WS-Rank: Bringing Sentences into Graph for Keyword Extraction

Fan Yang, Yue-Sheng Zhu(✉), and Yu-Jia Ma

Communication and Information Security Lab, Shenzhen Graduate School, Institute of Big Data Technologies, Peking University, Shenzhen, China
{yangfan0705,zhuys}@pku.edu.cn

Abstract. Graph-based method is one of the most efficient unsupervised ways to extract keyword from a single web text. However, rarely did the previous graph-based methods consider the sentence importance. In this paper, we propose a graph-based keyword extractor WS-Rank which brings sentences into graph where sentences are distinctively treated according to their importance. The candidate keywords are extracted through the voting mechanism between words and sentences. To evaluate the experiment, we compare our method with TextRank, a graph-based method which uses the logic distribution relationship only between words. Experiment on 13702 web texts carried out shows that WS-Rank achieves more ideal results with an average F-score of 25.20 %.

1 Introduction

Due to the explosion of web information, it becomes more difficult to search and manage the network resources. Keyword extraction aims to select a set of words from a text as its short summary, which can help people to identify whether they are interested quickly. Since keyword extraction in manual way is very expensive and time-consuming, many studies have been done for keyword extraction.

The keyword extraction method based on graph is simple and robust and has been used in many ways [2,7]. A representative method is TextRank [4] which uses a syntactic graph, where vertices represent words and edges represent word co-occurrence within a fixed window. After that, some varietal methods of TextRank are proposed to extract keyword such as Tag-TextRank [5] and TimedTextRank [6]. Besides, recent years have seen a lot of applications of keyword extraction using graph-based methods, especially in the field of social networks [1,3]. In this paper, sentence importance is brought into graph. We provide a method for keyword extraction using a graph where vertices represent words and sentences. The edges in graph represent the word existence in corresponding sentence. The graph is constructed according to the relationship between words and sentences. Besides, WS-Rank is proposed as a ranking algorithm to extract keyword, which will be introduced in the following section.

F. Li et al. (Eds.): APWeb 2016, Part II, LNCS 9932, pp. 474–477, 2016.
DOI: 10.1007/978-3-319-45817-5_49

2 WS-Rank: Sentence-Based Extractor

WS-Rank is an unsupervised, graph-based and language-independent keyword extractor. In this section, the candidate keyword graph and the words ranking algorithm of WS-Rank will be described.

WS-Rank uses a graph where vertices stand for words and sentences which are connected by undirected edges. We consider a word vertex and a sentence vertex are directly connected if the sentence contains the corresponding word. After the candidate keyword graph of WS-Rank is constructed, the score associated with each word vertex is set to an initial value of 1 and the score of each sentence vertex is set to 0. The ranking algorithm of WS-Rank runs on the graph for several iterations until a convergence is reached. The algorithm contains two steps: the transfer of score from word vertices to sentence vertices according to the corresponding edge weight which is determined by the importance of sentences and the transfer of score from sentence vertices to corresponding word vertices according to the importance of words. The score of the word vertex W_i is defined by the recursive formula (1):

$$WS(W_i) = (1-d) + d* \\ \sum_{S_m \in In(W_i)} \left(\frac{\mu_{mi}}{\sum_{W_k \in Out(S_m)} \mu_{mk}} * \sum_{W_j \in In(S_m)} \frac{\omega_{jm}}{\sum_{S_n \in Out(W_j)} \omega_{jn}} WS(W_j) \right) \quad (1)$$

where $In(W_i)$ and $In(S_m)$ represent the set of vertices that points to W_i and S_m respectively. $Out(W_j)$ and $Out(S_m)$ represent the set of vertices that W_i and S_m points to respectively. μ_{mi} represents the edge weight from S_m to W_i (the importance of W_i is measured by μ_{mi}). ω_{jm} is the edge weight from W_j to S_m (the importance of S_m is measured by ω_{jm}). d is a damping factor that gives the probability of jumping from a vertex to another random vertex in the graph. Usually, the damping factor is set to 0.85 [4]. In the formula, each word vertex gives its score to the adjacent sentence vertices and each sentence vertex gives its score back to the adjacent word vertices according to the corresponding edge weight. To evaluate the effect of sentence importance separately, we consider the words have the same importance, which means formula (1) can be described as:

$$WS(W_i) = (1-d) + d* \sum_{S_m \in In(W_i)} \frac{\sum_{W_j \in In(S_m)} \frac{\omega_{jm}}{\sum_{S_n \in Out(W_j)} \omega_{jn}} WS(W_j)}{|Out(S_m)|} \quad (2)$$

where $|Out(S_m)|$ represents the number of the edges which connect S_m. In this formula, each word vertex gives its score to the adjacent sentence vertices according to the corresponding edge weight and each sentence vertex gives its score back to the adjacent word vertices equally. A sample of WS-Rank in a short Chinese text is shown in Fig. 1 where the edge weight is represented by arrows with different size (Suppose $S1$ is more important than $S2$ and $S3$). The score of each vertex is computed iteratively until a convergence is reached. The top ranked vertices are extracted as the keywords.

