Term Extraction Through Unithood And Termhood Unification

Thuy VU, Ai Ti AW, Min ZHANG
Department of Language Technology, Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{tvu, aaiti, mzhang}@i2r.a-star.edu.sg

Abstract

Term Extraction (TE) is an important component of many NLP applications. In general, terms are extracted for a given text collection based on global context and frequency analysis on words/phrases association. These extracted terms represent effectively the text content of the collection for knowledge elicitation tasks. However, they fail to dictate the local contextual information for each document effectively. In this paper, we refine the state-of-the-art *C/NC-Value* term weighting method by considering both *termhood* and *unithood* measures, and use the former extracted terms to direct the local term extraction for each document. We performed the experiments on Straits Times year 2006 corpus and evaluated our performance using Wikipedia termbank. The experiments showed that our model outperforms *C/NC-Value* method for global term extraction by 24.4% based on term ranking. The precision for local term extraction improves by 12% when compared to pure linguistic based extraction method.

1 Introduction

Terminology Extraction (TE) is a subtask of information extraction. The goal of TE is to automatically extract relevant terms from a given corpus. These extracted terms are used in a variety of NLP tasks such as information retrieval, text mining, document summarization etc. In our application scenario, we are interested in terms whose constituent words have strong collocation relations and can be translated to another language in stable single word or multi-word translation equivalents. Thus, we define "term" as a word/phrase that carries a special meaning.

A general TE consists of two steps. The first step makes use of various degrees of linguistic filtering (e.g., part-of-speech tagging, phrase chunking etc.), through which candidates of various linguistic patterns are identified (e.g. noun-noun, adjective-noun-noun combinations etc.). The second step involves the use of frequency- or statistical-based evidence measures to compute weights indicating to what degree a candidate qualifies as a terminological unit. There are many methods in literature trying to improve this second step. Some of them borrowed the metrics from Information Retrieval to evaluate how important a term is within a document or a corpus. Those metrics are Term Frequency/Inverse Document Frequency (TF/IDF), Mutual Information, T-Score, Cosine, and Information Gain. There are also other works (Nakagawa and Mori, 2002; Frantzi and Ananiadou, 1998) that introduced better method to weigh the term candidates.

Currently, the C/NC method (Frantzi and Ananiadou, 1998) is widely considered as the state-of-the-art model for TE. Although this method was first applied on English, it also performed well on other languages such as Japanese (Hideki Mima and Sophia Ananiadou, 2001), Slovene (Špela Vintar, 2004), and other domains such as medical corpus (Frantzi and Ananiadou, 1998), and computer science (E. Milios et al, 2003).

In terminology research, a term is evaluated using two types of feature: termhood[1] and unithood

[1] Termhood refers to a degree of linguistic unit. It considers a term as a linguistic unit representative for the document content.

[2](Kyo Kageura, 1996). In C/NC method, the features used to compute the term weight are based on termhood only. In this paper, we introduce a unithood feature, T-Score, to the C/NC method. Experiment results show that by incorporating T-Score into C/NC to derive a new weight, *NTCValue*, it gives a better ranking of the global terms and outperforms *C/NC* method by 24.4%.

On the other hand, C/NC method extracts term candidates using linguistic patterns and derives their weights based on distribution of terms over all documents. The extracted terms thus represent global content of the corpus, and do not represent well the contextual information for each individual document. So, we propose a method to enrich the local terms through a Term Re-Extraction Model (TREM). Experiment results show that the precision for local TE has been improved significantly, by 12% when compared to pure linguistic based extraction method.

In the following sections, we introduce the state-of-the-art method, the C/NC Value method. We then introduce our proposed methods, the *NTCValue* method on section 3, the Term Re-Extraction Model (TREM) on section 4 followed by the experiment results and conclusion.

2 The C/NC value Method

C/NC method uses a combination of linguistic and statistical information to evaluate the weight of a term. This method has two steps: candidate extraction and term weighting by C/NC value.

2.1 Term Candidate Extraction

This method uses 3 linguistic patterns to extract the term candidates:

→ (Noun+Noun);
→ (Adj|Noun)+Noun;
→ (Adj|Noun)+|((Adj|Noun)*(NounPrep)?)(Adj|Noun)*)Noun.

