Research on the Building Method of Domain Lexicon Combining Association Rules and Improved TF*IDF

Shouning Qu, Xu-Simon

Abstract **: To improve the efficiency and accuracy of topic words extraction in information extraction and topic words classification, a new topic lexicon building method is gradually updated and finally formed by combining association rules and improved TF*IDF algorithm. The improved TF*IDF algorithm considers the affection of text length, feature item length, feature item location and identification of compound words on the topic extraction. Experiments show that the proposed method can greatly improve the efficiency and accuracy of topic words extraction, and make information extraction and text classification more effectively.**

***Index Terms*—domain lexicon, word segmentation, statistic the frequency of the word, weight, association rule**

I. Introduction

Text is one of the most important media for information recording and spreading. Most Internet information appears as text form as well. Text mining studies how to find useful information from considerable text data, It is also a process of extracting valuable knowledge that spreads in the document and is effective ,novel, useful , understood .using above knowledge to organize information The word segmentation and extraction is the foundation for building text-oriented domain lexicon and text mining. A good domain lexicon can greatly improve the efficiency and accuracy of text mining.

The building of domain lexicon addresses the problems that common lexicon efficiency decline with the growing of lexicon and technical terms can not be extracted. So it can improve the efficiency and accuracy of the topic words extraction.

The building of domain lexicon is mainly based on the topic words extraction. The domain topic lexicon is made up of text topic words extracted from considerable domain documents.

TF*IDF algorithm [3] is a widely used topic words extraction method based on statistic the frequency of the word. Its advantage is simple, efficient, easy to implement and high recall ratio. But TF*IDF algorithm is easy to affected by the text length, feature length and feature position, It does not consider the weights of feature items affected by feature items that distribute unevenly and incompletely in classes or between classes [4]; recognizes poorly on compound words and unknown words [5], [6]. Therefore, we present a building method of domain lexicon combining association rules and improved TF*IDF to solve the above problems.

Manuscript received July 26, 2010.
This work was supported in part by National Nature Science Fund under Grant 60573065 and State “863” Plan funded Project under Grant 2002AA4Z3240.

Research on the Building Method of Domain Lexicon Combining Association Rules and Improved TF*IDF
Shouning Qu , male, is with Information Network Center, University of Jinan. Jinan 250022, China (e-mail: qsn@ujn.edu.cn)
Xu-simon is with school of Information Science and Engineering, University of Jinan, Jinan 250022, China(e-mail: xu-simon@hotmail.com or xuyuyang007@sina.com)

II. The Building of Domain Lexicon

The building of domain lexicon includes several steps as follows.

1) Select background document collections from some large-scale document collections of several given domains [7] and fill them in a data sheet, select foreground document collections from part of document collections of a certain domain and fill them in another data sheet.

2) Segment foreground document collections and background document collections by using common lexicon and forward maximum matching algorithm [8], [9].

3) Extract features items and express as vectors.

4) Recognize compound words based on segmented adjacent feature items and association rules, and fill recognized compound words in common lexicon.

5) Statistic the foreground and background words based on feature items appearing in foreground documents.

6) Calculate the weight of feature items appearing in foreground document collections based on the last step.

7) Set the threshold value, select representative feature items for foreground domain lexicon.

The built domain lexicon may be meliorated by changing training parameters. The process is shown in Fig.1.

