Send Orders for Reprints to reprints@benthamscience.ae

The Open Automation and Control Systems Journal, 2015, *7*, 1347-1351

F_2N-Rank: Domain Keywords Extraction Algorithm

Zhijuan Wang[1,2*] and Yinghui Feng[1]

[1]*The College of Information Engineering, Minzu University of China, Beijing, 100081, China;* [2]*Minority Languages Branch, National Language Resource Monitoring & Research Center, Beijing, 100081, China*

Abstract: Domain keywords extraction is very important for information extraction, information retrieval, classification, clustering, topic detection and tracking, and so on. TextRank is a common graph-based algorithm for keywords extraction. For TextRank, only edge weights are taken into account. We proposed a new text ranking formula that takes into account both edge and node weights, named F_2N-Rank. Experiments show that F_2N-Rank clearly outperformed both TextRank and ATF*DF. F_2N-Rank has the highest average precision (78.6%), about 16% over TextRank and 29% over ATF*DF in keywords extraction of Tibetan religion.

Keywords: ATF*DF, Domian keywords, F_2N-Rank, TextRank.

1. INTRODUCTION

Domain keywords can serve as a highly condensed summary for a domain, and they can be used as labels for a domain. Domain keywords should be ordered by the "importance" of keywords.

In the study of keywords extraction, supervised methods [1-6] always depend on the trained model and the domain it is trained on. And in unsupervised methods [7-11], algorithms based on term frequency and based on graph are the most common methods. Algorithms based on term frequency such as TF, ATF, ATF*DF, ATF*DF are easy to realize but their precisions are not very high. Algorithms based on graph, such as TextRank [7], are more effective than algorithms based on term frequency because they take into account the relationships among words.

TextRank is one of the most popular graph-based methods. Each node of the graph corresponds to a candidate keyword from the document and an edge connects two related candidates. In the entire graph, only edge weights are taken into account. Node weights are also very important. TF*IDF is the common method for measuring node weights. However, TF*IDF is less suitable than ATF*DF when measuring word weights in a document if it appears frequently in a document and rarely occurs in the others [12]. For domain keywords extraction, terms reflecting a domain should appear frequently in a large number of documents [13] and ATF (average term frequency) should be used instead of TF.

In this paper, a graph-based algorithm inspired by TextRank is proposed, named F_2N-Rank. The node weights are taken into account and the idea of F_2-measure is used for calculating node weights. F_2-measure formula gives consideration to both ATF and DF.

This paper is organized as follows: Firstly, TextRank algorithm is introduced. Secondly, the algorithm that we call F_2N-Rank is proposed for extracting domain keywords. Thirdly, some experiments are performed on the dataset of Tibetan religious domain, and the results are given. Finally, the conclusion is given.

2. TEXTRANK ALGORITHM

TextRank is inspired from PageRank. It is propose by Mihalcea R and Tarau P in 2004. According to TextRank, a text or a corpus is represented as a graph, the words of them are considered as nodes. The formula of TextRank is shown in Eq. (1).

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{v_k \in Out(V_j)} W_{jk}} WS(V_j) \quad (1)$$

- $WS(V_i)$ is the score of node V_i.
- d is the damping factor that can be set from 0 to 1, which represents the probability of jumping from a given node to another random node in the graph. The value of d is usually set to 0.85.
- w_{ji} is the weight of the edge from the previous node V_j to the current node V_i.
- $In(V_i)$ is the set of nodes that point to it (predecessors).
- $Out(V_j)$ is the set of nodes that node Vi points to (successors).
- $\sum_{v_k \in Out(V_j)} W_{jk}$ is the summation of all edge weights in the previous node V_j.
- w_{ji} is defined as the numbers that the corresponding words (V_j and V_i) co-occur within a window of maxi-

1874-4443//15

mum N words in the associated text, where N ∈[2,10].[14].

TextRank only takes edge weights into account. Node weights are also very important for node scores. There are several methods can be used for computing node weights.

