Zhang, Hebei North University, China . Bionic intelligent optimization algorithm based on MMAS and fishswarm algorithm, TELKOMNIKA Indonesian Journal of Electrical Engineering. 2013, Vol 11 No 9, pp.5517-5522.
16. Liwei Tian, Shenyang University, China; Lin Tian, Liaoning Information integration technology engineering research center of internet of things, China.
17. Yan Aijun, Chai Tianyou, Wu Fenghua, Wang Pu, Hybrid intelligent control of combustion process for ore-roasting furnace. J Control Theory Appl 2008, 6(1) 80–85.
18. Liu Fuchun, Yao Yu, He Fenghua, Chen Songlin, Stability analysis of networked control systems with time-varying sampling periods.. J Control Theory Appl 2008, 6(1):22–25

F2N-Rank: Domain Keywords Extraction Algorithm

Zhijuan Wang*, Yinghui Feng

The College of Information Engineering, Minzu University of China,
No.27 South Street, Zhongguancun, Haidian District, Beijing, 100081, China
Minority Languages Branch, National Language Resource Monitoring & Research Center,
Beijing, No.27 South Street, Zhongguancun, Haidian District, 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 of words, named F2N-Rank. Experiments show that F2N-Rank clearly outperformed both TextRank and ATF*DF. F2N-Rank has the highest average precision (78.6%), about 16% over TextRank and 29% over ATF*DF in keywords extraction of Tibetan religion.
Keywords: F2N-RANK, TEXTRANK, ATF*DF

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 [2-7] always depend on the trained model and the domain it is trained on. And in unsupervised methods [1, 8-11], 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 [1], are more effective than

algorithms based on term frequency for 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 domain. TF*IDF function gives a higher weight to a term 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

Graph-based ranking algorithms are an essential way of deciding the importance of a node within a graph, based on global information recursively drawn from the entire graph [1]. The basic idea is to build a graph from the input document and rank its nodes according to their scores. A text is represented by a graph G (V, E). Each node (Vi) corresponds to a word. The goal is to compute the score of each node according to the formula. The score for Vi, WS(Vi), is initialized with a default value . Using (1), WS(Vi) is computed in an iterative manner until convergence. The final values are not affected by the choice of the initial value; only the number of iterations to convergence may be different [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 between 0 and 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.

wji is the weight of the edge from the previous node V_j to the current node V_i.

In(Vi) is the set of nodes that point to it (predecessors).

Out(Vj) is the set of nodes that node Vi points to (successors).[1]

$\sum_{V_k \in Out(V_j)} w_{jk}$ is the summation of all edge weights in the previous node V_j.

wji is defined as the numbers that the corresponding words V_j (and V_i) co-occur within a window of maximum 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.

(a) TF

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

TF(Vi) is the term frequency of node Vi.

(b) ATF

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

ATF(Vi) is the average term frequency of node Vi.

(c) ATF*DF

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

$DF(V_i)$ is the document frequency of node V_i.

$ATF^*DF(V_i)$ is the product of the average term frequency and document frequency of node V_i.

In Equation (2), Equation (3) and Equation (4), $n_{i,j}$ is the number of occurrences of the word t_i in document d_j. D is a collection of N documents; |D| is the cardinality of D.

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. Equation (5) integrates TextRank formula with the node weight $F(V_i)$().

$$FS(V_i) = (1-d)^*\mathbf{F}(\mathbf{V_i}) + d^*\mathbf{F}(\mathbf{V_i})^* \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} FS(V_j) \quad (5)$$

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:

$$F(V_i) = \frac{(1+\beta^2) * ATF(V_i) * DF(V_i)}{\beta^2 * ATF(V_i) + DF(V_i)} \quad (\beta = 2) \tag{6}$$

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

$$DF(V_i) = \log \frac{|\{d : t_i \in d\}|}{|D|} \tag{8}$$

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

Step1: Identify words (nouns, adjectives, and so on) that suitable for the task, and add them as nodes in the graph.

Step2: 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.

Step3: Calculate the weight of nodes in the graph.

Step4: Iterate the graph-based ranking algorithm until convergence.

Step5: Sort nodes based on their final score. Top T 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.

Table 1. The Corpora of Tibetan Religious Domain

Corpora	The numbers 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 shows the flow chat of extracting Tibetan religious domain keywords using F_2N-Rank algorithm. The first step is domain the documents processing.

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).

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

Finally, take the top T words as domain keywords.