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

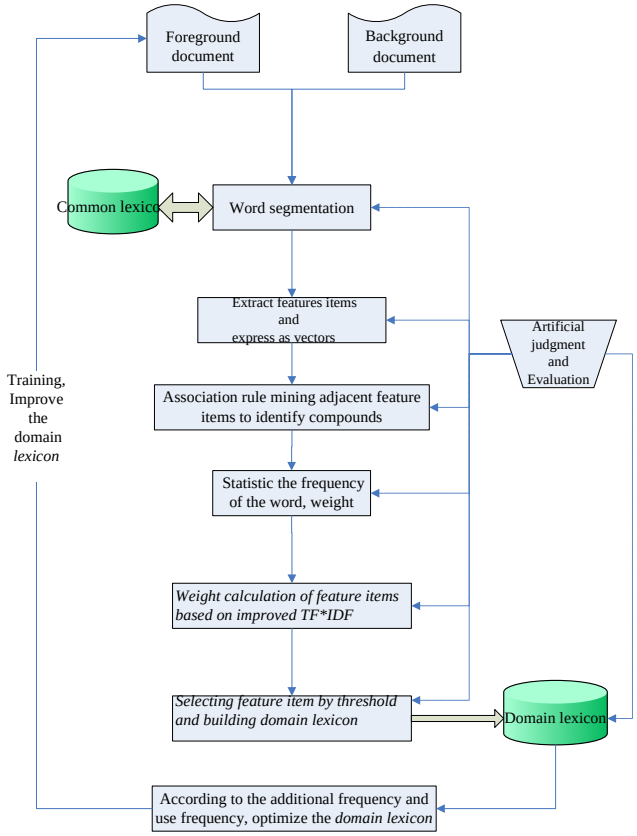


Fig.1 build domain lexicon

A. *Word segmentation*

Chinese word segmentation [10]-[12] is the basis for Chinese information processing and the necessary step for building domain lexicon. Different from alphabetic writing, Chinese processing must include the word segmentation procedure because of its intrinsic characteristics.

Because building a domain lexicon needs training a large number of document collections, so we should choose a word segmentation method of easy implement, simple, efficient method and easy maintenance. Because of its simple, fast and efficient, dictionary-based positive maximum matching algorithm is convenient for segmenting large-scale document collections, maintaining lexicon pertinently and building domain lexicon quickly. As for the weaknesses of positive maximum matching algorithm, such as too mechanical, and too simple, we can overcome them through compound words and unknown word recognition technology [13].

Based on the above analysis, we build domain lexicon using forward maximum matching algorithm for Chinese word segmentation. The basic idea of forward maximum matching algorithm is that: For a given segment Chinese word, we calculate the length of this text firstly. In this length range, we select n words from left to right, where n is determined by the longest word of the lexicon. Then we match it with words in lexicon. If the same words can be found, we segment the word from the text. Else, we reduce a word from it and match again. If there is only one word lastly and no word can match, it indicates that the word is not in the table. Then, beginning with the second word, choose n words and mach over again. If a given segment Chinese word can be matched, we begin new finding with behind word of it. Word segmentation results will be saved to a database or in memory for text mining purposes.

B. *Extraction of feature items and express as vectors*

Feature items of foreground documents usually appear several times. In order to statistic the frequency of foreground and background words and calculate the weight of each feature items, we extract features items appearing in foreground documents in advance. In this paper, we extract feature items in virtue of the vector space model [14] (Vector Space Model, VSM) which is used widely now.

In the VSM [15] model, document collections are seen as a vector space which consists of a set of orthogonal feature items. Each document is seen as one of the standardized vector, V (d) = (t1, w1 (d); ...; ti, wi(d); ... tn, wn(d)), in which ti is features items, wi(d) is the weight of ti for the document d. The features items of the vector space are equivalent to the segmented features items, may appear again and again. Accordingly, the weight of features items is calculated by the selected features items.

C. *Identification of compound words*

Submit your manuscript electronically for review. Although dictionary-based matching algorithm is simple, fast and efficient, the segmentation process is mechanical and can not match those long compound words that do not exist in dictionary. This makes some of the useful compound words segmented into a few simple words. While these compound words are of significance for understanding the text. So, it brings meaning alienation. Such as "NATO", it can be segmented into "North", "Western", "Covenant," and "organization", resulting in alienation of meaning. So, this weakness restricts the extraction of feature items and the building of domain lexicon.

In this paper, we use the compound word recognition method based on association rules [16], treat the foreground document collections as a transaction database. The feature items segmented are seen as a set of transaction items. The text transactions can be expressed as: (Document ID, t1 , t2, t3, ..., tn). Thus, the feature items association analysis is converted to association mining of transaction items in transaction database. At the same time, the issue comparability analysis is converted to association rule mining of transaction items in transaction database.

