Journal of Web Engineering, Vol. 17, No. 1&2 (2018) 093–120

UNSUPERVISED KEYWORD EXTRACTION FROM MICROBLOG POSTS VIA HASHTAGS[a]

LIN LI
[1]*School of Computer Science & Technology, Wuhan University of Technology*
[2]*Hubei Key Laboratory of Transportation Internet of Things, Wuhan University of Technology*
Wuhan, 430070, China
cathylilin@whut.edu.cn

JINHANG LIU YUEQING SUN
School of Computer Science & Technology, Wuhan University of Technology
Wuhan, 430070, China
ryukinkou@whut.edu.cn *yqsuan@whut.edu.cn*

GUANDONG XU
Advanced Analytics Institute, University of Technology, Sydney
NSW 2007, Australia
Guandong.Xu@uts.edu.au

JINGLING YUAN LUO ZHONG
School of Computer Science & Technology, Wuhan University of Technology
Wuhan, 430070, China
yjl@whut.edu.cn *zhongluo@whut.edu.cn*

Received August 27, 2016
Revised January 10, 2018

[a]This work was supported by projects of 15BGL048, 2015BAA072, and WUT:2017II39GX.

Nowadays, huge amounts of texts are being generated for social networking purposes on Web. Keyword extraction from such texts like microblog posts benefits many applications such as advertising, search, and content filtering. Unlike traditional web pages, a microblog post usually has some special social feature like a hashtag that is topical in nature and generated by users. Extracting keywords related to hashtags can reflect the intents of users and thus provides us better understanding on post content. In this paper, we propose a novel unsupervised keyword extraction approach for microblog posts by treating hashtags as topical indicators. Our approach consists of two hashtag enhanced algorithms. One is a topic model algorithm that infers topic distributions biased to hashtags on a collection of microblog posts. The words are ranked by their average topic probabilities. Our topic model algorithm can not only find the topics of a collection, but also extract hashtag-related keywords. The other is a random walk based algorithm. It first builds a word-post weighted graph by taking into account posts themselves. Then, a hashtag biased random walk is applied on this graph, which guides the algorithm to extract keywords according to hashtag topics. Last, the final ranking score of a word is determined by the stationary probability after a number of iterations. We evaluate our proposed approach on a collection of real Chinese microblog posts. Experiments show that our approach is more effective in terms of precision than traditional approaches considering no hashtag. The result achieved by the combination of two algorithms performs even better than each individual algorithm.

Keywords: Keyword Extraction, Microblog Post, Hashtag, Topic Model, Random Walk

Communicated by: D Schwabe & Q Li

1 Introduction

Recently, microblog as a new social media has widely attracted researchers' interests [14]. Compared with traditional media like newspapers and TV, microblog has several distinguished characteristics, such as rich information sources, quick transmission, large influence range, timeliness and active interactions among users, and so on. Microblog users naturally and easily initiate discussions on hot topics or events. Since there are usually thousands of posts updated daily in a miroblog platform, both academic and industrial communities show great interests in post content understanding. For this purpose, various tasks have been studied, such as tag recommendation [38], tag clustering [27], keyword/keyphrase extraction [42, 41], topic analysis [2, 36], spammer detection [11], and microblog retrieval [26, 32]. Keyword extraction is a fundamental work for the above tasks and targets to represent the core content of a post or a collection of posts. Therefore, effectively finding keywords in microblogs becomes an important and emergent research topic.

Keyword extraction from long text documents is a longstanding topic with various research directions, such as intuitive frequency based, cluster based [8, 20, 15], topic model based [6, 7, 9, 18, 29], random walk based [19, 22, 41] approaches. Compared with a traditional long text document, a microblog post is short, typically no more than 140 characters. Therefore, it is usually not so informative for users. For example, a user submits a query to a microblog retrieval engine and reads the returned results post by post. In such way, the user has to read through a large amount of results and summarize main topics for better understanding. It indicates that the collection of short posts for a topic is more interesting than a single post. Moreover, the embedded hashtag in some post governs the main topic of the posts as a social feature. It is recognized that the hashtag is a good topic indicator to build the topic relation

among posts and then can help us identify keywords from a collection of posts. However, how to use hashtags for keyword extraction is still an open problem. So far there is little work on keyword or keyphrase extraction from microblog posts [42, 41]. In addition, those works do not study the influences of social features on keyword extraction, e.g., hashtags. Current explorations are still in an early stage and our understanding of microblog post content still remains limited.

A hashtag is a type of label or metadata tag used on social network and microblogging services and is intended for discussion of a particular topic or event. Therefore, it makes users easier to find messages with a specific theme. Users create and use hashtags by placing the hash character (or number sign) # in front of a word or unspaced phrase, either in the main text of a post or at the end. Searching for that hashtag will then return each post that has been tagged with it. For example, searching Twitter for #worldcup2014 returns many tweets from individuals around the world about the 2014 FIFA World Cup. Because of its widespread use, hashtag was added to the Oxford English Dictionary in June 2014. This paper addresses how to extract keywords by utilizing hashtag to improve extraction effectiveness. Two hashtag based algorithms are proposed. One is a topic model algorithm and the other is a random walk based algorithm. The combination of them are also discussed through experimental evaluation.

Topic Model Algorithm

LDA (Latent Dirichlet Allocation), as a topic model analysis method [3, 4], is effective in uncovering the underlying semantic structure of a document collection and has been applied to many kinds of documents, including email, scientific abstracts, newspapers and so forth [6, 7, 9, 18, 29]. LDA can capture the topic distributions of documents, so keywords can be naturally selected according to their topic probabilities. When applying LDA on microblog posts, keyword extraction can benefit from their intrinsic features. As we discussed above, hashtags, as a social feature input by users in mircoblog posts, explicitly represent the topics of posts. If some posts share the same hashtag words, it is reasonable to say that those posts are topically related. Although a single post is short, a set of topically related posts can give us more clues to extract keywords. To best of our knowledge, there are few studies which take into account the hashtag feature in LDA based keyword extraction.

In this paper, we propose an LDA based algorithm with hashtag constraints for extracting topical keywords from a collection of posts. A hashtag in a post explicitly tells us its topic trend, so keyword extraction should make use of this valuable indicator. Our algorithm constructs a topic model by connecting hashtag words with posts and thus finds latent topics that can best represent the hashtag related content of posts. Words are ranked by their average topic probabilities. Our topic model algorithm can not only extract words with topical meanings, but also select words topically related to hashtags.

