Hashtag Biased Ranking For Keyword Extraction From Microblog Posts*

Lin Li[1], Chang Su[1], Yueqing Sun[1], Shengwu Xiong[1], Guandong Xu[2]

[1] School Of Computer Science & Technology, Wuhan University Of Technology
[2] Advanced Analytics Institute, University of Technology, Sydney
{cathylilin, suchang, yqsuan, xiongsw}@whut.edu.cn
[3] Guandong.Xu@uts.edu.au

Abstract. Nowadays, a huge amount of text is being generated for social networking purpose on the Web. Keywowrd extraction from such text benefit many applications such as advertising, search, and content filtering. Recent studies show that graph based ranking is more effective than traditional term or document frequecy based approaches. However, most work in the literature constructs word to word graph within a document or a collection of documents before applying a kind of random walk. Such a graph does not consider the influence of document importance on keyword extraction. Morevoer, social text like a microblog post usually has speical social features such as hashtag and so on, which can help us understand its topic. In this paper, we propose hashtag biased ranking for keyword extraction from a collection of microblog posts. We first build a word-post weighted graph by taking into account the posts themselves. Then, a hashtag biased random walk is applied on this graph, which guides our approach to extract keywords according to the hashtag topic. Last, the final ranking of a word is determined by the stationary probability after a number of interations. We evaluate our proposed method on a real Chinese microblog posts. Experiments show that our method is more effective than the traditional word to word graph based ranking in terms of precision.

1 Introduction

Recently, microblogs as a new social media have attracted researchers' interests [8]. Since there are usually thousands of posts in a miroblog platform, it is imporant for users to understand their content. For this purpose, various task have been studied, such as tag recommendation [23], keyword/keyphrase extraction [25, 26], topic analysis [1, 21], spammer detection [5], microblog retrieval [15, 18]. However, current explorations are still in an early stage and our understanding of microblog post content still remains limited. Keyword extraction is a foundation work for the above tasks and targets to represent the core

* This research was undertaken as part of Project 15BGL048, 2015AA015403, 2015BAA072 and 61303029 and supported by Hubei Key Laboratory of Transportation Internet of Things.

content of a post or a collection of posts. Therefore, it becomes an important and emergent research topic.

There are many approaches for keyword extraction from long text documents, such as intuitive frequency based, cluster based [3, 11], graph based [10, 13, 25] approaches. TextRank [13] is the first implemantation applying random walk on a word connnectivity graph. Those graph based ranking methods choose a word as one of topic keywowrds if the word frequently appears togehter with important words and show better than other approaches [10, 13, 25]. For graph based ranking, how to build graph is crucial. Previous methods mainly based on word to word relations weighted by statistical features such as term frequencies and co-occurences. For example, a link between two words are set up if the two words appear in at least a same document.

While it appears natural to use the graph based ranking to microblog posts, compared with traditonal long text collection, keyword extraction from microblog posts is more challenging in at least two aspects. The one is that microblog posts are short in length. Keyword extraction from traditional documents tries to filter less important words from a long text, but microblog posts themselve do not have enough good keywords. We obverse that users in microblogging like to publish posts related to a topic in a period of time. The accumulated number of topically related posts show the strength of the collective although a single post may not contain good enough keyword candidates. The traditional word to word graph in a single document does not model the relation between posts and thus is unable to use other posts to enhance keyword extraction. The other is that the social feature hashtag governs the main topic distribution of posts. The hashtag is a good topic indicator to build the topic relation among posts and then help us identify keywords from a collection of posts. How to use hashtag in keyword extraction is also another important problem. So far there is little work on keyword extraction from microblog posts [25, 26] and they still follow the direction of building a word to word connectivity graph without considering the influences of post importance and social feature on keyword extraction.

In this paper, we propose a hashtag biased ranking for keyword extraction from a collection of microblog posts. We think that given a post, keywords should be topically related to hashtag words and other topic related posts may have good keywords as supplementary. Based on these, our work follows the standard three steps of graph based keyword extraction. We first build a word-post weighted graph. If a word appears in a post, a link between them is set up. In such graph, a word will be selected as a keyword if the word frequently appears in imporant posts and the importance of a post is naturally determined by the linked important words. Also, kinds of weights can be added such as term frequency, document frequency and so forth. Then, a hashtag biased random walk is applied on this graph, which is similar to topical PageRank method [4]. A hashtag embedded post explicitly tells us its topic trend, so keyword extraction should make use of this indicator for better keywords. Last, the final ranking of a word is determined by the stationary probability of the hashtag biased random walk on the proposed word-post graph.