Comparing to conventional association rule mining algorithm, the keyword-based text association analysis, involves two major steps [17]: 1) Mine frequent appearing keywords, namely frequency item set. 2) According to the frequency item set, generate association rules between keywords.

According to the association rule mining algorithm, if ti and tj are adjacent with frequent co-occurrence, we can conclude that ti and tj form a compound word association rule. Similarly, if several adjacent feature items is of frequent co-occurrence, and confidence and minimum support exceed a certain threshold, the more complex association rules can be mined out, and the longer compound words can be identified. Further, the association rules will be filled in a form named association rule table as the form of (no, front, rear, S, C). Association rules table facilitate the identification of compound words, the extraction of more useful compound

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

words and the building of better domain lexicon. (No, front, rear, S, C) means (the serial number of generated association rules, the front association rules, the rear association rules, support grade, confidence).

D. Statistic the frequency of feature items

Submit your manuscript electronically for review. Word frequency statistics [18] is a kind of lexical analysis method, is the basis of text word segmentation and weight calculation of feature items. It plays an important role in natural language processing areas, for example, information retrieval, text proofreading, text classification, clustering, and so on. It describes vocabulary distributing through the statistical word frequency of some document collections and analytical results. The major steps are shown in Fig.2.

This article mainly involves two kinds of word frequency statistics, namely foreground word frequency statistics and background word frequency statistics. Foreground word frequency statistics aims at counting the appeared times of feature items in foreground documents. Background word frequency statistics counts the appeared times of feature items in background documents.

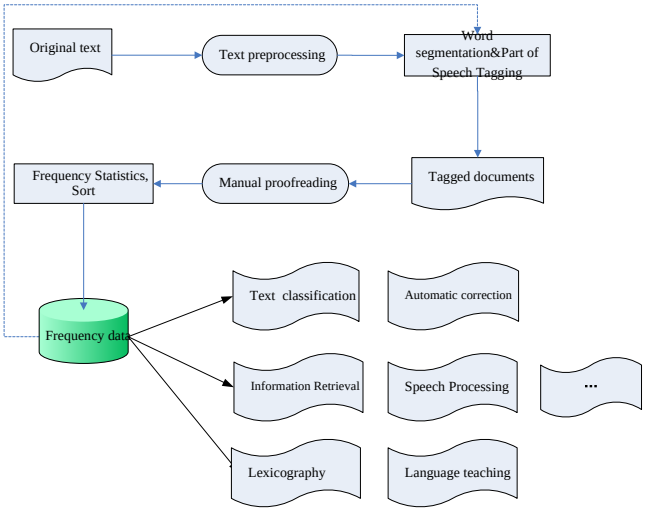


Fig.2 Basic steps of Statistic word frequency and its application

*E. Weight calculation of feature items based on improved TF*IDF*

Each feature item has a weight value which can be used to analyze domain lexicon. The weight value means the importance of the feature item for the text, which determines if the feature item should be added to the domain lexicon. This section presents an improved TF*IDF algorithm, and calculates weight using the improved TF*IDF.

The familiar weight calculation of feature items include TF algorithm [18], IDF algorithm [19], TF*IDF algorithm, and so on. TF*IDF algorithm inosculates advantages of TF algorithm and IDF algorithm, makes up for shortcomings each other. TF*IDF algorithm Combines TF and IDF, It can not only show contents of classifications, but can be distinguished from other classes. The more a feature item appears in a document or some kind documents, namely the higher the foreground word frequency of the feature items means the stronger capability of reflecting contents for the feature item. The wider a feature item appears in documents, namely a higher background word frequency of the feature item, means the lower capability of reflecting contents for the feature item. In VSM, the traditional TF*IDF weight calculation formula is:

$$w_i(d_j) = tf_{ij} \times idf_i = tf_{ij} \times \log_2(N / N_i + 0.01) \quad (1)$$

In the formula, tf_{ij} is the frequency of the i document feature item appearing in document d_j, N is the frequency of a document collection, N_i is the frequency of the i text feature item appearing in document collection.

