Research on Keyword Extraction Based on Word2Vec Weighted TextRank

Yujun Wen
School of Computer Science
Communication University of China
Beijing, China
e-mail: wenyujun@cuc.edu.cn

Hui Yuan
New Media Institute
Communication University of China
Beijing, China
e-mail: yuanhui@cuc.edu.cn

Pengzhou Zhang
Faculty of Science and Technology
Communication University of China
Beijing, China
e-mail: zhangpengzhou@cuc.edu.cn

***Abstract*—In this paper, we do a research on the keyword extraction method of news articles. We build a candidate keywords graph model based on the basic idea of TextRank, use Word2Vec to calculate the similarity between words as transition probability of nodes' weight, calculate the word score by iterative method and pick the top N of the candidate keywords as the final results. Experimental results show that the weighted TextRank algorithm with correlation of words can improve performance of keyword extraction generally.**

Keywords-keyword extraction; TextRank; word graph model; weight; Word2Vec

I. INTRODUCTION

With the rapid development of computer and Internet, there is more and more accumulation of document data. People gradually enter from the age of lack of information into the age of information overload: too much information makes users can't find what they want from the Internet. In the face of huge data, how to dig out the useful information and help users find the content they really need has been an important problem. To solve the problem, keyword extraction is a good solution.

Keyword is the smallest unit to express the core meaning of a document. Several keywords will be selected on the news, academic or social label to summary document theme content, helping users quickly understand the full articles and estimate whether the document is for their interested or necessary to read. So keyword extraction plays an important role in the field of text data mining. In practice, artificial keyword extraction is time-consuming and the results can't meet the different needs of different users, so automatic extraction is of great significance, which has drawn much attention in recent years.

II. RELATED WORK

A. Classification of Keyword Extraction Method

According to if it is necessary to label training corpus artificially, keyword extraction method can be divided into two categories[1]: supervised keyword extraction and unsupervised keyword extraction, which is detailedly showed in figure 1.

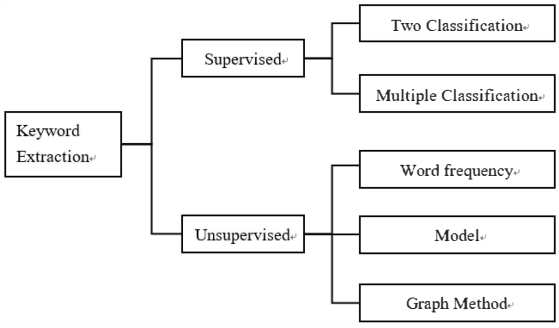

Figure 1. Classification of keyword extraction method.

Typical supervised extraction methods are classified as two or multiple classification problem. The former judges whether a candidate word can be a keyword or not, while the latter treats controlled vocabularies as the set of candidate keywords [1]. This method could make use of the existing information for maximum. Although it can obtain good effect, we need to manually label training data beforehand, which is laborious and does not apply to Internet age.

Unsupervised methods are without human intervention, which extracts keywords directly through the information of the text, improving the efficiency greatly. Thus, it becomes the main direction of current research. Mainstream unsupervised methods can be summarized as three kinds [2]: keyword extraction based on the statistical characteristics of TF-IDF, keyword extraction model based on the theme of keywords and keyword extraction based on word graph model.

Among them, the word graph model treats the document as a network composed of words, based on the theory of PageRank [3] link analysis to iterative calculation of the importance of words, which doesn't need training data set and can only use the information of document itself to

978-1-4673-9026-2/16/$31.00 ©2016 IEEE

calculate significance of candidate keywords and realize keyword extraction. On the basis of these advantages, it becomes the mainstream method of current unsupervised keyword extraction method [2].

TextRank method is typical of word graph model method. Because of its good extraction effect, TextRank has been widely researched in academic. So in the paper, we study the algorithm based on TextRank as the basis of word graph calculation.

B. *Keyword Extraction Algorithms Based on TextRank Word Graph*

TextRank [4] algorithm is derived from the classical PageRank algorithm [5]. PageRank is a famous algorithm from Google, which used to measure the value of particular web pages. Its main idea is: if there is a large number of web pages link to a page or some important pages link to the page, the page will has a high value. The score of a page achieves from all the pages with connection. Through iteration to calculate and finally getting the page's importance score.

When PageRank algorithm introduced into the field of natural language processing, TextRank was born, which is widely applied inkeywords extraction. It divides article into several component units, and chooses important components to build graph model. A set of inbound vertexes, means the vote of the supporters. The higher the number and the higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Through voting or recommend mechanism between the vertexes can realize keyword extraction without direction [6].Compared with TF-IDF which only considers word frequency itself, TextRank also considers the semantic relations between words in the document.