Our method can find keywords from both of a single post and a collection of posts by adjusting a random jumping vector. As we discussed above, a microblog post is short and not so informative for users. For example, a user submits a query to a microblog retreival engine and reads the returned results post by post. In such way, user has to summarize the main topics of the whole results set for better understand, which tells us that the collection of short posts is more interesting than a single post. Therefore, in this work we do evaluation on finding keywords from a collection of posts and conduct experiments on a Chinese microblog texts. Experimental results show that our method is effective in terms of precision. Our key contribution is to argue for building word to post graph and hashtag biased ranking, which considers both of posts and hastag influences on keyword extraction.

2 Related Work

Our work is related to unsupervised graph based keyword extraction. TextRank proposed by Mihalcea and Tarau [13] is the first graph based ranking algorithm to extract keywords and sentences for a given text. Following it, Liu et al. [10] used a topic model to learn topics of a document and than build a Topical PageRank (TPR) on word graph to measure word importance with respect to different topics. Based on the study by Liu et al. [10], recent work [25] addressed how to extract keyphrases from Twitter by improving the graph edge through a topic sensitive weighting and giving a probabilistic model for keyphrase ranking. The above studies rely only on a given single text to derive important key units like words, phrases and sentences. We think that a single short microblog post is not informative engough, so we model a word to post graph which takes into accounts the importance of other related posts in improving the quality of keyword extraction.

Other studies make use of external knowledge sources to improve the performance of keyword extraction. Wang et al. [20] represented a document as a semantic graph with synset from WordNet and extracted keywords from a modified PageRank algorithm. Wang et al. [22] used Wikipedia to construct a two-level concept based graph, instead of word based graph and ran PageRank and HITS rank on the graph. Wan et al. [19] proposed to use a small number of nearest neighbor documents to provide more knowledge to improve single document keyphrase extraction. Without utilizing any external corpus, our work applies random walk biased by hashtag context, a intrinsic feature in microblog posts.

Supervised approaches of keyword extraction are studied [6, 9, 17, 24]. Their experimental results show that supervised machine learning can obtain better results than traditional methods. Li et. al [9] investigated a set of features to measure the importance of keywords and select four supervised models for precision comparisons. Zhang et al. [24] utilized supervised randow walk for keyword extraction by combining multiple types of relations between words and automatically learning the weights of the edges between the words in the word graph of

each document. Labelled training data is crucial to optimze supervised model parameters and largely affects the extraction precision. Our work is unsupervised and orthogonal to supervised approaches.

3 Hashtag Biased Graph Ranking

3.1 Our Problem

Microblogging is such an infomatioin propagation flatform where users like to discuss hot events or topics, share their opinions and spread messages through their social networks. A single short microblog post may not satisfy the information needs of users. A collection of related posts could give users better understanding on what is going on regarding a topic. This characteristics is quite different from the traditional long text which keyword extraction is based on the assumptation that a single long document itself contain enough imporant words. Therefore, we assume that there is a collection of related microblog posts. Our task is to extract keywords from this collection. We aruge that the collection of posts can give users a more overall vision than a single post. Moreover, by adjusting a random jumping vector, we can still generate keywords for a single post. The collections of microblog posts are common, such as a set of search reults of microblog posts, a topic discussion group and so forth.

3.2 Word to Post Bipartite Graph Construction

Now given a collection of microblog posts, the word-post relationship can be intuitively represented as a bipartite graph. A bipartite graph, also called a bigraph, is a special graph from which the set of vertices can be decomposed into two disjoint sets such that no two vertices within the same set are adjacent. In the mathematical definition, a simple undirected graph G: $=(W \cup P, E)$ is called bipartite if W and P are disjoint sets, where W and P are the vertex set and E is the edge set of the graph. Let $n=|W \cup P|$. This graph is used as our original model where W is a set of words, the P is a set of microblog posts, as shown in Figure 1. An edge e connects a word w and a post p, if the word w is contained in the post p. In the context of KEYWORD EXTRACTION, we propose to rank words based on the inter-relationship of their corresponding posts. As a by-product, important posts could be mined as well by applying the proposed algorithm on the side of the posts with a relatively small modification.