2.1. TF (Term Frequency)

$$\mathrm{TF}(i,j) = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (2)$$

- TF(i,j) is the number of times the term t_i appears in the document d_j divided by the length of the document.
- $n_{i,j}$ is the number of occurrences of the word t_i in document d_j.

2.2. ATF (Average Term Frequency)

$$\mathrm{ATF}(i,j) = \frac{\sum_{|D|} \frac{n_{i,j}}{\sum_k n_{k,j}}}{|\{d : t_i \in d\}|} \quad (3)$$

- $\mathrm{ATF}(V_i)$ is the average term frequency of node V_i.
- D is a collection of N documents; |D| is the cardinality of D.
- d is a document of D.

2.3. ATF*DF

$$\mathrm{ATF}(i,j) * \mathrm{DF}(i) = \frac{\sum_{|D|} \frac{n_{i,j}}{\sum_k n_{k,j}}}{|\{d : t_i \in d\}|} * \log \frac{|\{d : t_i \in d\}|}{|D|} \quad (4)$$

- $\mathrm{DF}(V_i)$ is the document frequency of node V_i.
- ATF(i,j)*DF(i) is the product of the average term frequency and document frequency of node V_i.

In next section, a new ranking formula that takes into account both edge and node weights is proposed, named F_2N-Rank.

3. PROPOSED ALGORITHM

TextRank algorithm only focuses on the relationship among nodes, and node weights are not taken into account. Eq (5) integrates TextRank formula with the node weight ($F(V_i)$).

$$FS(V_i) = (1-d) * F(V_i) + d * F(V_i) * \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{v_k \in Out(V_j)} W_{jk}} WS(V_j) \quad (5)$$

There are several formulas can be used to calculate the value of $F(V_i)$, such as TF, ATF, ATF*DF. ATF*DF is the most suitable of the three formulas because it takes into account both term frequency and document frequency. However, the simple combination of ATF and DF does not account for their proportions. Here, the idea of F-measure is introduced for calculating $F(V_i)$. [15] The formulas are given as followings:

$$\mathrm{F}(i,j) = \frac{(1+\beta^2) * \mathrm{ATF}(i,j) * \mathrm{DF}(i)}{\beta^2 * \mathrm{ATF}(i,j) + \mathrm{DF}(i)} (\beta = 2) \quad (6)$$

$$\mathrm{ATF}(i,j) = \frac{\sum_{|D|} \frac{n_{i,j}}{\sum_k n_{k,j}}}{|\{d : t_i \in d\}|} \quad (7)$$

$$\mathrm{DF}(i) = \log \frac{|\{d : t_i \in d\}|}{|D|} \quad (8)$$

The main steps of extracting domain keywords using F_2N-Rank algorithm are as followings:

Step 1: Identify words (nouns, adjectives, and so on) that suitable for the task, and add them as nodes in the graph.

Step 2: Identify relations that connect such words, and use these relations to draw edges between nodes in the graph. Edges can be directed or undirected, weighted or unweighted.

Step 3: Calculate the weight of nodes in the graph.

Step 4: Iterate the graph-based ranking algorithm until convergence.

Step 5: Sort nodes based on their final score. Top N words are the domain keywords.

4. EXPERIMENT AND RESULTS

4.1. Experiment Description

To evaluate the proposed algorithm, Tibetan religious domain is selected. Tibetan is a universal religion nation and religious activities have been an integral part of most residents' daily life. Tibetan religious keywords are microcosms of Tibetan religious domain. Tibetan religious domain corpora come from three websites. The description of corpora is in Table **1**. The corpora can be downloaded from the religion channel of the websites.

Fig. (**1**) shows the flow chart of extracting Tibetan religious domain keywords using F_2N-Rank algorithm. The first step is domain the documents preprocessing.

Sub step 1 is preparing the domain documents dataset.

Sub step 2 is word segmentation.

Sub step 3 is removing stop words.

As they are all Chinese texts, the word segmentation and removing stop words is a must. The free Chinese word segment tool is ICTCLAS Segmenter [16].

The second step is running F_2N-Rank algorithm on the prepared dataset.