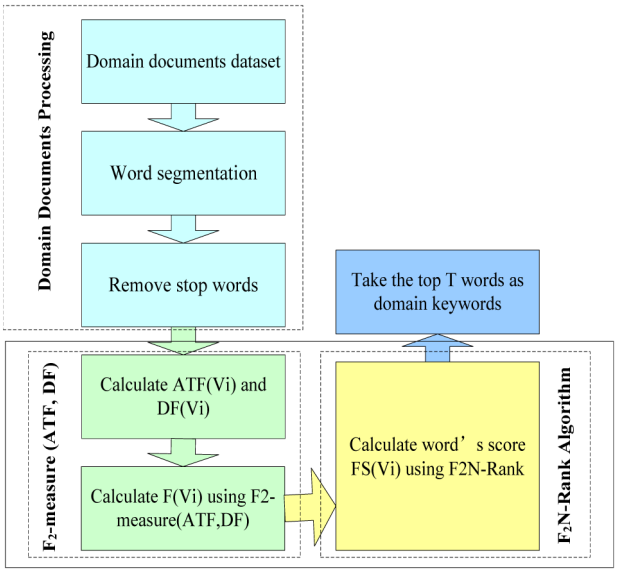

Figure 1. The flow chat of F_2N-Rank experiment

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 F1N-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.

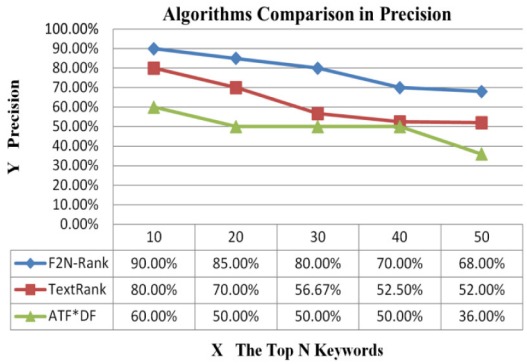

Figure 2. Algorithm Comparison in Precision

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 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.

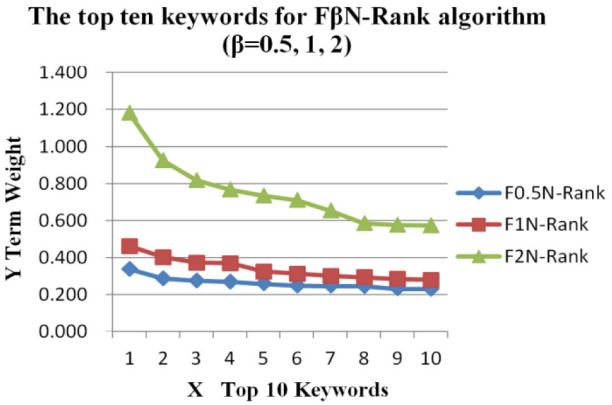

Figure (3). Term weight of the top ten keywords for F_βN-Rank algorithm (β=0.5, 1, 2)

Table 2. Top 20 Tibetan Religious Keywords Using F_2N-Rank Algorithm

No.	Keywords	Meaning
1	藏传佛教	Tibetan Buddhism
2	班禅	Panchen (a honorific title for monk)
3	宗教	religion
4	活佛	Living Buddha
5	仁波切	Rinpoche(a title of respect for a master of Tibetan Buddhism)
6	格西	Geshe(a religious degree of Gelug school of Tibetan Buddhism)
7	西藏	Tibet
8	格鲁派	Gelugpa(one of the four sects of Tibetan Buddhism)
9	萨迦寺	Sajia Temple(a temple of Sakyapas which is a faction of Tibetan Buddhism)
10	喇嘛	Lama(a title given to a spiritual leader in Tibetan Buddhism)
11	跳神	Tiaoshen(a kind of religious dance used to express stories of Gods and ghosts)
12	拉萨	Lhasa(capital of the Tibetan Autonomous Region)
13	拉让巴	lha-rams-pa(the highest degree of Geshe)
14	苯教	Bonismo(Tibetan Religion)
15	大昭寺	Dazhao Temple(a Tibetan Buddhism temple)
16	晒佛	Sun Buddha(a Tibetan traditional festival)
17	转世	Reincarnation(the belief that after somebody's death their soul lives again in a new body)
18	塔尔寺	Tar Temple(a Tibetan Buddhism temple)
19	大师	great master(a title of a highly respected monk)
20	活动	activity

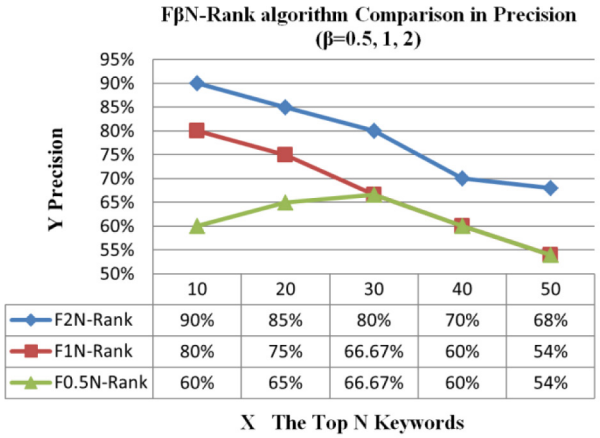

Figure (4). F_βN-Rank algorithm comparison in precision (β=0.5, 1, 2)

Conclusions

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.