At present, TextRank has been extensive researched in academic, and a brief introduction is as the follow:

Yang Jie et al. (2008) proposed a multiple document keywords extraction method based on TextRank [7].The method used ATF*PDF to calculate words' weight for selecting candidates, extracted content words of weighting for candidate keywords, constructed TextRank model according to the semantic relation between candidates, iterated calculation until convergence, extracted keywords after sequencing at last. This method considered word frequency, part of speech and the semantic relation between words. The result of the experiment showed that this method could effectively extract multiple document keywords. Compared with tag clustering method based on keyword, its precision was increased by 4.2%, recall rate was increased by 7%, and F-measure was increased by 4.6%.

Li Peng et al. (2012) proposed a web page keyword extraction method based on the Tag-TextRank [8].The method is on the basis of TextRank. By introducing related documents to get the important degree of edge weight through each Tag word the target document, Tag-TextRank estimated the weight of edge, calculated the score, and synthesized weight calculation results of different Tag finally. Experimental results of public corpus showed that the Tag-TextRank was superior to the classic TextRank keyword extraction method on each evaluation index and had good applicability.

Xia Tian (2013) proposed a keyword extraction algorithm based on weighted TextRank by words position [9]. The method treated keyword extraction as word(which made up of the document) importance scheduling. It based on the basic idea of TextRank, built candidate keywords graph model, indicated and incorporated three covers influence to build probability transfer matrix: the influence of coverage, the influence of location and the influence of frequency. Experimental result showed that the weighted TextRank method of words position was superior to the traditional TextRank method and keyword extraction method based on LDA.

Gu Yijun and Xia Tian (2014) proposed a keyword extraction algorithm fusedof LDA and TextRank [2]. The method improved TextRank algorithm by processing topic-modeling on documents and computing subject influence of candidate keywords through LDA. In the paper, the method transmitted significance of candidate keywords through theme influence and adjacency relation in a non-uniform way. The study found that the effect of keyword extraction was closely associated with the theme of training data distribution. When data set presented strong subject distribution, the method could significantly improve the extraction effect, but it needed expensive document theme analysis. In the experiment, it found out that the keywords were not only associated with the documents themselves, but also closely related with the document set they belong to (that meant the document collection with similar theme).

Duan Zhun and Liu Gongshen (2015) proposed a user-template construction method based on TextRank [6].the texts user interested in were preprocessed and the meaning of each word was determined. Then, clustering and building TextRank model with the unit of category respectively. Next, introducing three influence factors to improve the probability transfer matrix: the influence factor of similarity, the influence factor of co-occurrence and the influence factor of weight of category. Finally, getting the initial user template after iteration and selecting the most critical items. Experimental results showed that the algorithm could get the initial user template accurately and could achieve a better recommendation result.

C. *Introduction of Word2Vec*

In 2013, Google opened up the open source language modeling tools Word2Vec [11], which could learn syntactic and semantic information from a large number of unmarked data.Word2Vecuses the ideas of deep learning, maps each word into a word vector of K-dimension through training, and calculates similarity vector space between word vector (usually by cosine distance) to represent semantic similarity of the text [12] [13].The method has received wide attention in the field of natural language processing.

Word2Vec makes use of two training model to train term vectors quickly and efficiently: continuous bag-of-word model (CBOW) [14], [15] and skip-gram model. Among them, the CBOW model predicts the current words from context, while skip-gram model predicts the context through

the word[16].Word2Vec treats text set as input, and generates the corresponding words vector quickly and efficiently through the training [17].Because of the word vector capturing semantic characteristics between words in a natural language, word vector can be used as the feature of words to calculate the similarity between two words, which can be applied in clustering, finding synonyms, and part of speech analysis, etc.

III. BUILD WORD GRAPH MODEL: WEIGHT AND EXTRACTION

In this paper, similarity and co-occurrence frequency between words are used as the weight for keyword extraction. Maintaining uniform random surfer hypothesis of the graph-based ranking algorithm PageRank, using Word2vec to calculate the close degree between the words, and combining with relationship between adjacent words to construct the probability transition matrix.

A. Word Graph Construction Based on TextRank

When original TextRank algorithm applied in the keyword extraction in a simple way (Brin and Page, 1998), it doesn't consider how to process influence strength between adjacent nodes. According to the original literature [18], the score of a vertex V_i is defined as follows:

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \quad (1)$$

Formula (1) is the recursive formula of TextRank, whose basic idea is a node's importance depends on how many neighboring nodes points to item. For a given vertexV_i, let $In(V_i)$ be the set of vertices that point to it (predecessors), and let $Out(V_j)$be the set of vertices that vertex $V_i$points to (successors).Where d is a damping factor that can be set between 0 and 1 (d is usually set to 0.85), whose original meaning stands for the probability of a user continuing browsing the next page back at any time in the PageRank [19]. It has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. Applicable to the TextRank, it indicates the probability of each node with 1-drandom jumps to other nodes without any edge of the connection between two nodes, which ensures that the algorithm can converge in any graph.