The term candidates are passed to the second step.

2.2 Term Weighting

2.2.1 CValue

CValue is calculated based on the frequency of term and its subterms.

$$CValue(a) = \log_2|a| \cdot \left(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b) \right)$$

Where, $f(a)$ is the frequency of term a with $|a|$ words, T_a is the set of extracted candidate terms that contain a and $P(T_a)$ is the total number of longer candidate terms that contain a. The formula $\frac{1}{P(T_a)} \sum_{b \in T_a} f(b)$ will have value 0 when T_a is empty.

2.2.2 NC Value

NCValue combines the context information of a term together with the CValue. The weight of a context word[3] b is defined by the number of terms $t(b)$ in which it appears over the total number of terms considered, n. C_a is the set of distinct context words and $f_a(b)$ is the frequency of b as context word of a.

$$weight(b) = \frac{t(b)}{n}$$

$$NValue = \sum_{b \in C_a} f_a(b) \times weight(b)$$

$$NCValue(a) = 0.8 \cdot CValue(a) + 0.2 \cdot NValue(a)$$

From the above formula, we find that *NCValue* is mainly weighted by *CValue*.It treats the term candidate as a linguistic unit and evaluates its weight based on characteristics of the termhood, i.e. frequency and context word of the term candidate. The performance can be improved if feature measuring the adhesion of words within the term is incorporated.

3 Enhancement on Global TE: the *NTCValue*

Theoretically, the C/NC method can be improved by adding unithood feature to the term weighting formula. Based on the comparison of (Evert, S and B. Krenn, 2001), we explore T-Score, a competitive metric to evaluate the association between two words, as a unithood feature.

[2] Unithood refers to a degree of strength or stability of syntagmatic combinations or collocations.

[3] All experiments in this paper use the length of context is 3.

3.1 T-Score

The T-Score is used to measure the adhesion between two words in a corpus. It is defined by the following formula (Manning and Schuetze, 1999):

$$TS(w_i, w_j) = \frac{P(w_i, w_j) - P(w_i).P(w_j)}{\sqrt{\frac{P(w_i, w_j)}{N}}}$$

Where, $P(w_i, w_j)$ is the probability of bi-gram $w_i w_j$ in the corpus, $P(w)$ is the probability of word w in the corpus, and N is the total number of words in the corpus. The adhesion is a type of unithood feature since it is used to evaluate the intrinsic strength between two words of a term.

3.2 Incorporate T-Score within C/NC value

As discussed in 2.2, the most influential feature in the C/NC method is the term frequency. Our idea here is to combine the frequency with T-Score, a unithood feature. Taking the example in Table 1, the candidates have similar rank in the output using C/NC termhood approach.

massive tidal waves
gigantic tidal waves
killer tsunami tidal waves
deadly tidal waves
huge tidal waves
giant tidal waves
tsunamis tidal waves

Table 1. Example of similar terms [4]

To give better ranking and differentiation, we introduce T-Score to measure the adhesion between the words within the term. We use the minimum T-Score of all bi-grams in term a, $\min TS(a)$, as a weighted parameter for the term besides the term frequency. For a term $a = w_1.w_2...w_n$, the $\min TS(a)$ is defined as:

$$\min TS(a) = \min\{TS(w_i, w_{i+1})\}, i = 1...(n-1)$$

Term	$\min TS(\cdot)$
massive tidal waves	4.56
gigantic tidal waves	2.44
killer tsunami tidal waves	3.99
deadly tidal waves	3.15
huge tidal waves	2.20
giant tidal waves	1.35
***tsunamis tidal* waves**	**5.06**

Table 2. Term with Minimum T-Score value

Table 2 shows the $\min TS(a)$ of the different terms in table 1. Since $\min TS(a)$ can have a negative value, we only considered those terms with $\min TS(a) > 0$ and combined it with the term frequency. We redefine *CValue* to *TCValue* by replacing $f(a)$ using $F(a)$, as follows:

$$F(a) = \begin{cases} f(a) & \text{if } \min TS(a) \le 0 \\ f(a) \times \ln(2 + \min TS(a)) & \text{if } \min TS(a) > 0 \end{cases}$$

$$TCValue(a) = \log_2 |a| \cdot \left(F(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} F(b) \right)$$