Sub step 1 is calculating word's weight using F_2-measure (ATF,DF).

Table 1. The corpora of tibetan religious domain.

Corpora	The Number of Texts	The Number of Words	The Kinds of Words
http://www.amdotibet.com	437	368562	22814
http://www.tibetculture.net	446	228651	18293
http://www.tibet.cn	1230	683814	31755
Total	2113	1281027	40722

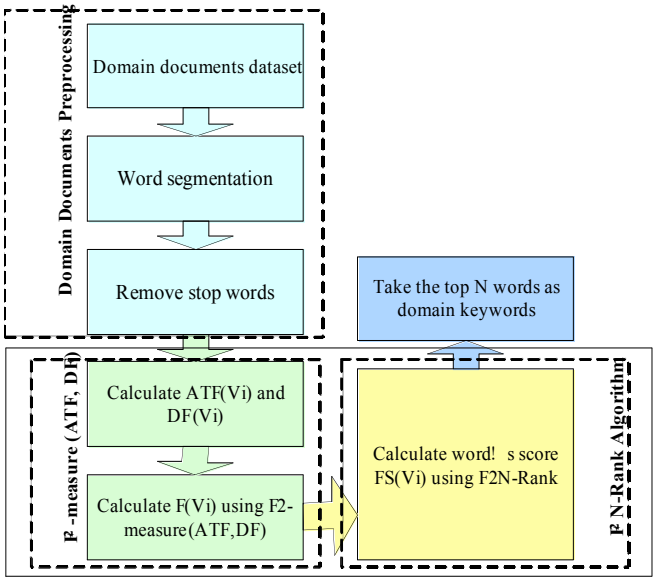

Fig. (1). The flow chart of F_2N-rank rxperiment.

Sub step 2 is calculating word's score using graph-based algorithm.

Finally, take the top N words as domain keywords.

4.2. Experiment Results

Two experiments are performed in this paper.

The aim of experiment 1 is to show which approach is best in extracting domain keywords. After comparing F_2N-Rank, TextRank and ATF*DF algorithm in precision, F_2N-Rank showed better results.

In order to show F_2N-Rank is better than $F_{0.5}$N-Rank and F_1N-Rank, namely DF-oriented is more suitable for domain keywords extraction. The experiment 2 is conducted. Experiment 2 showed that F_2N-Rank has the best performance of $F_{0.5}$N-Rank, F_1N-Rank and F_2N-Rank.

Experiment 1:

To evaluate the performance of ranking Tibetan religious keywords, we conducted a performance measurement using precision. Now, we discuss the evaluation of three different ranking algorithms. We compared algorithms which are: F_2N-Rank, TextRank and ATF*DF.

Results are shown in Fig. (**2**) by measuring the precision for top N keywords.

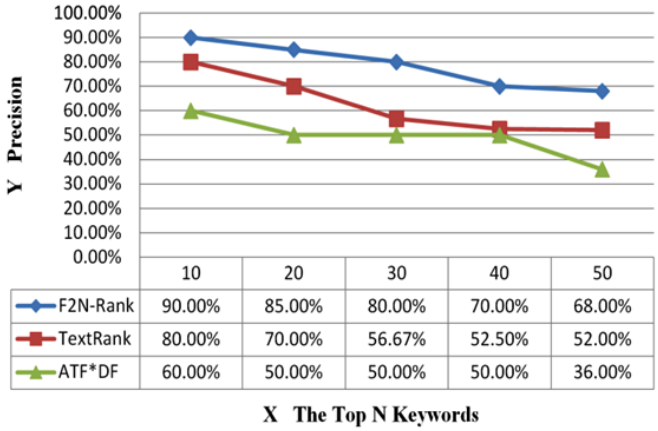

Fig. (2). Algorithm comparison in precision.

We can see that F_2N-Rank clearly outperformed both TextRank and ATF*DF. For F_2N-Rank, TextRank and ATF*DF, the average precision are 78.6%, 62.2% and 49.2%. The improvement over TextRank is around 16% in average precision and 29% over ATF*DF. Using F_2N-Rank for domain keywords extraction has showed better results.