3.3 Hashtag Biased Random Walk and Ranking

We rank word nodes in Figure 1 corresponding to the standing probability distribution (i.e. score) of a rankdom walker on the graph. Our hashtag biased random walk is defined as Equation 1, a modification of Tong et al. [16].

$$\overrightarrow{r'} = \alpha \widetilde{Q} \overrightarrow{r'} + (1 - \alpha) \overrightarrow{e_h} \tag{1}$$

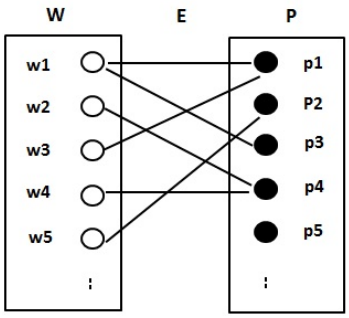

Fig. 1. Word-Post bipartite graph

$\overrightarrow{r}$ is $n \times 1$ rank vectors of nodes in the graph. $\widetilde{Q} = [q_{i,j}]$ is the weighted graph. In this paper, we investigate two popular weighting strategies, i.e., TF and TFIDF [12]. $\overrightarrow{e_h}$ is $n \times 1$ starting vector and h is the set of hashtag words with the constraint that $\sum_{j \in h} e(j) = 1$ and 0 for others. Our work directly uses the hashtag words to guide the jump probability of a random walk. Hashtag is generated by the author of a mircoblog post and explicitly reflect the topic of this post. Recent work [10, 25] discovered the topic of a word by latent topic model analysis, which ignore the intrinsic feature of microblog posts. We tune up the walk behavior and the jump behavior by a mixing parameter α, $0< \alpha <1$. From this formula we determine the overall score of a target node by counting *both* the number of nodes linking to a target node *and* the relative quality of each pointing node.

After constructing the word to post graph and applying the random walk on it, we can sort the nodes by their ranks using Equation 1. The recursive running of Equation 1 gives the probability distribution that the walker is on nodes after t iterations. When t equals to 1, no heuristic is used. When t is large enough, r_i will gradually converge to a stationary distribution. Then, the distribution induced on the state transitions of all the nodes in the graph produces a final ranking of these nodes. The initial state is chosen uniformly at random because in general the initial value will not affect final values, just the rate of convergence [14].

3.4 Algorithm Description

Equation 1 defines a linear system problem, where $\overrightarrow{r}$ is determined by:

$$\begin{aligned} \overrightarrow{r} &= (1-\alpha)(I - \alpha\widetilde{Q})^{-1}\overrightarrow{e_h} \\ &= (1-\alpha)Q^{-1}\overrightarrow{e_h} \end{aligned} \tag{2}$$

As dicussed in [16], directly computing Q^{-1} is impractical when the dataset is larege, since it requires quadratic space and cubic pre-computation. Linear correlations exist in many real graph, which means that we can approaximate $\widetilde{Q}$ by low rank approximation and then compute Q^{-1} efficiently. In this paper,

eigen-value decomposition is used after partition the whole graph into several commuties. We provide a sketch of our hashtag biased ranking procedure in the format of pseudo code in Table 1 and Table 2.

Table 1. Hashtag biased graph ranking(Offline Part)

Input: The normalized weighted matrix $\widetilde{Q}$ and the starting vector $\overrightarrow{e_h}$.	
Output: The ranking vector $\overrightarrow{r}$.	
Offline: Graph Partition and Matrix Decomposition	
1. Initializing disjoint-sets structure on word to post undirected graph [2];	
2. The k connected components (partitions) are calculated based on the edges in the graph.	O($n + \|E\|$)
3. Decompose $\widetilde{Q}$ into two matrices: $\widetilde{Q}$ =$\widetilde{Q_x}$ +$\widetilde{Q_y}$	O($\|E\|$)
4. Let $\widetilde{Q}_{x,i}$ be the i^{th} partition.	
5. Compute and store $Q_{x,i}^{-1}$=$(I - \alpha\widetilde{Q}_{x,i})^{-1}$ for each partition i according to Equation 2;	O($2n^3 + 2n^2$)
6. Do eigen-value low rank approximation for $\widetilde{Q_y} = USV$ where each column of U is the eigen-vector of $\widetilde{Q_y}$ and S is a diagonal matrix whose diagonal elements are gigen values of $\widetilde{Q_y}$;	O($2n^3$)
7. Let Q_x^{-1} is a block-diagonal matrix where each block is denoted as $Q_{x,i}^{-1}$;	
8. Compute and store $\widetilde{\Lambda} = (S^{-1} - \alpha V Q_x^{-1} U)^{-1}$;	O($6n^3 + 4n^2$)