Random Walk Based Algorithm

Recent studies show that the random walk based algorithm is more effective for keyword extraction than traditional term or document frequency based approaches [19, 22, 41]. It is crucial to build a graph where a random walk can be applied. Previous methods to build the graph are mainly based on word to word relations weighted by statistical features such as term frequencies and co-occurrence. For example, a link between two words is set up if these two words appear together in at least a same document. While it appears natural to

use the random walk based algorithm to microblog posts, compared with traditional long text documents, keyword extraction from microblog posts is more challenging. Microblog posts are short in length. Conventionally, keyword extraction from traditional documents aims to filter a few important words from a long text document, but microblog posts are short in length and do not have enough good keywords. We observe that users in microblogging publish posts related to a topic during a period of time. The accumulated number of topically related posts shows the strength of the collective although a single post may not contain good enough keyword candidates. However, the traditional word to word graph in a single document does not well model the relation between posts and thus the conventional approach is unable to use other posts to enhance keyword extraction. Current studies [41, 42] still follow the direction of building a word to word connectivity graph within a document or a collection of documents before applying a kind of random walk [19, 22, 41]. We argue that such a word to word graph does not consider the influence of document importance on keyword extraction.

In this paper, we propose a hashtag biased ranking for keyword extraction on microblog posts by using random walk mechanism on a word to post graph. We think that given a post, keywords should be topically related to hashtag words and other topic related posts might have good keywords as supplementary. Our algorithm has three steps. We first build a weighted word-post bipartite graph. If a word appears in a post, a link between them is set up. In such a graph, a word will be selected as a keyword if it frequently appears in important posts and the importance of a post is naturally determined by its linked important words. Also, different kinds of weights can be added on the graph edges, 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 [10]. A hashtag embedded post explicitly tells us its topic trend, so keyword extraction should make use of this indicator for better extraction results. 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 Contributions are lists as follows:

(1) We propose an LDA based algorithm with hashtag constraints. It can produce hashtag related keywords for better understanding on a collection of microblog posts.

(2) We propose a hashtag biased random walk algorithm which builds a word to post bipartite graph. It considers the influences of both posts and hashtag on keyword extraction.

(3) Comprehensive experiments are conducted on a collection of Chinese microblog posts and our algorithms are compared with a set of popular approaches including traditional clustering based, random walk based and topic model based methods. Moreover, the extraction results of the two proposed algorithms are combined for more effectiveness.

The remainder of this paper is organized as follows: Section 2 gives a summary of related works and Section 3 describes the overview of our approach. Section 4 and 5 present our proposed two algorithms. Section 6 illustrates the experimental results and discussions. Finally, Section 7 concludes this paper.

2 Related Works

2.1 Clustering in Keyword Extraction

Clustering based methods often divide the words into clusters and then choose the representative words in each cluster and finally merge them into keywords. Qamra [25] apply a community-based approach with temporal clustering by finding shared interests to identify topics and keywords. Grineva et al. [8] study graph community detection techniques to partition the word to word graph into thematically cohesive groups of terms. And they introduce a criterion function to select groups that contain key terms discarding groups with unimportant terms. Liu et al. [20] suggest an unsupervised clustering based method. Firstly the terms in a document are grouped into clusters based on semantic relatedness. Each cluster is represented by an exemplar term that is also the centroid of each cluster. keyphrases are extracted from the document using these exemplar terms. Wan and Xiao [33] propose to adopt clustering methods to find a small number of similar documents which provides more knowledge for building word graphs for keyword extraction.

The above studies show that combing the clustering algorithm with other techniques or criteria performs better than using clustering alone. Following this direction, the extraction results of our LDA algorithm and random walk algorithm are fused for performance improvement. Moreover, our LDA algorithm shows better than the traditional K-means clustering algorithm in terms of precision.

2.2 Topic Model in Keyword Extraction

LDA is widely studied for topical analysis. A number of approaches are extended from the original LDA model by designing their own probabilistic models [1, 23, 39]. However, these models have a higher complexity than the original LDA model. Furthermore, these approaches are dependent on specific information, such as citations or comments, which is difficult to generalize for every mircoblog post. Nallapati and Cohen suggest a method to find the topic-specific influences of blog posts, by analyzing citations between blog posts using machine learning techniques [23]. They propose a model named Link-PLSA-LDA that groups blog posts into two groups, "cited" and "citing", and builds a bipartite graph by citations, because citations were a good indicator of influences. Their model also considers the content of blog posts. Ahmed and Xing analyze blog posts from a perspective of ideology, using topical analysis by multi-view LDA [1]. They assume that the contents of blog posts are affected by the writers' ideological beliefs and the background topics of each ideology, so they add some more steps of the generation of each word in a document to the original LDA generative model. The study of Yano et al. introduces a comment prediction method from political blog posts, by applying LDA on blog posts [39].

Our topic model algorithm adds hashtag words as constraints when estimating the model parameters. It connects hashtag words with posts and thus finds latent topics that can best represent the hashtag related content of posts. Our topic model algorithm can not only extract words with rich topical meanings, but also select words topically related to hashtags .

2.3 Random Walk in Keyword Extraction

Random walk based algorithm usually works on a graph build from documents for keyword extraction. It chooses a word as one of topic keywords if the word frequently appears together

with important words [19, 22, 41]. TextRank proposed by Mihalcea and Tarau [22] is the first graph based ranking algorithm to extract keywords and sentences for a given text. Following it, Liu et al. [19] use a topic model to learn topics of a document and then build a Topical PageRank (TPR) on word to word graph to measure word importance with respect to different topics. Based on the study by Liu et al. [19], recent work [41] addresses 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 enough, so we model a word to post bipartite graph which takes into account the importance of other related posts in improving the quality of keyword extraction. Our preliminary results are in [16] and the combination of our LDA and random walk based algorithms are discussed for better performance in this paper.