Then, the "strength" of the connection between two vertices is incorporated into the model, which defined as a weight w_{ij} added to the corresponding edge. The score of a vertex $V_i$with weight is defined as follows:

$$WS(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \quad (2)$$

In the paper, a research was done on how to calculate the weight w_{ij} to improve the performance of the algorithm.

B. Weight Assignment by Word2Vec and the Number of Co-occurrence of Words

Once the weight connects the two vertices is established, it will no longer be changed.

For any two nodes, the influence of one node passes to the other through its directed edge. The weight of the edge determines the final score vertex V_j obtained from vertex V_i .The weight is mainly composed of two parts: the proportion of co-occurrence frequency of vertex V_i and vertex V_j to all nodes connected to vertexV_i in the full text and the similarity of vertex V_i and vertexV_j .A specific model is shown in figure 2.

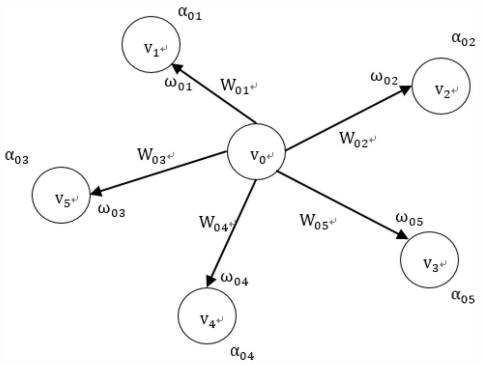

Figure 2. Example of candidate keywords weight assignment.

If the weight between the words stands for the transition probability, how to build the weight between the two words becomes the crux of keyword extraction. Word2Vec is used to calculate the distance between vertex V_i and vertex V_j , which can be treated as the similarity and close degree between two words. The score of W_{ij} is defined as the weight between vertexV_i and vertexV_j :

$$W_{ij} = (1-t) * \alpha_{ij} + t * \frac{\omega_{ij}}{\sum_{k \in Out(V_i)} \omega_{ik}} \quad (3)$$

where ω_{ij} stands for the number of co-occurrence of word i and words j, and α_{ij} stands for the similarity of vertexV_i and vertex V_j calculated by Word2Vec.In order to prevent the appearance of the condition that co-occurrence between two words is 0, which leads to the value of W is 0 and a connected graph can't form in the next part, a damping factor$t$ is added, which making the final calculation results of importance score can be convergent.

C. Keyword Extraction with Weight Assignment

Incorporating the weight between nodes, the new score is as follows:

$$WS(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{W_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \quad (4)$$

IV. EXPERIMENTAL RESULTS

This experiment adopts a set of data from Sohu News(a total of 500 articles), and each of the news has been marked

with the key words (mostly two of every news). So in the analysis of experimental results, keywords extracted can be compared with the manual annotation keywords to come to the conclusion. Precision ratio(P), recall ratio(R) and macro average value F-measure(F) is used as the judgment standard of the experimental results. The three values are calculated by the following formula:

$$P = \frac{ComputWord \cap ArticleWord}{ComputeWord} \quad (5)$$

$$R = \frac{ComputeWord \cap ArticleWord}{ArticleWord} \quad (6)$$

$$F = \frac{2*P*R}{P+R} \quad (7)$$

In the above formula, *ComputeWord* stands for the keywords calculated by the algorithm, *ArticleWord* stands for the keywords manually marked by Sohu News. In this experiment, the document is preprocessed at first, including word segmentation, removing stop words, tagging part-of-speech, finally leaving only nouns and verbs. Calculating the weight between words is used as word's relationship to other words in the full text, while keywords are extracted from the title as the result. In this paper, three, four and five is choose as the number of keyword extraction. According adjusting the regulating factor t, testing the performance of the algorithm. Experimental results are shown in the following table I II III.

TABLE I. RESULTS COMPARISON OF EXTRACTING **THREE** KEYWOODS

Algorithm	Result			
	t	*P*	*R*	*F*
TextRank	——	0.27075	0.31340	0.29052
Weighted TextRank	0.3	0.31735	0.37000	0.34166
	0.5	0.33027	0.38728	0.35651
	0.7	0.33299	0.39024	0.35935

TABLE II. RESULTS COMPARISON OF EXTRACTING **FOUR** KEYWOODS

Algorithm	Result			
	t	*P*	*R*	*F*
TextRank	——	0.25442	0.39068	0.30816
Weighted TextRank	0.3	0.30238	0.46510	0.36649
	0.5	0.30748	0.47354	0.37286
	0.7	0.30901	0.47660	0.37493

TABLE III. RESULTS COMPARISON OF EXTRACTING **FIVE** KEYWOODS

Algorithm	Result			
	t	*P*	*R*	*F*
TextRank	——	0.25656	0.48452	0.33548
Weighted TextRank	0.3	0.28575	0.53687	0.37298
	0.5	0.28616	0.53772	0.37353
	0.7	0.28575	0.53636	0.37286

From the above table can be seen, the performance of the keyword extraction algorithm based on weighted TextRank in this paper is far superior to the original TextRank. The Value of the regulating factor twill have influence on the P, R, F value of our new algorithm. But no matter what value t chooses, the new algorithm of all scores are higher than the original algorithm. What's more, while the number of keywords extracted is changed in each new, precision of weighted TextRank changes slightly, but all the results are superior to the original TextRank in the same condition.