Table 2. Hashtag biased graph ranking(Online Part)

Online: Iteration Computation	
Do Loop	
9. $\overrightarrow{r}_0 \leftarrow Q_x^{-1}\overrightarrow{e_h}$, do random walk within the partition that contains the starting point $\overrightarrow{e_h}$;	O($2n^2$)
10.$\overrightarrow{r} \leftarrow V\overrightarrow{r}_0$, jump from word-post space to latent space V;	O($2n^2$)
11.$\overrightarrow{r} \leftarrow \widetilde{\Lambda}\overrightarrow{r}$, do random walk within the latent space $\widetilde{\Lambda}$;	O($2n^2$)
12.$\overrightarrow{r} \leftarrow U\overrightarrow{r}$, jump back to word-post space U;	O($2n^2$)
13.$\overrightarrow{r} \leftarrow Q_x^{-1}\overrightarrow{r}$, do random walk within each partition;	O($2n^2$)
14.$\overrightarrow{r} \leftarrow (1-\alpha)(\overrightarrow{r}_0 + \alpha\overrightarrow{r})$;	O($3n + 1$)
Until convergence	
15.Quicksort the elements in $\overrightarrow{r}$ BY ASCENT;	O($nlogn$)

The input matrix $\widetilde{Q}$ is weighted by TF or TFIDF and normalized by graph Lapalician($\widetilde{Q}$=$D^{-1/2}Q'D^{-1/2}$) where Q' is the original weighting matrix [27]. The extraction of connected components from an undirected graph is calculated in Step 1 and 2. On the basic initialization of the disjoint-sets structure [2], each node in graph is in its own set. The connected components are calculated based on the edges, so update the disjoint-sets structure when each edge is added into the graph. Readers can refer to [2] for detail. The time complexity for calculating

the connected components is only slightly larger than $O(n + |E|)$ where n, i.e., $|W \cup P|$ is the number of nodes and $|E|$ is number of edges in the graph.

Step 3 decomposes $\widetilde{Q}$ into two matrices: $\widetilde{Q} = \widetilde{Q_x} + \widetilde{Q_y}$ according to the connected components, where $\widetilde{Q_x}$ contains all within-paritition links and $\widetilde{Q_y}$ contains all cross-partition links. The time complexity of Step 3 is $O(|E|)$ depending on the number of edges in the graph [7]. Step 4 and 5 do matrix compuation for each partition in $\widetilde{Q_x}$ based on Equation 2. The time complexity of matrix muliplication and matrix substraction, i.e., $I - \alpha\widetilde{Q}_{x,i}$ is $O(2n^2)$ and its invert matrix computation needs $O(2n^3)$.

Step 6 and 7 do low rank approximation for $\widetilde{Q_y}$ for computation preparation of Q^{-1} in Equation 2. Step 8 is a key process to compute Q^{-1} by combing $\widetilde{Q_x}$ and $\widetilde{Q_y}$. As dicussed in [16], it is the most time-consuming step with the time complexity $O(6n^3 + 4n^2)$. The following proof gives computation details of Q^{-1}. According to Step 3, we have:

$$\widetilde{Q} = \widetilde{Q_x} + \widetilde{Q_y} = \widetilde{Q_x} + USV \tag{3}$$