The final weight, defined as *NTCValue*, is computed using the same parameter as *NCValue*.

$$NTCValue(a) = 0.8 \cdot TCValue(a) + 0.2 \cdot NValue(a)$$

4 Enhancement on Local Terms: Term Re-Extraction Method (TREM)

The extracted term candidates are ranked globally with best global terms promoted due to their distinguishing power. However, preliminary investigation on using linguistic patterns for extracting global term candidates for identifying term candidates of each document does not perform satisfactory, as high rank global terms do not reconcile well with the local term candidates identified using the linguistic patterns. A re-extraction process is thus evolved to derive local terms of a document from global terms using the *NTCValue* of the global terms.

4.1 Local Term Candidate Extraction

A string (or term candidate) extracted based on linguistic pattern follows the maximum matching algorithm. As long as the longest string whose part-of-speech tag satisfies the linguistic pattern, it will be extracted. For this reason, some noises are extracted together with these candidates. Table 3 shows some examples of noisy term candidates.

Strait Times yesterday
THE World Cup
gross domestic product growth forecast
senior vice-president of DBS Vickers security on-line

Table 3. Examples of noisy candidates.

[4] The *italic* means a week adhesion.

Our intention here is to reduce the noise and also mine more good terms embedded within the noise by using the global terms. We favor recall over precision to get as many local terms as possible.

The examples in table 3 show the problem in detecting term candidate's boundary using linguistic patterns. The "Strait Times yesterday" is a bad term identified by linguistic patterns because all three words are tagged as "noun". The second one is caused by an error of the POS tagger. Because of capitalization, the word "THE" is being tagged wrongly as a "proper-noun" (NNP/NNPS), and not determiner (DT). Similarly, "gross domestic product growth forecast" and "senior vice-president of DBS Vickers security on-line" are complex noun-phrases that are not symbolized good terms in the document. The more expressive terms would be "gross domestic product", "DBS Vickers security", etc.

Our proposed algorithm utilizes the term weight from section 3.2 to do term re-extraction for each document through dynamic programming theory (Viterbi algorithm) to resolve the above problem.

4.2 Proposed algorithm

The algorithm for term re-extraction is outlined in **Figure 1**.

Algorithm: Term re-extraction for a document

Input: **L** ← global term list with *NTCValue*
T ← input for TREM $\mathrm{T} = w_1 w_2 ... w_n$

1: **For** $i = 2 \rightarrow n$
2: **If** $(\mathrm{T}_{1,i} = w_1 ... w_i) \in \mathrm{L}$
3: $MaxNTC(1,i) = NTC(\mathrm{T}_{1,i})$
4: **Else** $MaxNTC(1,i) = 0$
5: **End If**
6: **For** $j = 1 \rightarrow i-1$
7: **If** $(\mathbf{T}_{j+1,i} = w_{j+1} ... w_i) \in \mathbf{L}$
8: $MaxNTC(1,i) = \max \{MaxNTC(1,j) + NTC(\mathbf{T}_{j+1,i}); MaxNTC(1,i)\}$
9: **End If**
10: **End For**
11: **End For**

Output: Updated term list for a document

Figure 1. Term Re-Extraction Algorithm

Where, $\mathrm{T}_{i,j}$ is the word chain formed by the words from i to j of the term $\mathrm{T} = w_1 w_2 ... w_n$; $MaxNTC(1,i)$ is the maximum *NTCValue* value from 1 to i of the term $\mathrm{T} = w_1 w_2 ... w_n$; and $NTC(\mathrm{T}_{1,i})$ is the *NTCValue* of $\mathrm{T}_{i,j}$.

5 Experiments and Evaluations

5.1 Term Bank Collection

Term boundary is one of the main issues in terminology research. In our experiments, we consider a term based on the resources from Wikipedia. In each Wikipedia article, the editor annotated the key terminologies through the use of hyperlinks. We extracted the key terms for each article based on this markup. The entire Wikipedia contains about 1,910,974 English articles and 8,964,590 key terms. These terms are considered as Wikipedia term-bank and we use it to evaluate our performance. An extracted term is considered correct if and only if it is in the term-bank.