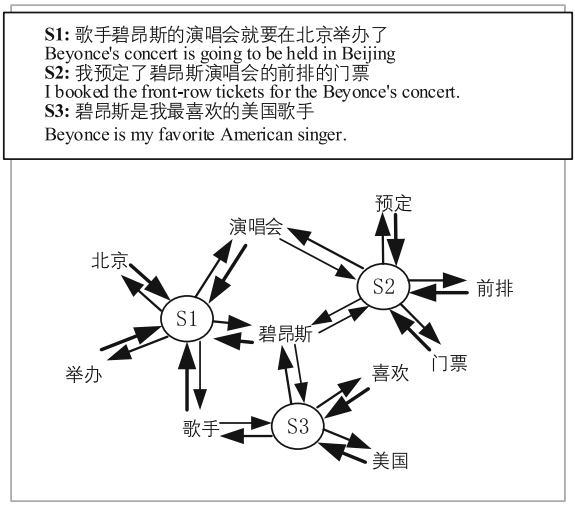

Fig. 1. Sample graph for keyword extraction with WS-Rank

3 Experiment and Discussion

To investigate experimental result when introducing the sentence importance, we give different values to the edge weight from word vertices to sentence vertices.

The experiment is based on a large collection of Chinese texts[1] from 163.com. In this dataset, there are 13702 texts. The average keywords number of the dataset is 2.4 words. The word segmentation method used in our experiment is based on NLPIR[2]. The stop words are removed after word segmentation. To evaluate our method, we use precision($P = |CA \cap CB|/|CB|$), recall($R = |CA \cap CB|/|CA|$) and the macro-average F-score($F = |2 \times P \times R|/(P+R)$) where CA represents the set of keywords extracted in manual way and CB represents the set of keywords extracted by the method used in this paper.

As we know, the first or the last sentence usually plays a summative role of the corresponding paragraph. In the experiment, these sentences are selected as important sentences while the others are treated as normal sentences. The value of ω_{ij} (as is mentioned above, ω_{ij} represents the edge weight from W_i to S_j) is set to k if S_j is important sentence. Otherwise, ω_{ij} is set to 1. The keywords are extracted according to different value of k, the result of which is shown in Table 1. The highest F-score is achieved when k is 1.97, which means when the first and the last sentences are given 1.97 times the importance of other sentences, the experimental result is the optimum. It is also shown in Table 1 that if we neglect or give an overemphasis on the important sentences, the F-score will decrease. We also compare our method with TextRank which merely considers the relationship between words. We can see the performance of WS-Rank is better than that of

[1] http://nlp.csai.tsinghua.edu.cn/~lzy/#Data.
[2] http://ictclas.nlpir.org/.

Table 1. Results of WS-Rank and TextRank when keyword number is 3

Method	k	Prescion(%)	Racall(%)	F-score(%)
WS-Rank	1.00	22.26	27.91	24.77
	1.50	22.55	28.27	25.09
	1.95	22.63	28.37	25.18
	1.97	22.65	28.39	25.20
	2.00	22.63	28.37	25.17
	2.50	22.48	28.18	25.01
TextRank	-	20.91	26.22	23.26

TextRank. In particular, when k is set to 1(the important and normal sentences are equally treated like TextRank), WS-Rank is still better than TextRank, which illustrates WS-Rank is meaningful even without the consideration of edge weight. In TextRank, a word with wide distribution means the corresponding word vertex has more adjacent vertices, from which the word can get more score through the rank algorithm. Like TextRank, word distribution is also considered in WS-Rank for that if a word has wide distribution, it can get more votes from the adjacent sentence vertices. That is to say, WS-Rank has the advantage of TextRank and gets a better performance.

In the end, it needs to be stressed that the highlighted sentence of WS-Rank is not limited to the first or the last sentence in a paragraph but also the sentence which can be manually labeled, which is more helpful to improve the experimental result. Through the experiment and comparison, we can draw a conclusion that the introduction of sentence importance is considerable in the graph-based keyword extraction.

References

1. Abilhoa, W.D., de Castro, L.N.: A keyword extraction method from twitter messages represented as graphs. Appl. Math. Comput. **240**, 308–325 (2014)
2. Boudin, F.: A comparison of centrality measures for graph-based keyphrase extraction. In: International Joint Conference on Natural Language Processing (IJCNLP), pp. 834–838 (2013)
3. Li, L., Su, C., Sun, Y., Xiong, S., Xu, G.: Hashtag biased ranking for keyword extraction from microblog posts. In: Zhang, S., Wirsing, M., Zhang, Z. (eds.) KSEM 2015. LNCS, vol. 9403, pp. 348–359. Springer, Heidelberg (2015). doi:10.1007/978-3-319-25159-2_32
4. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Association for Computational Linguistics (2004)
5. Peng, L., Bin, W., Zhiwei, S., Yachao, C., Hengxun, L.: Tag-textrank: a webpage keyword extraction method based on tags. J. Comput. Res. Dev. **11**, 014 (2012)
6. Wan, X.: Timedtextrank: adding the temporal dimension to multi-document summarization. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 867–868. ACM (2007)
7. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. AAAI **8**, 855–860 (2008)