TF*IDF can effectively weaken high frequency stop-words in most documents. Empty words, such as "的", "呢", "了", and so on, is easy to be filtered out. This reduces the step of removing stop words. It can not only show contents of classifications, but can be distinguished from other classes. Traditional TF*IDF weight calculation method is vulnerable to the length of the document, length of feature items and the location of feature items. We put forward a improved TF*IDF to address these problems.

In 1988, Salton put forward that for those feature items with the same frequency, it is of more importance appearing in short documents than in long documents. So, we standardize the length of documents; convert the foreground document and background document into the same length of 100 characters. Background word frequency and foreground word frequency can be converted using (2).

$$stf_{ij} = \frac{tf_{ij}}{d_{j.length}} \times 100 \quad (2)$$

In the formula, stf_{ij} is standardize foreground document frequency of the i feature item document. tf_{ij} is the frequency of the i feature item appearing in document j. $d_{j.length}$ is the length of the document j.

The feature item length can be seen as a weight factor. The statistical result of automatic word segmentations shows that: The single word appeared in the document is usually of the largest amount, but of less information. The multi-word is usually of the least amount, but of more information and of more importance. Usually, the longer feature term expresses some special "concept" rightly. Such as "U.S. Open" specifically refers to "sports". So a higher weight should be endowed to a multi-word. In general, feature items appeared in title, abstract or the first few lines are of more importance than that in the body, and of more topic information. So, these feature items be endowed with higher weights.

In VSM, the formula (3) is commonly used for the traditional weight calculating of text feature items in TF*IDF.

$$w_i(d_j) = tf_{ij} \times \log_2(N / N_i + 0.01) \quad (3)$$

In the formula, tf_{ij} is the frequency of the i document feature item appearing in document d_j, N is the frequency of a document collection, N_i is the frequency of the i text feature item appearing in document collection.

Considering the weight calculating effected by the text length, the feature item length and the location, we standardize foreground word frequency and make appropriate weighted processing. Finally, the improved weight calculating formula (4) is obtained by normalization

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

processing.

$$w_i(d_j)=\frac{stf_{ij}\times(wp_{ij}+wl_{ij})\log_2(N/N_i+0.01)}{\sqrt{\sum_{k=1}^{n}(stf_{kj}\times(wp_{kj}+wl_{kj}))^2\times[\log_2(N/N_k+0.01)]^2}} \quad (4)$$

In the formula, stf_{ij} (stf_{kj}) is standard foreground document frequency of the nth feature item of document j, wl_{ij}(wl_{kj}) and wp_{ij}(wp_{kj}) represent the weighted weight of the nth feature item length and its location of document j respectively. The value of wl_{ij} and wp_{ij} can take an experience value or a appropriate value by repeated training document collections.

Based on the document standardized and weighted processing approach, in this paper, we solve the feature item weight calculation problem of which the weight value is effected by the text feature item length and its position. We improve TF*IDF algorithm as well. The improved TF*IDF algorithm can calculate the feature item weight more accurately and effectively. This facilitates the building of high-quality domain lexicons.

F. Selecting feature item by threshold and building domain lexicon

Based on the weight of feature items, we analyze and sort feature items for further validating and evaluating. We select representative feature items by setting a certain threshold, and add them to domain lexicons.

The main purpose of building domain lexicon is to address the efficiency descending of common lexicons with the size increasing. Therefore, the built domain lexicons should comprise of words representing its domain in order to mining text application more rapidly and effectively, such as text classification, topic extraction, and so on. Based on a certainty number of vocabulary, the selecting threshold may be set higher as soon as possible for ensuring the correct ratio of extracted domain topic. After repeated attempts, we think that setting the weight threshold as 0.16 is appropriate. You can set other appropriate threshold according to actual situation as well. At the same time, we can set the selecting rules according to actual situation, such as eliminating out some single words of high weight. When the selected feature items can roughly represent the topic of the training document, they may be added to the domain lexicon.