V. CONCLUSIONS

In this paper, we introduced the basic idea of TextRank used as keyword extraction and process of constructing candidate keywords graph model, and put forward the method that made use of Word2Vec to calculate the close degree of words and considered co-occurrence frequency between two words in the text as the transition probability to calculate transition probability between the candidate keywords to build a probability transfer matrix. Then, we used it as the theme words influence to improve the iterative calculation formula of calculating the score of words. Experimental results showed that the improved weighted TextRank algorithm could improve accuracy and coverage of extracting keywords.

Next work: The experiment found that in addition to the exactly right keywords, some other keywords were mostly similar with the artificial marked keywords. However, these words were not treated as the accurately computed results properly reflected in precision, recall and F-measure value. According to this situation, we will focus on solving the problem of synonyms in the following research.

ACKNOWLEDGMENT

The work was supported by the project of National Key Technology R&D Program (2014BAK10B01-01).

REFERENCES

[1] Z. Y. Liu, Research on Keyword Extraction Using Document Topical Structure. Beijing: Tsinghua University, 2011.

[2] Y. J. Gu, T. Xia,"Research on Keyword Extraction Using LDA and TextRank. New Technology of Library and Information Service",no.7/8, 2014,pp.42-47.

[3] Page L, Brin S, Motwani R, et al,"The PageRank Citation Ranking: Bringing Order to the Web". Stanford InfoLab, 1999.

[4] Mihalcea R, Tarau P. "TextRank: bringing order into texts" // Proceedings of empirical methods in natural language processing. [s.l.]:[s.n.],2004.

[5] Langville A N, Meyer C D,"Google's PageRank and beyond: the science of search engine rankings".Princeton: Princeton University Press, 2006.

[6] Z. Duan, G. S. Liu,"Method of Building User Profile Based on TextRank", Computer Technology and Development, vol.25, no.10, Oct. 2015, pp.1-6.

[7] J. Yang, D. Ji, D. F. Cai, C. Dai, "Keyword Extraction in Multi-Document Based on TextRank Technology", The fourth session of national academic conference on information retrieval and information content security , Nov. 2008, pp.397-404.

[8] P. Li, B. Wang, Z. W. Shi, Y. C. Cui, H. X. Li,"Tag-TextRank: a Webpage Keyword Extraction Method Based on Tags", Journal of

Computer Research and Development, 2012, vol.49, no.11, pp.2344-2351

[9] T. Xia, "Research on Keyword Extraction Using Document Topical Structure".New Technology of Library and Information Service, 2013, vol.9, pp.30-34.

[10] K. Fang, L. X. Han,"Weighted Textrank keyword extraction single document based on hidden Markov model",Information Technology, 2015, no.4, pp.114-120.

[11] Mikolov T,Chen K,Corrado G, et al."Efficient estimation of word representations in vector space".arXiv Preprint arXiv, 2013, pp.1301-3781.

[12] W. Dong, Research of recommendation algorithm based on LDA and Word2Vec,Beijing: Beijing University of Posts and Telecommunications, pp.30-37, 2015.

[13] Q. Li,H. W. Zhu, Z. G. Lu, "Using Word2vec to Extract Abstract Keywords".Beijing: http://www.paper.edu.cn.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean,"Efficient Estimation of Word Representations in Vector Space", In Proceedings of Workshop at ICLR, 2013.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean,"Distributed Representations of Words and Phrases and their Compositionality", In Proceedings of NIPS, 2013.

[16] L. Zhou, Research on the principle and application of Word2vec, Sci-Tech Information Development & Economy,2015, vol.25, no.2, pp.145-148.

[17] Y. P. Li, C. Jin, J. C. Ji,A Keyword Extraction Algorithm Based on Word2vec, e-Science Technology & Application, 2015, vol.6, no.4, pp.54-59.

[18] Mihalcea R, Tarau P,"TextRank: Bringing Order into Texts", In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004, pp.404-411.

[19] Page L, Brin S, Motwani R, et al. "The PageRank Citation Ranking: Bringing Order to the Web", Stanford Digital Library Technologies Project, 1998.