2.4 Supervised Learning in Keyword Extraction

Supervised approaches of keyword extraction become popular recently [12, 17, 31, 40]. Their experimental results show that supervised machine learning can obtain better results than traditional unsupervised methods. Li et. al [17] investigate a set of features to measure the importance of keywords and select four supervised models for precision comparisons. Zhang et al. [40] utilize supervised random walk for keyword extraction by combining multiple types of relations between words and automatically learning the edge weights in the word to word graph of each document. However, labelled training data is crucial to optimize supervised model parameters and largely affects the extraction precision. The main problem of supervised approaches is getting an appropriate set of training data, which is not easy for large web data. Our work is unsupervised and orthogonal to supervised approaches.

2.5 Using External Sources in Keyword Extraction

There are some studies that make use of external knowledge sources to improve the performance of keyword extraction. Wang et al. [35] represent a document as a semantic graph with synset from WordNet and extracted keywords from a modified PageRank algorithm. Wang et al. [37] use Wikipedia to construct a two-level concept based graph, instead of word based graph and run PageRank and HITS rank on the graph. Wan et al. [34] propose to use a small number of nearest neighbor documents to provide more knowledge for improving single document keyphrase extraction. Without utilizing any external corpus, our work considers hashtag context, a intrinsic feature in microblog posts.

3 Overview Of Our Hashtag Enhanced Approach

3.1 Problem Description

Microblogging is such an information propagation platform where users like to discuss hot events or topics, share their opinions and spread messages through their social networks. Here, a screen shot of extracted posts from a Chinese microblog platform is shown in Figure 1. We use "haze" as search keyword and its search results are partly listed in Figure 1. In China, microblogs Sina Weibo and Tencent Weibo utilize a double-hashtag "#HashName#" format, since the lack of spacing between Chinese characters necessitates a closing tag. Generally, a

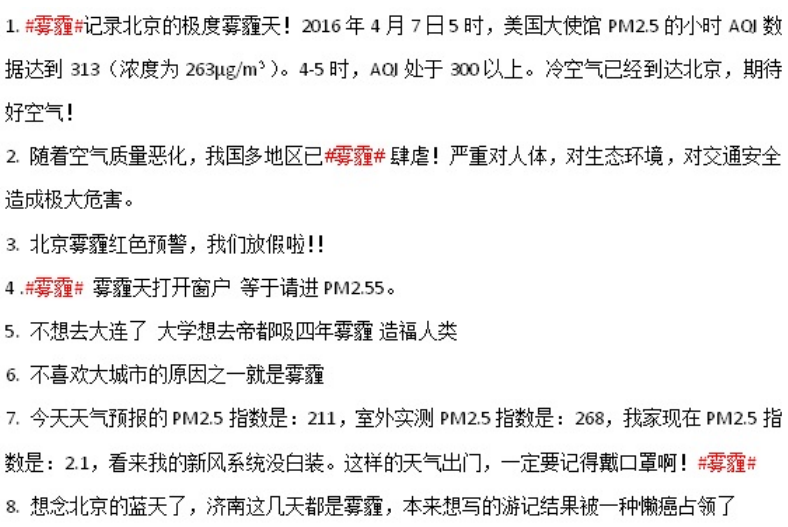

Fig. 1. A screenshot of extracted posts from a Chinese microblog platform

hashtag can be recognized as a label for content. From Figure 1, we can see that a single microblog post is short and it may not satisfy the information needs of users. Several related posts together could give users better understanding about what is going on regarding a topic. This characteristic is quite different from the traditional long text for which keyword extraction is based on the assumption that a single long document itself contains enough important words.

Therefore, we assume that there is a collection of related microblog posts, and our task is to extract keywords from the collection. We argue that the collection of posts can give users a more overall vision than a single post. The collections of microblog posts are common, such as a set of search results of microblog posts, a topic discussion group and so forth. Moreover, a post with hashtag explicitly shows its content in a certain topic. We believe that keywords from this kind of posts are more topically important than others without hashtags.

3.2 Flowchart Of Our Approach

We propose a hashtag enhanced keyword extraction approach for microblog posts by treating hashtags as topical indicators. The flowchart of our approach is shown in Figure 2. Our approach consists of two hashtag enhanced algorithms. One is a topic model algorithm that infers topic distributions biased to hashtags on a collection of microblog posts. Words are ranked based on their topic probabilities averaged by a number of topics. The topic model algorithm can estimate the topics of a collection and then extract hashtag-related keywords. The other is a random walk based algorithm. It first connects a word with a mircoblog post if the word appears in this post and thus a word-post graph is constructed. This graph could have weights on its edges by using term frequency based statistics, such as TF and TFIDF [21]. Next, a random walker works on this graph and restarts from hashtag words, which give higher probabilities on keyword candidates closer to hashtag words. Last, the final

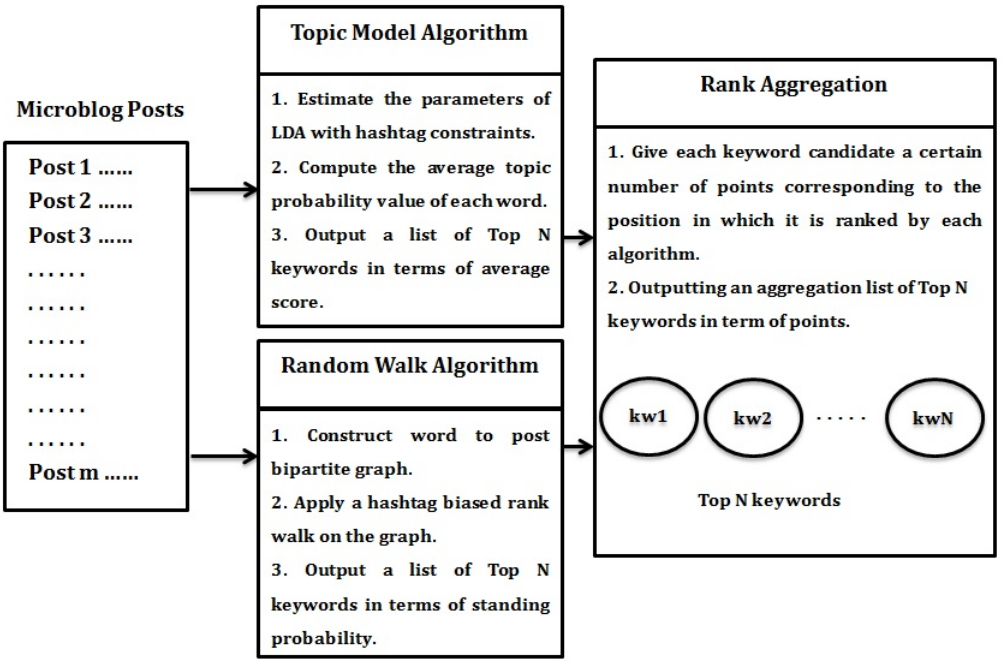

Fig. 2. Flowchart Of Our Approach

ranking of a word is determined by the stationary probability after a number of iterations.