III. THE DATA DESIGN OF DOMAIN LEXICON

A. Data dictionary

In this paper, Microsoft Office Access 2003 database is used to domain lexicon, involves the following 5 tables.

Common information table (words), namely the common lexicon, is used to store the selected feature items from foreground and background document collections. Its structure is shown as in Table I.

Table I structure of words

Listing	Data Type	Field size	primary key	Function Description
words_ID	Automatic ID	Long Integer	yes	lexicon item No.
words_word	Text	50	no	Storage lexicon item

Background document collections information table (backtext) is used to store or access background document collections. Its structure is shown as in Table II.

Table II structure of backtext

Listing	Data Type	Field size	primary key	Function Description
backtext_ID	Automatic ID	Long Integer	yes	lexicon item No.
backtext_sentence	Text	300	no	Storage lexicon item

Foreground document collections information table (backtext) is used to store or access foreground document collections. Its structure is shown as in Table III.

Table III structure of foretext

Listing	Data Type	Field size	primary key	Function Description
foretext_ID	Automatic ID	Long Integer	yes	lexicon item No.
foretext_sentence	Text	300	no	Storage lexicon item

Weight information table (weight) is used to store the weight information of analyzed and sorted feature items. Its structure is shown as in Table IV.

Table IV structure of weight

Listing	Data Type	Field size	primary key	Function Description
Weight_ID	Number	Long Integer	yes	lexicon item No.
Weight_word	Text	255	No	Storage lexicon item
Weight_weight	Number	Double precision	no	Stored weight value

Domain lexicon information table (domain_words) is used to store the feature items extracted from the document segmentation, word frequency, weight analysis, weight sorting and selecting. In the table, average_weight, usefrequency and addfrequency represent the change of average weight, the using frequency and the add frequency of feature items of domain lexicon respectively. They are set to optimize and the training and using processes. If their values are lower, we can remove the corresponding feature items from the domain lexicon.

Table V structure of domain_words

Listing	Data Type	Field size	primary key	Function Description
Domain_words_ID	Number	Long Integer	yes	lexicon item No.
Domain_words_word	Text	255	no	Storage lexicon item
Average_weight	Number	Double precision	no	Storage average weight of feature items
usefrequency	Number	Long Integer	no	Record use frequency of feature items
addfrequency	Number	Long Integer	no	Record add times of feature items

B. Data definition specification

In order to improve data access efficiency, we use the uniform naming specification which can effectively eliminate the redundant data in the database to meet varies user applications and requirements.

1) Data table name naming specification:

(1) Choose meaningful words in English or Pinyin, mixed

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

case letters, such as the word information table, words.

(2) If it is needed to express several words or Hanyu Pinyin, we use "_" as the separator between the words or Pinyin, such as the domain lexicon table domain_words.

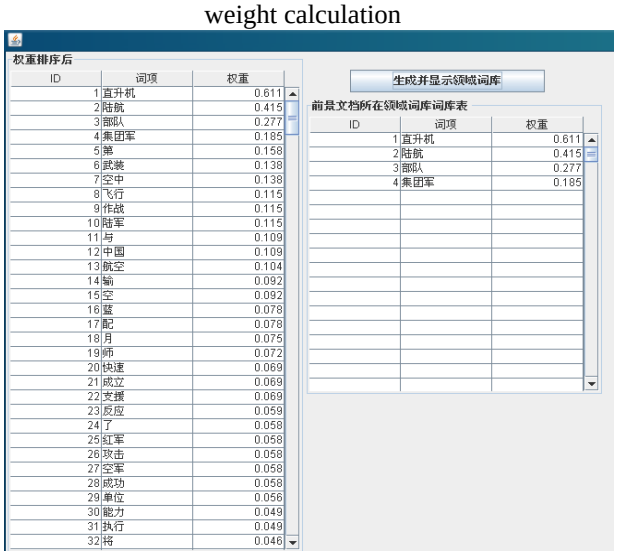
2) Data table field naming specification

Naming rules: The basic meaning adds field information. All field names spell in lower case letters or lower case Pinyin. Such as the id field of words table: words_id.