Then the inverse matrix in Equation 2 is computed as:

$$\begin{aligned} Q^{-1} =& (I - \alpha\widetilde{Q})^{-1} \\ =& (I - \alpha\widetilde{Q_x} - \alpha USV)^{-1} \\ =& Q_x^{-1} + \alpha Q_x^{-1} U\widetilde{\Lambda} V Q_x^{-1} \end{aligned} \tag{4}$$

where

$$\begin{aligned} X &= (I - \alpha\widetilde{Q}_x)^{-1} = Q_x^{-1} \\ (X - USV)^{-1} &= X^{-1} + X^{-1}U\widetilde{\Lambda}VX^{-1} \\ \widetilde{\Lambda} &= (S^{-1} - VX^{-1}U)^{-1} \end{aligned}$$

Based on Equation 2, Step 9 to 14 in online phase $\widetilde{r}$ is computed step by step, represented as:

$$\widetilde{r} = (1 - \alpha)(Q_x^{-1}\overrightarrow{e_h} + \alpha Q_x^{-1} U\widetilde{\Lambda} V Q_x^{-1}\overrightarrow{e_h}). \tag{5}$$

It can be seen that the approximation of our algorithm comes from the low rank decomposion for $\widetilde{Q}_y$. Our experiments show the online computation is very fast compared to the offline computation. In addition, users can select some words as the starting vector $\overrightarrow{e_h}$ to extract keywords related to it online. In this paper, we set the starting vector consisiting of hashtag words in microblog posts since hashtag intrinsically represents the key topics of a post. It will help us find good important keywords, which is verified by our experimental results.

4 Experiments

4.1 Dataset and Evaluation

The data used in our paper was crawled from Sina Weibo [4] from the end of March 2012 to the end of June 2012. There were 74662 microblog posts in total.

[4] http://www.weibo.com, one of most popular microblogging platform in China.

They were posted in 14 IT/technology related topics discussion groups. We segmented these mircoblog posts, filtered stop words, and finally got 13167 distinct words. After that, we computed TF and $TFIDF$ scores of those words and build different graphs for each discussion group. The precision score at the top K keywords of a discussion group is defined as:

$$Precision@K = \frac{\#important\ keywords}{K}. \tag{6}$$

The measure $Precision@K$ means how many good important keywords our algorithm gives at the top K list. We set K=5 and 10 in our evaluation. The average precision score of 14 topic discussion groups is reported.

We treat each topic group as a collection of posts where hashtags are topic related. Our target is to identify top K important keywords from each group. Whether a keyword is important or not in a group is judged by our three lab members. When we take the extracted top K representative keywords, the three lab memebers manually judge whether these keywords can be considered to accurately reflect the meaning of its post. The precision values of all 14 topics are computed and its average score by three members is reported in our experiments. In addition, for each word ranking list generated by different graphs, we remove the hashtag words from it. Since hashtag words are clearly important in these posts, we want to get other important keywords that should be more interesting to users.

4.2 Experimental Results and Discussions

We will compare the effectiveness of a set of ranking approaches with our approach according to three apsects, i.e., node types, jumping strategies and weighting strategies. The ranking approaches are listed as follow.

1. **WW-OC-A:** This approach builds **word to word** graph weighted by **co-occurences** and jumps to **any** nodes in the graph. It is widely used in recent works [10, 13, 25].
2. **WP-TF-A:** This is a **word to post** graph based ranking. The graph is weighted by **TF** and a random walker jumps to **any** nodes including word and post nodes. It is a variant and weighted verison proposed in [16].
3. **WP-TF-W:** This is a **word to post** graph based ranking. The graph is weighted by **TF** and a random walker jumps to any of **word** nodes. It is also a variant and weighted verison proposed in [16].
4. **WP-TF-H:** This is our proposed **word to post** graph based ranking. The graph is weighted by **TF** and a random walker jumps to any of **hashtag** word nodes.
5. **WP-TI-A:** This is a **word to post** graph based ranking. The graph is weighted by **TFIDF** and a random walker jumps to **any** nodes including word and post nodes.
6. **WP-TI-W:**This is a **word to post graph** based ranking. The graph is weighted by **TFIDF** and a random walker jumps to any of **word** nodes.

Table 3. The average precision scores of 14 discussion groups

	Precision@5	Precision@10
WW-OC-A	0.3857	0.3357
WP-TF-A	0.3429	0.4286
WP-TF-W	0.3571	0.4357
WP-TF-H	**0.6**	**0.5429**
WP-TI-A	0.4714	0.4571
WP-TI-W	0.4714	0.5143
WP-TI-H	**0.7571**	**0.6786**

7. **WP-TI-H:** This is our proposed **word to post graph** based ranking. The graph is weighted by **TFIDF** and a random walker jumps to any of **hashtag** word nodes.