Our two algorithms generate two keywords ranking lists given a collection. The two lists are aggregated by Borda's rule that is a single winner election method. The winner of an election is determined by giving each candidate a certain number of points corresponding to the position in which each voter ranks her. Here a voter is a keyword extraction algorithm. Once all points have been counted, the candidate with the most points is the winner. This rank aggregation benefits from the advantages of both two algorithms and generates better keywords representing the content of microblog posts.

4 Our Topic Model Algorithm with Hashtag Constraints

Assume that there is a collection of search results of mircorblog posts, or a collection of published posts in some microblog discussion group. We want to know what are the topics in this collection and what are the representative words. We propose a topic model based algorithm that enhances a popular latent topic model. i.e., LDA [4, 3] by adding hashtag words as constraints in topic estimation. The goal is to infer topic distributions related to hashtags.

4.1 Algorithm Description

In our topic model based approach, we treat the words of a mircoblog post collections as arising from a set of topics and add to LDA hashtag words associated with some posts. We jointly model the posts and the hashtag words, in order to find topics that will best predict

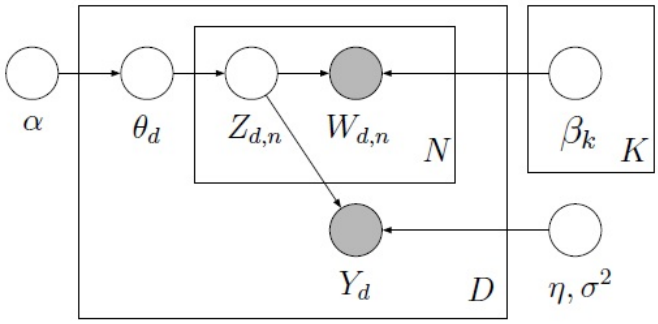

Fig. 3. LDA with hashtag constraints

the hashtags for these posts. As shown in Figure 3, we present the general version of our approach in a graphical model. Here the hashtag word is called response, notated by $y \in R$. The model parameters are the K topics $\beta_{1:K}$, the Dirichlet parameter α, and the response parameters η and σ^2. Note that each β_k is a vector of word probabilities. Under our model, each post and hashtag arises from the following generative process:

1. Draw topic proportions $\theta|\alpha \sim Dir(\alpha)$.

2. For each word
   (a) Draw topic assigment $z_n| \sim Mult(\theta)$.
   (b) Draw word $w_n|z_n, \beta_{1:K} \sim Mult(\beta_{z_n})$.

3. Draw hashtag word variable $y|z_{1:N}, \eta, \sigma^2 \sim N(\eta^T \bar{z}, \sigma^2)$.

$\bar{z} := (1/N)\sum_{n=1}^{N} z_n$ is defined, which means the components of $\bar{z}$ always sum to one. We assume that the hashtag word variable comes from a normal linear model. The covariates in this model are the (unobserved) empirical frequencies of the topics in the post. η is constituted of the regression coefficients on those frequencies. We treat $\alpha, \beta_{1:K}, \eta$, and σ^2 as unknown constants to be estimated and carry out approximate maximum-likelihood estimation using a variational expectation-maximization (EM) procedure, which is the approach taken in classical LDA as well [4]. We give a brief description of EM procedure in next part. More details about it are in the work [3]. After applying hashtag constraints based LDA on a collection of microblog posts, we can get its topic distributions. The words are ranked by their average topic probabilities. In this way, our approach can not only find the topics of a collection, but also give higher scores to hashtag-related keywords than others.

4.2 Parameter Estimation

Given a post and its hashtag, the posterior distribution of the latent variables is

$$p(\theta, z_{1:N}|w_{1:N}, y, \alpha, \beta_{1:K}, \eta, \sigma^2) = \frac{p(\theta|\alpha)(\prod_{n=1}^{N} p(z_n|\theta, \beta_{1:K}))p(y|z_{1:N,\eta,\sigma^2})}{\int d\theta p(\theta|\alpha) \sum_{z_{1:N}} (\prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta_{1:K}))p(y|z_{1:N}, \theta, \sigma^2)} \quad (1)$$

The normalizing value is the marginal probability of the observed data, i.e., the post $w_{1:N}$ and hashtag y. This normalizer is also known as the likehood. For efficient computation, the authors in [3] used variational methods to approximate the posterior.

Using the variational inference algorithm, we can estimate the approximate posterior distribution for each post-hashtag pair. And then we maximize the corpus-level evidence lower bound with respect to the model parameters $\beta_{1:K}, \eta, \sigma^2$. Note that α is fixed to $1/K$ times the ones vector. The probability of a word under a topic is proportional to the expected number of times that it was assigned to that topic [4], defined as

$$\hat{\beta}_{k,w}^{new} \propto \sum_{d=1}^{D} \sum_{n=1}^{N} 1(w_{d,n} = w)\phi_{d,n}^{k}. \tag{2}$$

Here, proportionality means that each $\hat{\beta}_{k,w}^{new}$ is normalized to sum to one. Till now, we get the topic distribution probabilities of a word. The keywords are selected according to their average probabilities. In other words, for each collection, a keyword is ranked higher if its average topic probabilities are higher. The number of latent topics learned from LDA is K which parameter study will be done in our experiment part.

5 Our Hashtag Biased Random Walk Algorithm

5.1 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 4. 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.

Here, we give an example to explain the idea of our proposed approach. As shown in Figure 4, the importance of the word w_1 is not only depended on its connected posts, i.e., p_1 and p_3, but also relied on its co-occured words, i.e., w_3. A random walk is applied on such graph through an iterative score propagation. If a word is with high frequency in important posts, it will be ranked highly. At the same time, if a post is connected by important words, it gets important score. When several words belong to a hashtag, we can let random walk just jump to these words. Thus, a hashtag biased ranking is generated.