Table **2** shows the top 20 keywords of Tibetan religious domain using F_2N-Rank Algorithm.

Experiment 2:

The order of domain keywords is also very important because it can reflect features of domain keywords. In order to illustrate the keywords extracted using F_2N-Rank are more distinguishing, experiments are conducted when taking β as 0.5,1 and 2. F_2N-Rank comes from F_βN-Rank when taking β as 2. F_2N-Rank is more DF-oriented.

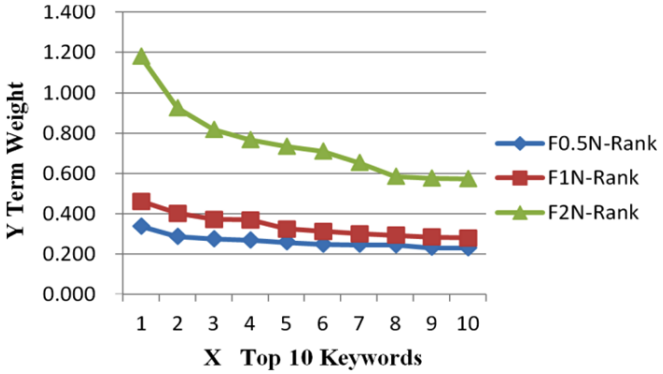

Fig. (3). Term weight of the top ten keywords for F_βN-rank algorithm (β=0.5, 1, 2).

Table 2. Top 20 tibetan religious keywords using F_2N-rank algorithm.

No.	Keywords	Meaning
1	Zangchuan Fojiao	Tibetan Buddhism
2	Banchan	a honorific title for monk
3	Zongjiao	religion
4	Huofo	Living Buddha
5	Renboqie	Rinpoche a title of respect for a master of Tibetan Buddhism
6	Gexi	a religious degree of Gelug school of Tibetan Buddhism
7	Xizang	Tibet
8	Gelupai	one of the four sects of Tibetan Buddhism
9	Sajiasi	a temple of Sakyapas which is a faction of Tibetan Buddhism)
10	Lama	a title given to a spiritual leader in Tibetan Buddhism
11	Tiaoshen	a kind of religious dance used to express stories of Gods and ghosts
12	Lasa	the capital of the Tibetan Autonomous Region
13	Larangba	the highest degree of Geshe
14	Benjiao	a Tibetan Religion
15	Dazhaosi	a Tibetan Buddhism temple
16	Shaifo	a Tibetan traditional festival
17	Zhuanshi	the belief that after somebody's death their soul lives again in a new body
18	Taersi	a Tibetan Buddhism temple
19	Dashi	a title of a highly respected monk
20	Huodong	activity

Fig. (**3**) shows term weights of the top ten keywords in Tibetan religious domain using F_βN-Rank algorithm when taking β as 0.5, 1 and 2. In F_2N-Rank, Y decreases significantly with increasing X of the three liners. Keywords in F_2N-Rank are more distinguishing. F_2N-Rank has the best performance of $F_{0.5}$N-Rank, F_1N-Rank and F_2N-Rank. Fig. (**4**) shows the precision of F_2N-Rank is the highest of $F_{0.5}$N-Rank, F_1N-Rank and F_2N-Rank.

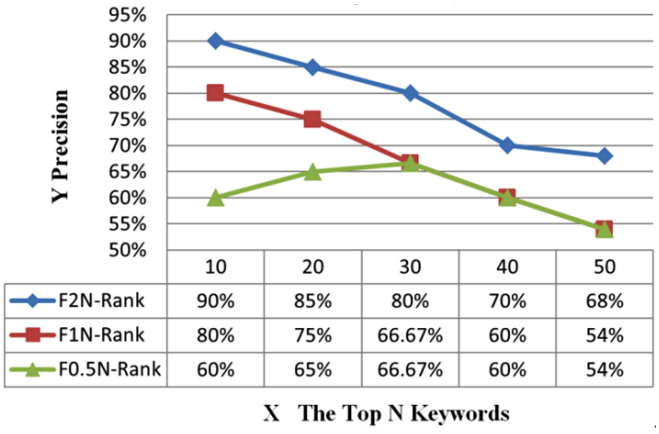

Fig. (4). F_βN-rank algorithm comparison in precision (β=0.5, 1, 2).