Comparisons among Node Types The overall experimental results are show in Table 3. The highest precison scores are achieved by our proposed hashtag biased ranking in both of Precision@5 and Precision@10. The nodes in the baseline WW-OC-A are only words and those in our proposed word to post graph are both of words and posts. We compare WW-OC-A with our proposed word to post graph (the last six rows in Table 3). In terms of Precision@10, our word to post graph based ranking shows higher scores than the word to word graph based ranking. The best one is our hashtag biased ranking by only jumping to hashtag words and its precision scores are **0.5429** using TF weighting and **0.6786** using TFIDF weighting. In terms of Precision@5, our word to post graphs win the word to word graph in most cases. Especially, our hash biased ranking shows much bettern results than the baseline, i.e., **0.6** VS. **0.3857**, **0.7571** VS. **0.3857**. These results tell us that the word to post graph takes into account the quality of posts in ranking, which can improve the quaity of keyword extraction. In other words, important keywords come from important posts with high probability.

Comparisons mong Jumping Strategyies Moreover, hashtag is natually existed in some posts and it highlights their topic. As defined in Equation 1, our proposed word to post graph with hashtag biased random walk produces keywords closely related to hashtag words, i.e., $\overrightarrow{e_h}$. We compare it with two other jumping strategies. One is jumping to any nodes in the word to post graph and the other is jumping to any word nodes. As shown in Figure 2, the last columns are produced by our hashtag biased jumping strategies, i.e., WP-TF-H and WP-TI-H. We can see that our hashtag biased jumping is much better than the two jumping strategies in both Precision@5 and Precision@10. Its improvements are **60.6%** and **48.46** compared with WP-TF-A and WP-TI-A which jump to any nodes in the word to post graph. Also its precision scores are higher than WP-TF-W and WP-TI-W which jump to any word nodes in the word to post graph. Users in a microblogging flatform publish posts and like to use hashtag to attract other users' attention. The user-generated hastag is a useful evidence

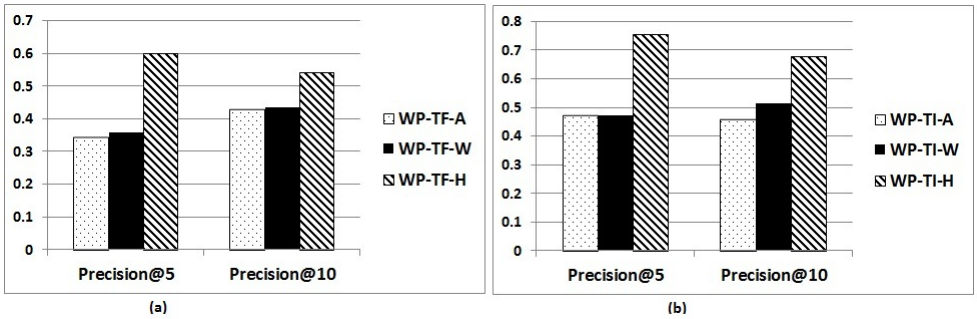

Fig. 2. Comparisons among different jumping strategies

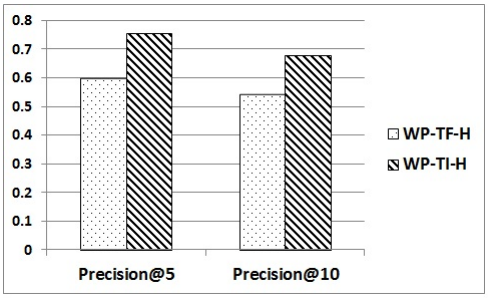

Fig. 3. Comparisons among different weighting strategies

to clearly tell us that those posts are topically related to it. Random walk in our word to post graph with hashtag biased jumping lets our ranking algorithm put more hashtag related keywords in the top ranking list.