5.2 Hashtag Biased Ranking

We rank word nodes in Figure 4 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 3, a modification of Tong et al. [30].

$$\overrightarrow{r} = \alpha \widetilde{Q} \overrightarrow{r} + (1 - \alpha)\overrightarrow{e_h} \tag{3}$$

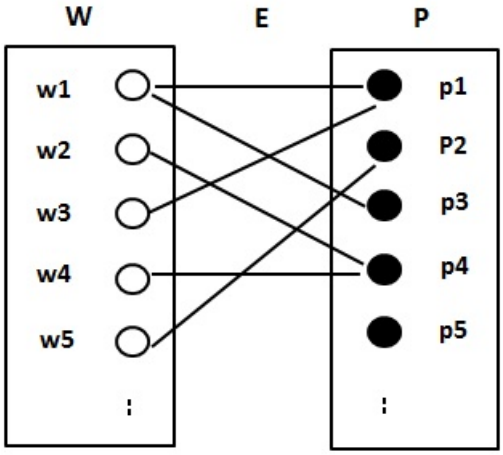

Fig. 4. Word-Post bipartite graph

$\overrightarrow{r}$ 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 [21]. $\overrightarrow{e_h}$ 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 [19, 41] 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 3. The recursive running of Equation 3 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 [24].

5.3 Algorithm Description

Equation 3 defines a linear system problem, where $\overrightarrow{r}$ is determined by:

$$\begin{aligned} \overrightarrow{r} &= (1-\alpha)(I - \alpha\widetilde{Q})^{-1}\overrightarrow{e_h} \\ &= (1-\alpha)Q^{-1}\overrightarrow{e_h} \end{aligned} \tag{4}$$

As dicussed in [30], directly computing Q^{-1} is impractical when the dataset is large, since it requires quadratic space and cubic pre-computation. Linear correlations exist in many real graph, which means that we can approximate $\widetilde{Q}$ by low rank approximation and then compute Q^{-1} efficiently. In this paper, eigen-value decomposition is used after partition the

Fig. 5. Hashtag biased graph ranking

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 [5];	
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 4;	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($6n^3 + 4n^2$)
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($nlogn$)

whole graph into several communities. We provide a sketch of our hashtag biased ranking procedure in the format of pseudo code in Table 5. 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 [43]. 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 [5], each node in graph is in its own set. The connected components are calculated based on the edges, so the disjoint-sets structure is updated when each edge is added into the graph. Readers can refer to [5] 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 [13].

Step 4 and 5 do matrix computation for each partition in $\widetilde{Q_x}$ based on Equation 4. The time complexity of matrix multiplication and matrix subtraction, 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 4. Step 8 is a key process to compute Q^{-1} by combing $\widetilde{Q_x}$ and $\widetilde{Q_y}$. As dicussed in [30], 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:

$$\widetilde{Q} = \widetilde{Q_x} + \widetilde{Q_y} = \widetilde{Q_x} + USV \tag{5}$$

Then the inverse matrix in Equation 4 is computed as:

$$\begin{aligned} Q^{-1} =& (I - \alpha\widetilde{Q})^{-1} \\ =& (I - \alpha\widetilde{Q_x} - \alpha USV)^{-1} \\ =& Q_x^{-1} + \alpha Q_x^{-1} U\widetilde{\Lambda} V Q_x^{-1} \end{aligned} \tag{6}$$

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

$$\widetilde{r} = (1 - \alpha)(Q_x^{-1}\overrightarrow{e_h} + \alpha Q_x^{-1} U\widetilde{\Lambda} V Q_x^{-1}\overrightarrow{e_h}). \tag{7}$$

It can be seen that the approximation of our algorithm comes from the low rank decomposion for $\widetilde{Q}_y$. 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 consisting of hashtag words in microblog posts since hashtag intrinsically represents the key topics of a post. It will help us find good keywords, which is verified by our experimental results.

6 Experiments

6.1 Dataset and Evaluation Methodology

The data set used here was crawled from Sina Weibo [b] from the end of March 2012 to the end of June 2012. There were 74,662 microblog posts in total. 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 n keywords of a discussion group is defined as:

$$Precision@n = \frac{\#important\ keywords}{n}. \tag{8}$$

The measure $Precision@n$ means how many good important keywords our algorithm gives at the top n list. We set n=5 and 10 in our evaluation. We compute the average precision scores

[b] http://www.weibo.com, one of the most popular microblogging platforms in China.

Table 1. Kappa and strength of agreement

Kappa	Strength	Kappa	Strength
0.00	Poor	0.41-0.60	Moderate
0.01-0.20	Slight	0.61-0.80	Substantial
0.21-0.40	Fair	0.81-1.00	Almost perfect

of 14 topic discussion groups from each evaluator and the average value of three evaluators is reported as experimental result.

We treat each topic group as a collection of posts where hashtags are topic related. Our target is to identify top n important keywords from each group. Whether a keyword is important or not in a group is judged by our three laboratory members. The three evaluators worked separately without knowing how our algorithms work. In addition, for each words ranking list to be judged by evaluators, 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.

After collecting evaluators' judgment results, we study the quality of three evaluators. We want to know the variability of evaluator's ratings to measure evaluator disagreement which tells us how evaluators judge individual keywords into the same category (important or not-important) on the measurement scale. The judgments from different evaluators should largely reach a good agreement for a same keyword.

Kappa statistics is one of the most common approaches [28]. Kappa can be thought of as the chance-corrected proportional agreement, and possible values range from +1 (perfect agreement) via 0 (no agreement above that expected by chance) to -1 (complete disagreement). Table 1 provides a rough guide of what is a good agreement. We require the three evaluators to answer questionnaires that supply top 10 extracted keywords per discussion group. Because two evaluators are grouped as a pair to compute a Kappa value, the total number of test pairs is 3 ($C_3^2 = 3$). We collected the relevance judgment results of the 14 topic groups and the average of all Kappa values is 0.532. This value is in the range [0.41,0.6], a moderate agreement in general.

To sum up, in terms of the statements in Table 1, the agreement in our results is moderate at 95% confidence level. The number of the evaluators is not very large, but our Kappa statistics analysis shows that the quality of the evaluators is satisfactory. Thus, their judgments are reliable for evaluation.