IV. Experimental Results and Analysis

Based on the planed functions of the domain lexicon and the techniques of text mining, we build the domain lexicon under the Windows environment. The OO ideas and NetBeans visualization approaches are combined with our system. Based on the classed document collections from Chinese Department of Fudan University, we select a document collections of A domain as the background document collections, select another document collections of B(different from A) domain as the foreground document collections. The domain lexicon is built by extracting the domain topic words of the foreground document. Let's take the building of military domain lexicon as an example to analysis and explain. The whole program running figure is shown as in Fig.3.

(a) Document segmentation, Statistic word frequency and weight calculation



(b) sorting and building weight domain lexicon

Fig.3 domain lexicon generation

In Fig.3, we select 100 pieces of educational documents as the background document, extract domain topic words from a military document named "中国陆军航空兵", gain 4 military words, namely "直升机", "陆航", "部队" and "集团军". This reflects the document topic in some degree. Because the "陆军航空兵" usually appears as the name "陆航", so "陆航" is extracted. Thus, we can add "陆航" to the domain lexicon artificially. In the follow study, we may further improve the domain lexicon by setting thesaurus table.

The building of the domain lexicon is implemented based on extracting topic words of domain documents. Its quality may be evaluated though the accuracy shown in (5)

$$domain\_words\_precision = \frac{topic\_NO.}{word\_NO.} \quad (5)$$

In the formula, $domain\_words\_precision$ represents the precision of the domain lexicon. $topic\_NO.$ represents the amount of the representative lemmas of the domain lexicon. $word\_NO.$ represents the sum lemmas of the domain lexicon.

Through training 76 pieces of military documents from Chinese Department of Fudan University, we gain 321 pieces of lemmas, in which 211 pieces is no repeated, and 13-16 pieces are of no obvious representation significance. The lemma accuracy rate of the military domain lexicon reaches 92.41% to 93.83%. The trained military domain lexicon is shown as in Fig.4.

Fig.4 trained military lexicon

The trained military domain lexicon is shown as in Fig.4. As can be seen from Fig.4, although "held" and "reports" are of greater weight and bigger adding times, it is eisegetical for taking them as a military vocabulary. This is mainly due to that the document is interdisciplinary and the selected document classification is incomplete. Thus, although the extracted feature item is the document's topics words, not all these topic words are correspondingly belong to the document's domain. In the following using process, based on the average weight of domain lemmas, we may optimize the domain lexicon by adding the frequency and the using frequency.

If only background documents reach to a certain account, a better extraction accuracy of feature items can be obtained. The extracted lemma account and domain lemma accuracy are shown in Table VI. These data are obtained by disposal 76 pieces military domain under given 100, 300, 500 pieces background documents.

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table VI feature items and there accuracy extracted from different background documents

Background documents numbers	100	300	500
Extraction of feature items	321	320	323
Characteristics of non-duplicate entries	211	208	206
Not representative of the term	13-16	13-16	13-16
Accuracy	0.924~0.938	0.923~0.937	0.922~0.936

As is shown in Table VI, the accuracies of 100, 300, 500 pieces background documents are of little difference. The accuracies of 300 and 500 pieces background documents are lower than that of 100 pieces. When the account of background documents are selected as 300 and 500 pieces, the program runs awfully slowly, only very few feature items appear and disappear, weight values are of some subtle changes. So, 100 pieces of representative background documents are sufficient to ensuring the accuracy and efficiency of the domain lexicon, rather than more is better.

For verifying whether the word segmentation and topic extraction efficiency can be improved by using the domain lexicon, we do experiments by using the domain lexicon as experiment lexicon. We arbitrarily select 10 pieces military documents from document collections and 100 pieces military documents from other Internet documents for testing. As the results, some test results of a training sample and the open test samples are shown in Fig.5 and Fig.6.

