. Any term candidate *w* with its *Unithood(w)* value bigger than it is considered a term which will then be subjected to the termhood algorithm. The actual value of U_{cut} used in the experimental data is 0.0157.

The semiconductor optoelectronics corpus, ***SO***, has 19 documents and the People's Daily corpus, ***G***, has 3,147 documents. After Step 4, a list of 428 terms are obtained from ***SO***. Among them, 308 are correct terms. The precision of the unithood algorithm is listed in **Table 1**.

Top *n*	Precision
100	0.910
200	0.850
300	0.790
428	0.720

Table 1: Precisions of unithood algorithm

The value *n* for the first column in **Table 1** indicates the precision of the top ranked *n* terms. It is easy to understand that the smaller the *n*, the better the precision.

Among the 308 terms, 269 terms are indeed domain specific terminologies. To evaluate the performance of the three algorithms, we conducted two sets of experiments. The first set takes the output from the Unithood algorithm. As the 428 terms from the Unithood algorithm contains noise, we call it the *noisy term set*. The second set takes the 308 correct terms as the input of the three algorithms. We call this set the *correct term set*. This way, we can also see the effect of noise to the performance of terminology extraction algorithm.

Top *n*	Precision		
	LSA	DCA	DRA
100		0.75	0.58
200		0.68	0.61
300		0.647	0.627
392	0.661	0.633	0.663
428		0.621	0.621

Table 2: Precisions using noisy term set

The precisions of the results on the noisy term set are given in **Table 2**. For convenience, we use the abbreviations LSA, DCA and DRA as the shorthand for *Lexicon Set Algorithm*, *Document Crossing Algorithm* and *Domain Relativity Algorithm* respectively. To LSA, its output is a set of terms, not a ranked list with order and the size of the set in this evaluation is 392. So there are only one precision for LSA.

We can see that DCA performs better than DRA on the top 300 or less results. But on the top 392 results, its performance is worse than performance of LSA and DRA. Some items near 392 of the ranked list generated by DCA show that their rank is 0. This is caused by the low frequency of the items. This means that DCA is very sensitive to frequency statistics. However, DCA is more effective when adequate occurrences are available.

Top *n*	Precision of		
	LSA	DCA	DRA
100		0.940	0.850
200		0.865	0.870
291	0.890	0.869	0.890
308		0.873	0.873

Table 3 Precisions using the correct term set

Table 3 shows the performance of the three algorithms using the correct term set. The relative performance of these three algorithms are almost the same as those using the noisy term set. Comparing the data in the noisy set, however, we can see that every algorithm in the correct term set enjoys about 20% better performance than the corresponding algorithm in the noisy term set. This 20% increase in performance can be seen as the direct remove of the 20% noise in the data set. This shows that the performance of Unithood calculation is extreme important for terminology extraction.

From the evaluations both the noisy term set and the correct term set, we can see that the performance of DRA on the top 200 items is the worst. This is because the top 200 items get the same rank. Given that a term occurs in the corpus SO, *p(d,w)* is equal to *p(w)*. Then, the formula *Association()* for DRA can be transformed into:

Association(d,w)=1-p(d).

Since p(d) is a constant in the evaluation, the result is a constant, too. So DRA lacks the ability to distinguish the relativity when the terms only occur in one domain. And since most of the terms do not occur in the general corpus, the poor performance is reasonable. Better algorithms needs to be developed to compute the relativity between a term and the domain in the future.

By examining the three algorithms in terms of complexity, we can see that the simplest approach in LSA achieves the best precision in the evaluation. However, the relatively poor performance of DCA and DRA may well be due to the relatively small size of the domain corpus used in the experiment which are more sensitive to statistical information.

5. Conclusion

In this study, we have designed three terminology extraction algorithms. Without noisy data, the precision of the algorithms can reach upper 80% to over 90% and degrade when unithood algorithm introduces noises. The result also show that the simple lexicon set algorithm can be very effectively in filtering out terms of general use.

In the future, combinations of these algorithms can be investigated so that different features are used in the same algorithm can improve the performance. Good unithood identification algorithms should be explored to minimize the noise to termhood identification. Experiments with much large domain data should also be investigated to reduce sensitive of these algorithms.

6. Acknowledgement

The project is partially supported by CERG project PolyU 5190/04E and a Hong Kong Polytechnic University funded project 4-Z08K.

7. References

Jing-Shin Chang, (2005). Domain Specific Word Extraction from Hierarchical Web Documents: A First Step Toward Building Lexicon Trees from Web Corpora, in *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Learning*, pp. 64-71,

Chen Yirong, (2005). The Research on Automatic Chinese Term Extraction Integrated with Unithood and Domain Feature. Master Thesis in Beijing, Peking University, pp. 4-4.

Church, K. W. and Hanks, P. (1990). Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 6(1), pp. 22-29.

Cohen, J. D. (1995). Highlights: Language- and Domain-independent Automatic Indexing Terms for Abstracting, *Journal of American Soc. for Information Science*, 46(3), pp. 162-174.

Dunning, T. (1993). Accurate Method for the Statistics of Surprise and Coincidence, *Computational Linguistics,* 19(1), pp. 61-74.

E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin and C. G. Nevill-Manning. (1999). Domain-specific keyphrase Extraction, In *Proceedings of 16th International Joint Conference on Artificial Intelligence* IJCAI-99, pp. 668-673.

Firth, J. (1957). A synopsis of linguistic theory. 1930-1955. *Studies in Linguistic Analysis*, Philological Society, Oxford

Hiroshi Nakagawa and Tatsunori Mori. (2002). A simple but powerful automatic term extraction method. In *COMPUTERM-2002 Proceedings of the 2nd International Workshop on Computational Terminology*, Taipei,Taiwan, August 31, 2002. pp. 29-35.

Jing-Shin Chang. (2004). Mining Domain Specific Words from Web Documents. *Proceedings of the 4-th China-Japan Joint Conference to Promote Cooperation in Natural Language Processing (CJNLP-04)* (abstracts, invited),p. 1, CJNLP-2004, City University of Hong Kong, China, pp. 10-15.

Kageura, K. and Umino, B. (1996). Methods of automatic term recognition: A review. Terminology 3(2), pp. 259-289.

Lu Qin, S.T. Chan, R.F. Xu, T.S. Chiu, B.L. Li, S.W. Yu. (2003). "A Unicode Based Adaptive Segmentor", *2nd SIGHAN Workshop on Chinese Language Processing*, ACL 2003, pp. l64-167.

Martin, L.E. (1990). Knowledge Extraction. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 252-262.

T. Hisamitsu and Y. Niwa. (2002). A measure of term representativeness based on the number of co-occurring salient words. In *Proceedings of the 19th COLING*.

Zhifang SUI, Yirong Chen etc. (2002). The Research on the Automatic Term Extraction in the Domain of Information Science and Technology, *Proceedings of the 5th East Asia Forum of the Terminology*