6.2 Experimental Results Of Our Topic Model Algorithm

The topic model algorithm can be classified into the clustering based approaches, so the following approaches are compared for effectiveness evaluation. Parameter study is done for K, i.e., the number of clusters or topics.

1. **K-means (TF):** This is of the most popular clustering algorithms in the literature [21]. K represents the number of clusters and should be set in advance and each word is weighted by **TF**.

2. **K-means (TI):** This is similar to **K-means (TF):**, but each word is weighted by **TFIDF**.

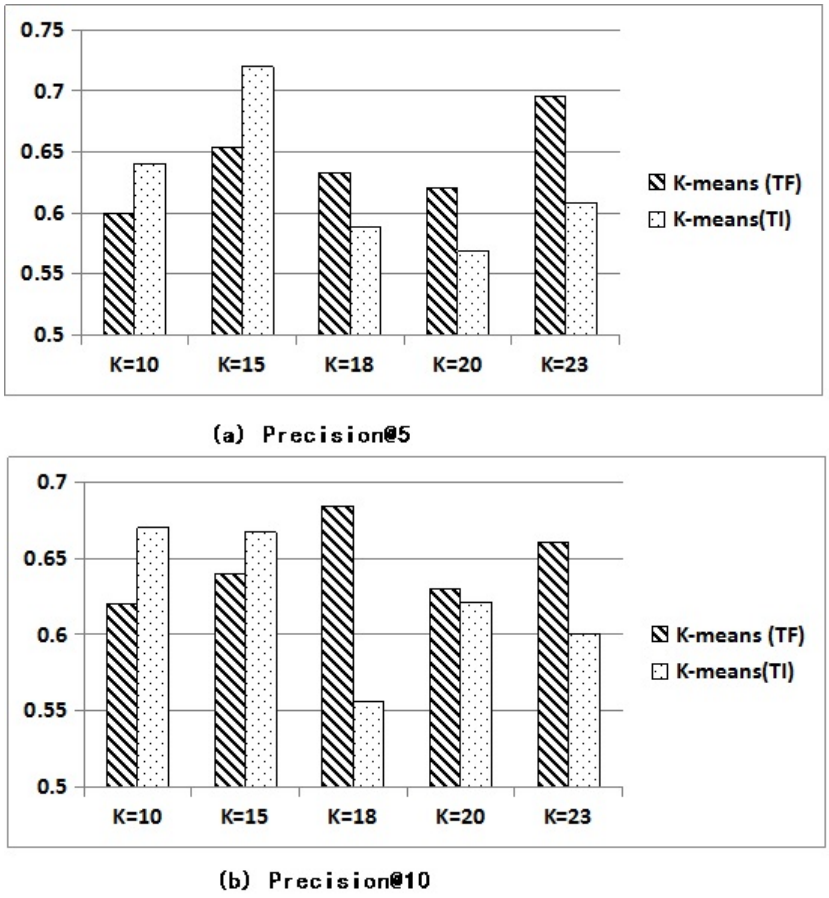

Fig. 6. Comparisons among different K for K-means(TF) and K-means(TI). Note that Y axis represents precision@5 in (a) and precision@10 in (b).

3. **LDA:** This is of the most popular topic model algorithms in the literature [4]. The number of latent topics, i.e., K, should be given before model estimation.

4. **Hash-LDA:** This is our proposed LDA algorithm with hashtag constraints and the number of latent topics (K) should be fixed before modelling.

6.2.1 Results of Different K

We illustrate the performance of the above four approaches at different K in terms of precision@5 and precision@10 in Figure 6 and Figure 7. In Figure 6, the highest precision scores are achieved when K is near the number of topic discussion groups our experimental data have, i.e., 14. For example, in terms of precision@5, K-means(TI) at K=15 shows best. In terms of precision@10, K-means(TF) at K=18 is the winner and a little higher than K-means(TI) at K=15 by 2.6%, i.e., 0.6844 VS. 0.667. Moreover, When K is smaller than 14, K-means(TI) is better than K-means(TF); when K is larger than 14, K-means(TF) is better. In Figure 7, our Hash-LDA performs better than LDA. In terms of precision@10, Hash-LDA

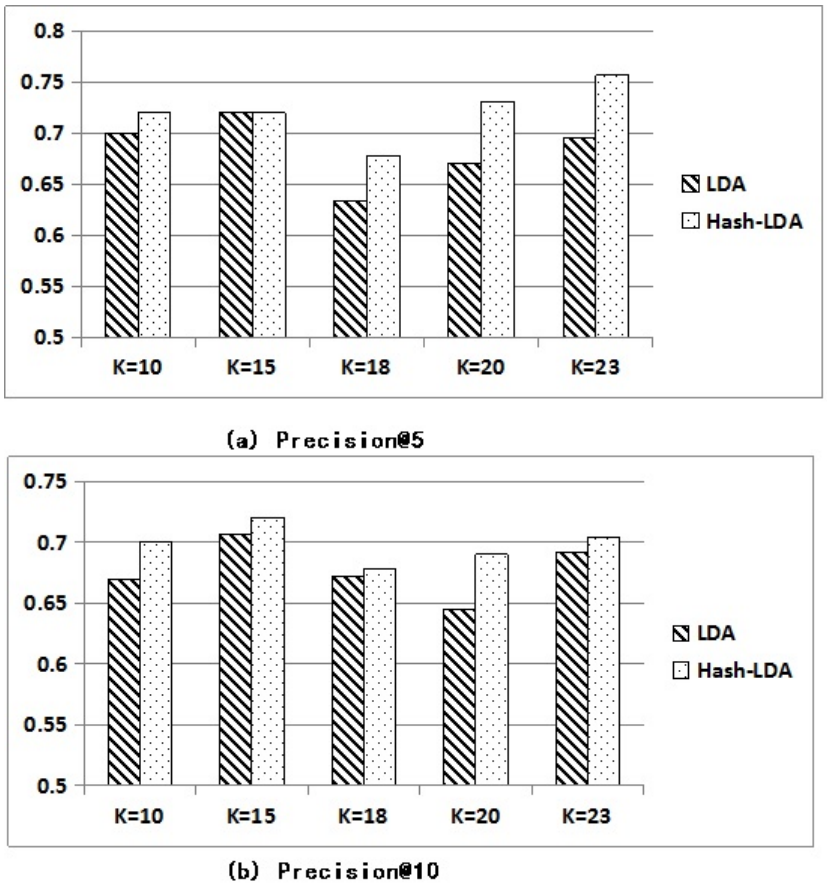

Fig. 7. Comparisons among different K for LDA and Hash-LDA. Note that Y axis represents precision@5 in (a) and precision@10 in (b).