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

References

1. Mihalcea R, Tarau P, “TextRank, Bringing order into texts”, Association for Computational Linguistics, 2004.
2. Frank, Eibe, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning, “Domain-specific keyphrase extraction”, In Proceedings of the 16th International Joint Conference on Artificial Intelligence, 1999, pp. 668–673.
3. Medelyan, Olena, Eibe Frank, and Ian H. Witten, “Human-competitive tagging using automatic keyphrase extraction”, In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009, pp. 1318–1327.
4. Tomokiyo, Takashi and Matthew Hurst, “A language model approach to key phrase extraction”, In Proceedings of the ACL Workshop on Multiword Expressions, 2003.
5. Turney, Peter, “Learning algorithms for key phrase extraction. Information Retrieval”, 2000, Vol.2, pp.303–336.
6. Turney, Peter, “Coherent key phrase extraction via web mining”, In Proceedings of the 18th International Joint Conference on Artificial Intelligence, 2003, pp.434–439.
7. Tomokiyo T, Hurst M, “A language model approach to key phrase extraction”, Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment, Association for Computational Linguistics, 2003, Vol.18, pp.33-40.
8. Zha H, “Generic summarization and key phrase extraction using mutual reinforcement principle and sentence clustering”, Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002, pp. 113-120.
9. Wan X, Yang J, Xiao J. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction[C]//Annual Meeting-Association for Computational Linguistics. 2007, 45(1): 552.
10. Wan X, Xiao J, “Single Document Key phrase Extraction Using Neighborhood Knowledge”, AAAI, 2008, Vol. 8, pp. 855-860.
11. Liu F, Pennell D, Liu F, et al, “Unsupervised approaches for automatic keyword extraction using meeting transcripts”, The 2009 annual conference of the North American chapter of the association for computational linguistics. Association for Computational Linguistics, 2009, pp.620-628.
12. Sebastiani F, “Machine learning in automated text categorization”, ACM computing surveys (CSUR), 2002, Vol. 31, pp.1-47.
13. Gao Y, Liu J, Ma P X, “The hot key phrase extraction based on tf*pdf”, Trust, Security and Privacy in Computing and Communications (TrustCom), 2011, pp. 1524-1528.
14. Hasan K S, Ng V, “Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art”, Proceedings of the 23rd

International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010, pp. 365-373.
15. Sasaki Y, “The truth of the F-measure”, Teach Tutor mater, 2007, pp. 1-5.
16. Information on http://ictclas.org

Point Cloud Data Simplification Using Movable Mesh Generation

Huang Ming *, Yang Fang, Zhang Jianguang, Wang Yanmin

Beijing University of Civil Engineering Architecture Key Laboratory for Urban Geomatics of National Administration of Surveying, 100044, Beijing, 100044, China
Engineering Research Center of Representative Building and Architectural Heritage Database, the Ministry of Education, Mapping and Geoinformation, 100044, Beijing, 100044, China

Abstract
In order to realize massive point cloud data simplification, a new movable mesh generation algorithm was proposed. Firstly, point cloud model was divided into a number of spatial grids. Secondly, the mesh generation was made like the first step again but more finely and with respecting of the distance threshold, and then, each appropriate point of the grids generated by the second step was filtered out according to the weight values. At last, the position of the minimum point of point cloud model's bounding box was moved, next do the mesh generation again and filter out the final points. Experimental results indicate that the proposed simplification method is able to eliminate redundant data effectively. The reduction performance of this method is superior to the distance simplification method of Geomagic Studio obviously under the same required accuracy. It can be used in the real-time data acquisition process of 3D Reconstruction.
Keywords: REVERSE ENGINEERING, POINT CLOUD, SIMPLIFICATION, MESH GENERATION, MOVABLE

1. Introduction

In recent years,the application of 3D laser technology has been become more and more widely in the field of Surveying and Mapping.However,with the characteristics of large redundancy,existence of errors and weak rules,if point cloud data acquired by scanning are disposed directly,it will be a major expenditure of time and resources.So some preprocessing works should be underway before the subsequent processing of point cloud data,includes point cloud denoising,point cloud simplification and segmentation of point clouds, etc. Point cloud simplification is not only the most basic and important step, but also a hotspot in the field of Reverse Engineering.

The current methods are mainly concentrated in the typical methods as follows: bounding box method, the simplification method based on geometric image, the simplification method based on curvature and