CONCLUSION

Domain Keywords extraction is important for many applications of Natural Language Processing. They not only relate to the term frequency, but also relate to the relationship of words. In this paper, F_2N-Rank algorithm inspired by TextRank is proposed for extracting domain keywords. In F_2N-Rank, word weights are taken into account and F_2-measure (ATF, DF) is adopted to calculate word weights. Experiments show that F_2N-Rank has the highest average precision (78.6%), about 16% over TextRank and 29% over ATF*DF. F_2N-Rank clearly outperformed both TextRank and ATF*DF. The method is generic, in the sense it can be applied to extract keywords in different domains.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

The project was supported by by Key Program of National Natural Science Foundation of China (Grant No. 61331013), National Language Committee of China (Grant No. WT125-46 and WT125-11), and also supported by

Graduate Students Projects of Minority Languages Branch, National Language Resource Monitoring & Research Center (Grant No. CML15A02), respectively.

REFERENCES

[1] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C.G. Nevill-Manning, "Domain-specific keyphrase extraction", In: *Proceedings of Ijcai,* Stockholm, Sweden, 1999, pp. 668-673.

[2] O. Medelyan, E. Frank, and I.H. Witten, "Human-competitive tagging using automatic keyphrase extraction", In: *Empirical Methods in Natural Language Processing,* Singapore, 2009, pp. 1318-1327.

[3] T. Tomokiyo, and H. Matthew, "A language model approach to key phrase extraction", In: *ACL Workshop on Multiword Expressions,* Sapporo, 2003.

[4] T. Peter, "Learning algorithms for key phrase extraction", *Information Retrieval*, vol. 2, pp. 303-336, 2000.

[5] P.D. Turney, "Coherent key phrase extraction *via* web mining", In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence,* Sapporo, Japan, 2003, pp. 434-439.

[6] T. Tomokiyo, and M. Hurst, "A language model approach to key phrase extraction", In: *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment,* Sapporo, Japan, 2003, pp. 33-40.

[7] R. Mihalcea, and P. Tarau, "TextRank, Bringing order into texts", In: *Proceedings of EMNLP 2004,* Barcelona, Spain, 2004, pp. 404-411.

[8] H. Zha, "Generic summarization and key phrase extraction using mutual reinforcement principle and sentence clustering", In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland*, pp. 113-120, 2002.

[9] X. Wan, J. Yang, and J. Xiao, "Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction", *Annual Meeting-Association for Computational Linguistics*, vol. 45, pp. 552, 2007.

[10] X. Wan, and J. Xiao, "Single document key phrase extraction using neighborhood knowledge", *AAAI*, vol. 8, pp. 855-860, 2008.

[11] F. Liu, D. Pennell, F. Liu, and Y. Liu, "Unsupervised approaches for automatic keyword extraction using meeting transcripts", In: *2009 Annual Conference of the North American chapter of the Association for Computational Linguistics,* Boulder, Colorado, 2009, pp. 620-628.

[12] F. Sebastiani, "Machine learning in automated text categorization", *ACM Computing Surveys (CSUR)*, vol. 34, pp. 1-47, 2002.

[13] Y. Gao, J. Liu, and P. X. Ma, "The hot keyphrase extraction based on tf*pdf", In: *Trust, Security and Privacy in Computing and Communications (TrustCom),* Liverpool, UK, 2011, pp. 1524-1528.

[14] K. S. Hasan, and V. Ng, "Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art", In: *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 2010, pp. 365-373.

[15] Y. Sasaki, "The truth of the F-measure", *Teach Tutor Mater*, 2007, pp. 1-5.

[16] Information on http://ictclas.org

Received: June 10, 2015 Revised: July 29, 2015 Accepted: August 15, 2015