Comparisons among Weighting Strategies Last, our word to post graph can be weighted by TF or TFIDF which are commonly used in the field of Information Retrieval (IR). We investigate the influences of the two weighting stratgies on keyword extraction. To make it clear, we show the results of our hashtag biased random walk approaches, i.e., WP-TF-H and WP-TI-H, as shown in Figure 3. The left column is TFIDF wieghting and the right column is TF weighting at each precision measure. It is obvious that TFIDF weighting is much better than TF in both of Precision@5 and Precision@10. For example, using word to post graph with hashtag biased jumping, TFIDF produces **0.7571** and the score of TF is **0.6**. The improvement is **26.18%** in term of Precision@5 and it is **25%** in term of Precision@10. The results are consistent with the viewpoint of IR. TFIDF gives less weights on words with high document (post) frequency. In other words, the extracted keywords should be representative and infomative in a post, not commonly appeared in other posts.

5 Conclusions

In this paper, we introduce a novel word to post graph based ranking by adopting a hashtag biased random walk. The proposed ranking algorithm can extract important keywords from a collection of microblog posts. The experimental reults show that our algorithm has higher precision scores that traditional word to word graph based ranking and the word to post graph based ranking without a hashtag biased randwom walk. In the future, we can easily extend our algorithm to extract keywords from a single post by considering the other related posts. Topic space based weighting is also an interesting topic.

References

1. B. Bi, Y. Tian, Y. Sismanis, A. Balmin, and J. Cho. Scalable topic-specific influence analysis on microblogs. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 513–522, 2014. ACM.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms. McGraw-Hill, 1990.
3. M. Grineva, M. Grinev, and D. Lizorkin. Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 661–670, 2009. ACM.
4. T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. on Knowl. and Data Eng.*, 15(4):784–796, July 2003.
5. X. Hu, J. Tang, and H. Liu. Leveraging knowledge across media for spammer detection in microblogging. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 547–556, 2014. ACM.
6. A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, 2003.
7. G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48(1):96–129, Jan. 1998.
8. H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.
9. Z. Li, D. Zhou, Y.-F. Juan, and J. Han. Keyword extraction for social snippets. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1143–1144, 2010. ACM.
10. Z. Liu, W. Huang, Y. Zheng, and M. Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
11. Z. Liu, P. Li, Y. Zheng, and M. Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266, 2009. Association for Computational Linguistics.
12. C. D. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval, 2008.

13. R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In D. Lin and D. Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, 2004. Association for Computational Linguistics.
14. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
15. R. Qiang, F. Liang, and J. Yang. Exploiting ranking factorization machines for microblog retrieval. In *Proceedings of the 22nd ACM international conference on Conference on Information and knowledge management*, CIKM '13, pages 1783–1788, 2013. ACM.
16. H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 613–622, 2006. IEEE Computer Society.
17. P. D. Turney. Learning algorithms for keyphrase extraction. *Inf. Retr.*, 2(4):303–336, May 2000.
18. J. Vosecky, K. W.-T. Leung, and W. Ng. Collaborative personalized twitter search with topic-language models. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 53–62, 2014. ACM.
19. X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 855–860. AAAI Press, 2008.
20. J. Wang, J. Liu, and C. Wang. Keyword extraction based on pagerank. In *Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'07, pages 857–864, 2007. Springer-Verlag.
21. W. Wang, H. Xu, W. Yang, and X. Huang. Constrained-hlda for topic discovery in chinese microblogs. In *Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD'14. Proceedings, Part II*, pages 608–619. Springer, 2014.
22. X. Wang, L. Wang, J. Li, and S. Li. Exploring simultaneous keyword and key sentence extraction: Improve graph-based ranking using wikipedia. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2619–2622, 2012. ACM.
23. W. Wu, B. Zhang, and M. Ostendorf. Automatic generation of personalized annotation tags for twitter users. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 689–692, 2010.
24. W. Zhang, W. Feng, and J. Wang. Integrating semantic relatedness and words' intrinsic features for keyword extraction. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 2225–2231. AAAI Press, 2013.
25. W. X. Zhao, J. Jiang, J. He, Y. Song, P. Achananuparp, E.-P. Lim, and X. Li. Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 379–388, 2011.
26. L. Zhiyuan, C. Xinxiong, and S. Maosong. Mining the interests of chinese microbloggers via keyword extraction. *Foundations and Trends in Information Retrieval*, 6(1):76–87, 2012.
27. D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.