5.2 Corpus Collection

To evaluate the model, we use the corpus collected from Straits Times in year 2006. We separate the data into 12 months as showed in Table 4.

Month	Total articles	Total words
1	3,134	1,844,419
2	3,151	1,824,970
3	3,622	2,098,459
4	3,369	1,969,684
5	3,395	1,957,962
6	3,187	1,781,664
7	3,253	1,818,606
8	3,497	1,927,180
9	3,463	1,853,902
10	3,499	1,870,417
11	3,493	1,845,254
12	3,175	1,711,168

Table 4. Evaluation data from Straits Times.

5.3 NTCValue Evaluation

We evaluate the performance of global ranked terms using *average-precision*. A higher *average-precision* would mean that the list contains more good terms in higher rank. The average precision $\mathbf{Ave}P(.)$ of a term-list $L = \{t_1, t_2, ..., t_{|L|}\}$ with

L_c as the list of all correct terms in L $(L_c \subset L)$, is calculated by the following formula:

$$\mathbf{Ave}P(L) = \frac{1}{|L_c|} \sum_{1 \le k \le |L|} \left[r_k \times \left(\frac{1}{k} \sum_{1 \le i \le k} r_i \right) \right]$$

Where:

$$r_i = \begin{cases} 1 & t_i \in L_c \\ 0 & t_i \notin L_c \end{cases}$$

Table 5 shows the comparison result of the origin *NCValue* and our *NTCValue* on the ranking of global terms. The experiment is conducted on the data described in section 5.2. We evaluate the performance based on 8 different levels of top ranking terms.

Each cell in Table 5 contains a couple of $\mathbf{Ave}P(.)$ for *NCValue* and *NTCValue* $(NCValue / NTCValue)$ respectively. The $\mathbf{Ave}P(.)$ decreases gradually when we relax the threshold for the evaluation . The result shows that the term ranking using *NTCValue* improves the performance significantly.

Number of top high term	01	02	03	04	05	06
50	0.70/0.77	0.57/0.81	0.52/0.80	0.51/0.78	0.55/0.80	0.67/0.69
100	0.60/0.73	0.59/0.77	0.51/0.79	0.50/0.74	0.57/0.78	0.64/0.70
200	0.55/0.70	0.56/0.75	0.53/0.78	0.49/0.72	0.55/0.77	0.62/0.69
500	0.53/0.67	0.54/0.70	0.54/0.71	0.48/0.68	0.53/0.71	0.57/0.65
1000	0.51/0.62	0.52/0.66	0.52/0.66	0.47/0.64	0.51/0.65	0.53/0.60
5000	0.48/0.58	0.49/0.61	0.49/0.62	0.45/0.60	0.49/0.61	0.49/0.56
10000	0.43/0.52	0.44/0.55	0.44/0.56	0.42/0.54	0.44/0.56	0.44/0.50
All_terms	0.38/0.47	0.39/0.49	0.40/0.50	0.37/0.48	0.39/0.49	0.38/0.45
Number of top high term	07	08	09	10	11	12
50	0.67/0.67	0.65/0.70	0.49/0.65	0.62/0.71	0.65/0.76	0.63/0.86
100	0.64/0.71	0.62/0.74	0.47/0.66	0.59/0.74	0.59/0.76	0.61/0.82
200	0.65/0.72	0.59/0.75	0.48/0.68	0.55/0.72	0.56/0.73	0.58/0.77
500	0.62/0.71	0.56/0.70	0.50/0.66	0.52/0.66	0.54/0.67	0.55/0.69
1000	0.59/0.66	0.54/0.66	0.50/0.64	0.49/0.64	0.51/0.64	0.54/0.65
5000	0.54/0.60	0.51/0.62	0.49/0.60	0.46/0.61	0.48/0.60	0.51/0.61
10000	0.46/0.53	0.46/0.55	0.45/0.55	0.43/0.56	0.44/0.55	0.46/0.55
All_terms	0.40/0.47	0.40/0.50	0.40/0.50	0.38/0.49	0.38/0.48	0.39/0.48

Table 5. Performance of NTCValue with C/NC value.