前景文档所在领域词库词库表

ID	词项	权重
1	直升机	0.611
2	陆航	0.415
3	部队	0.277
4	集团军	0.185

(a) Topic words extraction base on common lexicon

前景文档所在领域词库词库表

ID	词项	权重
1	直升机	0.701
2	陆航	0.476
3	部队	0.344
4	集团军	0.212

(b) Topic words extraction based on military lexicon

Fig.5 comparison of topic words extraction between common lexicon and domain lexicon based on the training document named "中国陆军航空兵"

Fig.5 is a comparison table of topic words extraction between common lexicon and domain lexicon based on the training document named "中国陆军航空兵". Fig.5 (a) is the result of topic words extraction for the document based on common lexicon. These extracted topic words reflect the document topic generally. The running time is 43 second on the local computer. Fig.5 (b) is the result of topic words extraction for the document based on the domain lexicon. The running time is 14 second on the local computer. The efficiency and weight values are improved greatly, more convenient for distinguishing. Fig.6 is a comparison table of topic words extraction between common lexicon and domain lexicon based on the open test document named "中国舰队完美行动为航母舰队问世铺垫". Fig.6a is the result of topic words extraction for the document based on common lexicon. These extracted topic words reflect the document topic generally. The running time is 41 second on the local computer. Fig.6b is the result of topic words extraction for the document based on the domain lexicon. The running time is 16 second on the local computer. The running efficiency of the domain lexicon is improved greatly for the open test document. At the same time, more meaningful topic words can be extracted.

前景文档所在领域词库词库表

ID	词项	权重
1	中国	0.435
2	海军	0.408
3	潜艇	0.282
4	航母	0.219
5	而且	0.167

(a) Topic words extraction of base on common lexicon

前景文档所在领域词库词库表

ID	词项	权重
1	中国	0.549
2	海军	0.537
3	潜艇	0.372
4	航母	0.289
5	北京	0.21
6	军事	0.165

(b) Topic words extraction of base on military lexicon

Fig.6 Topic words extraction result comparison of common lexicon and military lexicon in open test document

Fig.6 is based on the open test document "China Fleet aircraft carrier fleet to come out perfect foreshadowing action" were on the field of general vocabulary and thesaurus for keywords were extracted. Fig.6 (a) is the document for keywords based on a common lexicon extraction results, the process conditions in the plane of absolute running time 41s, the extraction of key words fundamental to the document should be the theme. Fig.6 (b) is a thesaurus of the document that based on domain keywords extracted results, the process in the plane of the absolute running time of 16s. For the open test text, the field of thesaurus is not only greatly improve operation efficiency, but also more accurate to extract more meaningful keywords.

The extraction efficiency comparison about common lexicon and domain lexicon is shown in Table VII. These experiment results are obtained at local computer through testing a large number of documents repeatedly

Table VII Topic words extraction efficiency contrast between common lexicon and domain lexicon

Type of lexicon to extract topic words	Absolute running time under local host
One based on the common lexicon	39~45s
One based on the domain lexicon	13~17s

As can be seen from Table VII, in the same conditions, the extraction time of topic words of domain lexicons is shorten greatly comparing with that of common lexicons. While the

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

topic words extraction efficiency is improved greatly.

Table VIII is a comparison table of domain lexicon accuracy for several familiar building methods. According to Table VIII, the accuracy of the built domain lexicon based on association rules and improved TF*IDF is higher than that of traditional TF*IDF and pseudo feedback model [20].

Table VIII Comparison of the accuracy of the domain lexicon

Structure type of domain lexicon	The accuracy of the domain lexicon (more than 1000 items)
Base on traditional TF * IDF algorithm's domain lexicon	83.72%~83.74%
Base on pseudo feedback model's domain lexicon	86%~88%
Base on Association Rules and Improved TF*IDF's domain lexicon	92%~94%

According to a large number of open text test, we find some problems. The words covering of the domain lexicon need to improve continue. The representation of individual words is not of strong inbeing. For some documents from distant domains or of ambiguous classification, the extracted topic words may be incomplete. In response to these problems, we need to select mass document collections of obvious characteristics and train sequentially.