Table 2. The average precision scores of 14 discussion groups

	Precision@5	Precision@10
K-means	0.72	0.6844
LDA	0.72	0.7067
Hash-LDA	**0.7565**	**0.72**

Table 3. The average precisions of LDA using different training data

Precision@n	Authenticated users	Total users
n=10	0.4357	0.4286
n=5	0.3286	0.3214

at K=15 works better than Hash-LDA at K=23 by 2.2%, i.e., 0.72 VS. 0.7043; tn terms of precision@5, Hash-LDA at K=23 achives the highest score, i.e., 0.7565.

From our experimental results as shown in Figure 6 and Figure 7, for K-means based approaches, it is generally acceptable to set K to be near 14 that is the number of topic discussion groups in our data. For LDA based approaches, K had better to be set larger than 14. Our Hash-LDA show best at K=23. We check our experimental data and find that some discussion groups have related topics, such as, ipad sales and new iphone. LDA based algorithms estimate the latent topics in the text collection and can give a cleared topic vision at K >14 by separating co-related topics into more specific sub-topics.

6.2.2 Comparisons Among Different Clustering Approaches

The experimental results are shown in Table 2. The highest precision scores are achieved by our proposed approach (Hash-LDA) in both of Precision@5 and Precision@10. We select two baselines to compare: K-means and LDA. The parameters of each baseline are finely tuned by changing the number of estimated topics or clusters. Besides, we use TF weighting and as the score for each keyword candidate in K-means. We rank each keyword candidate in LDA and our proposed approach by its average topic distribution probabilities. Our proposed approach performs the best among all. For example, in terms of Precision@5, the improvements over the two baselines are 17.74% and 8%, respectively.

6.2.3 Selection of LDA Training Data

We notice that the set of all tweets consist of the tweets from authenticated users and unauthenticated users. It is usual to using all the tweets from both of the two kinds of users as LDA training data. It seems that the set of all the tweets has the larger suggestion context source and the richer content information than the subset from authenticated users or unauthenticated users. Thus, it should output higher precision scores. We try to test this intuitive conclusion. In other words, we want to make it clear about how many blog posts are neccessary to be considered at least to be able to compute LDA model properly and that it is generalizable for other applications. Therefore, the tweets from authenticated users are used LDA training data compared with those from all the users. The results are different to what we intuitively have expected. The average precision value of top 10 keywords extracted from the tweets of authenticated users is slightly higher than that from the tweets of all the users. The average results of all the 14 topics are listed in Table 3. It illustrates that the tweets that from authenticated users could be used for LDA training instead of the whole tweets. As we

Table 4. 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**

know, the information in tweet platform is updated quiet frequently. Even though new topics are emerged, we can use this finding to extract training data directly. Also, it is interesting to study how to carefully select a number of authenticated users for each topic.

During the preprocessing, we observed that under the background of computer configuration with a 32-bit operating system, dual-core CPU and 3.00GB memory, it takes about 4 hours for all users' tweets of a certain topic. But for processing authenticated users' tweets, it just takes about 30 minutes. Using tweets that from authenticated users saves not only the processing time, but also the storage space. How much storage space does it save at all? From experimental data, we find that average Authenticated/Total is about 0.3575. In other words, the authenticated tweet data accounts for around 1/3 in total tweets and almost saves 2/3 storage space.

6.3 Experimental Results Of Our Random Walk based Algorithm

The overall experimental results are show in Table 4. The highest precision scores are achieved by our proposed hashtag biased ranking in both of Precision@5 and Precision@10. We will discuss it according to three aspects, i.e., node types, jumping strategies and weighting strategies and compare the following ranking approaches to show effectiveness.

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 [19, 22, 41].
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.
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.
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.
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.

6.3.1 Comparisons Among Node Types

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 4). 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 graph wins the word to word graph in most cases. Especially, our hash biased ranking shows much better 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 quality of keyword extraction. In other words, important keywords come from important posts with high probability.

6.3.2 Comparisons Among Jumping Strategies

Moreover, hashtag is naturally existed in some posts and it highlights their topic. As defined in Equation 3, 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 8, 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 platform publish posts and like to use hashtag to attract other users' attention. The user-generated hastag is a useful evidence 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 of a ranking list.

6.3.3 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 strategies on keyword extraction, as shown in Figure 9. The left column is TFIDF weighting 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 fewer weights on words with high document (post) frequency. In other words, the extracted keywords should be representative and informative in a post, not commonly appeared in other posts.

6.4 Rank Aggregation of Our LDA and Graph based Algorithms

We have browsed the extracted keywords and find that LDA and random walk based algorithms generate keywords from different viewpoints. The random walk based algorithm

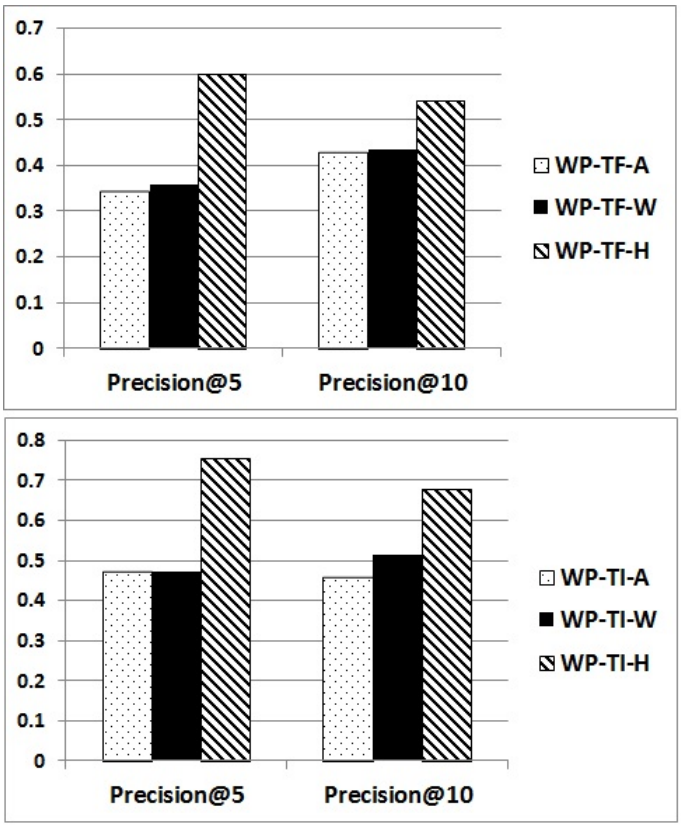

Fig. 8. Comparisons among different jumping strategies. Note that Y axis represents precision values.