Method	Without TREM		TREM+NC		TREM+NTC	
Month	Precision	No. terms	Precision	No. terms	Precision	No. terms
1	44.98	23915	50.81	34910	50.85	34998
2	44.74	23772	50.22	34527	50.33	34657
3	44.39	28772	49.58	41691	49.59	41778
4	42.89	25857	48.78	38564	48.91	38589
5	44.67	25787	50.44	38252	50.38	38347
6	46.58	23293	51.80	33574	51.91	33651
7	46.35	23638	51.31	33990	51.35	34041
8	46.50	25869	51.91	37896	51.96	37973
9	46.16	25276	51.34	36632	51.39	36731
10	45.79	24987	50.99	36082	51.05	36179
11	45.28	24661	50.43	35894	50.54	35906
12	45.67	22745	50.73	32594	50.73	32673

Table 6. Term Re-Extraction evaluation result.

5.4 TREM Evaluation

We evaluate TREM based on the term bank described in section 5.1. Let M_i be the number of extracted terms for article i, N_i be the number of extracted terms in the term bank for article i, and n is the total articles in the test corpus. The accuracy is evaluated by the following formula:

$$P = \sum_{i=1}^{n} \frac{N_i}{M_i}$$

Table 6 shows the result of TREM. From the results, we can find that the accuracy has improved significantly after the re-extraction process. On top of that, the results of TREM based on *NTCValue* is also slightly better than using *NCValue*. Moreover, the number of correct terms extracted by TREM using *NTCValue* is higher than using *NCValue*.

6 Conclusions and Future Works

We introduce a term re-extraction process (TREM) using Viterbi algorithm to augment the local TE for each document in a corpus. The results in Table 6 show that TREM improves the precision of terms in local documents and also increases the number of correct terms extracted. We also propose a method to combine the C/NC value with T-Score. The results of our method, *NTCValue*, show that the motivation to combine the termhood features used in C/NC method, with T-Score, a unithood feature, improves the term ranking result. Results on Table 6 also show that *NTCValue* gives a better result than the origin *NCValue* for TREM.

In Table 5, the average scores for "All Term" are 38.8% and 48.3% for *NCValue* and *NTCValue* respectively. Therefore, NTCValue method improves global TE by 24.4% when compared to the origin NCValue method. With the same calculation, we also conclude that TREM outperforms the linguistic pattern method by 12% (*average scores are 50.7% and 45.3% for TREM and TREM-NTC respectively*).

In the future, we will focus on improving the performance of TREM by using more features, besides the weighting score.

References

C. Manning and H. Schuetze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press Cambridge, Massachusetts.

E. Milios, Y. Zhang, B. He, L. Dong. 2003. *Automatic Term Extraction and Document Similarity in Special Text Corpora*. Proceedings of the 6th Conference of the Pacific Association for Computational Linguistics (PACLing'03), Halifax, Nova Scotia, Canada, pp. 275-284.

Evert, S. and B. Krenn. 2001. *Methods for Qualitative Evaluation of Lexical Association Measures*. Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, pages 369 – 381.

Hideki Mima, Sophia Ananiadou. 2001. *An Application and Evaluation of the C/NC-Value Approach for the Automatic Term Recognition of Multi-Word Units in Japanese*. International Journal on Terminology.

Hiroshi Nakagawa, Tatsunori Mori. 2000. *Automatic Term Recognition based on Statistics of Compound Nouns*. Terminology, Vol.6, No.2, pp.195 – 210.

Hiroshi Nakagawa, Tatsunori Mori. 2002. *A Simple but Powerful Automatic Term Extraction Method*. 2nd International Workshop on Computational Terminology, ACL.

Katerine T. Frantzi, Sophia Ananiadou, and Junichi Tsujii. 1998. *The C-Value/NC-Value Method of Automatic Recognition for Multi-word terms*. Journal on Research and Advanced Technology for Digital Libraries.

Kyo Kageura. 1996. *Methods of Automatic Term Recognition - A Review*. Terminology, 3(2): 259 – 289, 1996.

Špela Vintar. 2004. *Comparative Evaluation of C-value in the Treatment of Nested Terms*. Memura 2004 – Methodologies and Evaluation of Multiword Units in Real-World Applications. Proceedings of the International Conference on Language Resources and Evaluation 2004, pp. 54-57.