V. Conclusion

By training classed Chinese document collections, we build domain lexicons. Using domain lexicons to extract topic words from given documents can greatly improve the efficiency and accuracy of words segmentation, we can build a domain lexicon by selecting some given document collections, finding out the domain topic words compared to another document collections ,recording the average weight change ,frequency of adding and using . Above all , The proposed extraction algorithm fully considers the length of the text, feature item length, feature item position, and compound words recognition and so on, improves traditional TF*IDF algorithm, identifies compound words by using association rules. Therefore, our algorithm can obtain good extraction results, and can be applied to Chinese keywords extraction as well as other aspects of text mining.

Acknowledgment

Xu-simon would like to thank Shouning Qu, and Xu-simon would like to thank WCECS2010 give me time and chance as well.

References

[1] Ronen Feldman, James Sanger, *The Text Mining Handbook* . Beijing: Posts & Telecom Press. 2009.
[2] Kodratoff Y. "Knowledge Discovery in Texts: A Definition, and Applications," Proc. ISMIS' 99, Warsaw, June 1999.
[3] Salton G, Buckley B, "Term-Weighting approaches in automatic text retrieval," Information Processing and Management, 1988,24(5):513-523.
[4] Zhongyang Xiong, Gang Li, Chen Xiaoli Chen, Wei Chen, "Improvement and application to weighting terms based on text classification," Computer Engineering and Applications, 2008,44 (5) :187-189
[5] Wei Huang, Bing GAO, Yi Liu, Kewei Yang, "Word Combination Based Chinese Word Segmentation Methodology ," Science Technology and Engineering, 2010,10 (1) :85-89.
[6] Jing DU, Hailing Xiong, "Algorithm to recognize unknown Chinese words based on BBS corpus ," Computer Engineering and Design, 2010,31 (3) :630-631.
[7] Li Juanzi, FAN Qi'na, ZHANG Kuo, "Keyword Extraction Based on tf/idf for Chinese News Document," Wuhan university Journal of Natural Sciences,2007,12(5): 917-921.
[8] Xiheng Hu, "Application of Maximum Matching Method in Chinese Segmentation Technology ," Journal of Anshan Normal University, 2008,10 (2) :42-45.
[9] Chunhui Liu, *Research on Chinese Segmentation Method Based on Optimization Maximum Matching*, Yanshan University, 2009.
[10] Hongzhi Liu, "Research on Chinese Word Segmentation Techniques," Computer Development & Applications, 2010, 23 (3):1-3.
[11] Bin Sun, Modern Chinese text word segmentation technology. Peking Institute of Computational Linguistics. Http://icl.pku.edu.cn/bswen/nlp/report1-sementation.html. 2004.
[12] Yaofeng Liu, Zhiliang Wang, Chuanjing Wang, "Model of Chinese Words Segmentation and Part-of-Word Tagging ," Computer Engineering, 2010,36 (4) :17-19.
[13] Fei Su, Danli Wang, Guozhong Dai, "A Rule-statistic Model Based on Tag and an Algorithm to Recognize Unknown Words," Computer Engineering and Applications, 2004, 15:43-45, 91.
[14] Salton G, Wang A, Yang C S, "A vector space model for automatic indexing," Communication of the ACM, 1975,18(11): 613-620.
[15] Fabrizio Sebastiani, "Machine learning in automated text categorization," ACM Computing Surveys, 2002, 34(1):1-47.
[16] Agrawal R, Srikant R, "Fast Algorithm for Mining Association Rules," In Proceeding 1994 International conference Very Large Data Bases(VLDB'94). Santiago, Chile.Sept, 1994:487-499.
[17] John K.Holt, Soon M.Chung, "Mining association rules using inverted hashing and pruning," Information Processing Letters, 83(2002):211-220.
[18] Wenhua Dai, *Research on Text Classification and Clustering based on genetic algorithms*. Beijing: Science Press, 2008.
[19] James Auen. Natural Language Understanding. Benjamin / Cummings Publishing Company, 1991.
[20] Yulan Huang, Caicun Gong, Hongbo Xu, XueQi Cheng, "A Domain Dictionary Generation Algorithm Based on Pseudo Feedback Model," Journal of Chinese Information Processing, 2008, 22 (1):111-115.