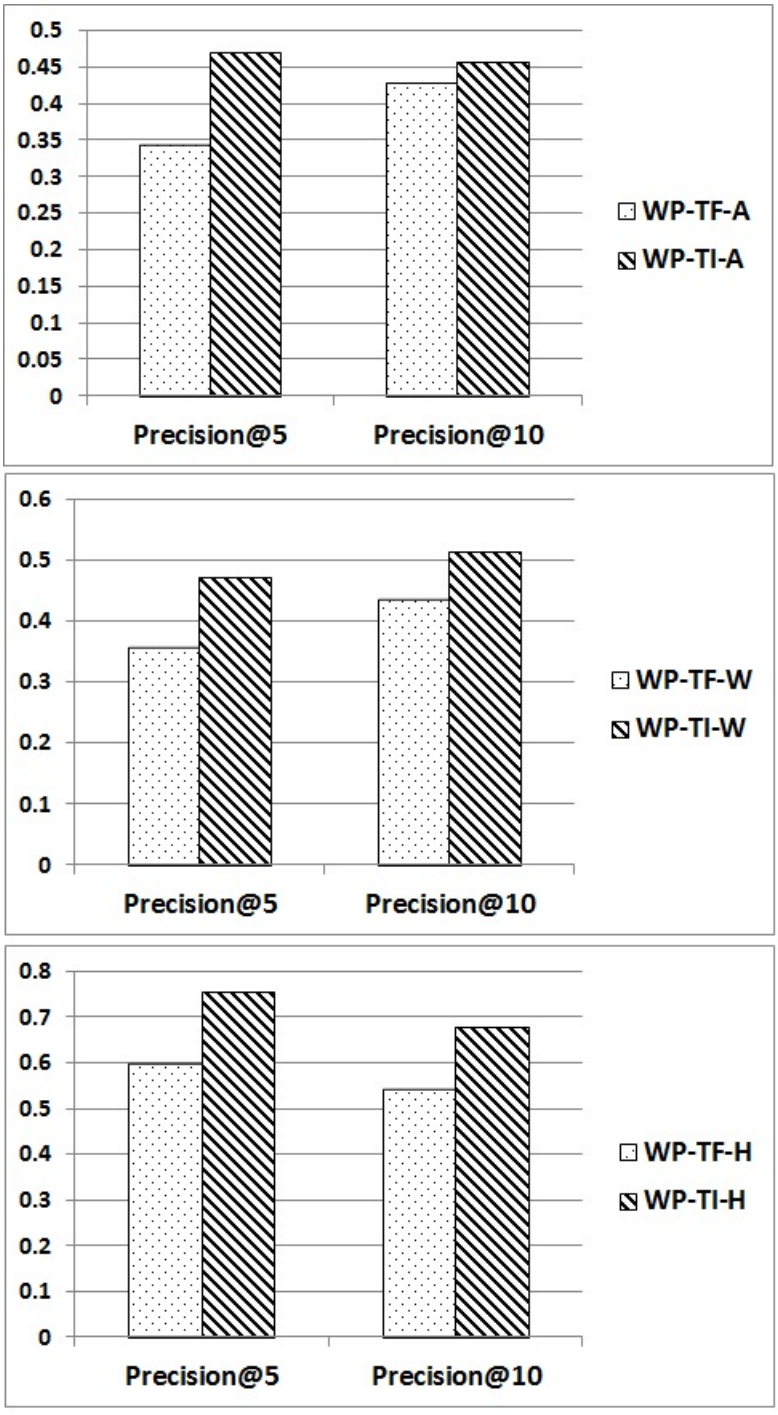

Fig. 9. Comparisons among different weighting strategies. Note that Y axis represents precision values.

Table 5. The rank aggregation of two algorithms

	Precision@5	Precision@10
Hash-LDA	0.7565	0.72
WP-TI-H	0.7571	0.6786
Aggregation	**0.7803**	**0.7374**

computes the importance scores of words according to their *explicit* relationship with posts. Moreover, our hashtag biased ranking give more attentions on words close to hashtag words. In the other hand, the LDA based algorithm assumes that there are *implicitly* latent topics inside texts and then trains a model to group topically related words. Our LDA algorithm with hashtag constraints further infers topic distributions related to hashtags for each word. Therefore, we consider aggregating the keyword lists of our two algorithms by Borda's rule for better performance.

Borda's rule determines the winner of a final list by the number of points corresponding to its position in individual lists. The number of points given to candidates for each ranking (list) is determined by the position of candidates standing in the list. Under the simplest form of the Borda count (the exact form we applied), if there are five candidates in an list then a candidate will receive five points each time they are ranked first, four for being ranked second, and so on, with a candidate receiving 1 point for being ranked last (or left unranked). In other words, where there are n candidates a candidate will receive n points for a first preference, n-1 points for a second preference, n-2 for a third, and so on. The aggregation results are show in Table 5. We can see that our LDA algorithm performs better than our random walk algorithm, i.e., 0.72 VS. 0.6786 at precision@10, which tells us that keyword extraction prefers topical words. Our LDA based algorithm build their models by generating documents through a set of hidden topics and is good to keyword extraction. We get higher precision scores by combing the keyword list of our LDA algorithm with that of our random walk algorithm.

6.5 Case Study

To show the results of our keyword extraction in a visual view, here we give the keywords of two hot social topics in China as examples. We have crawled the Chinese microblog posts with the hashtag "Internet Plus" or "haze" from December 2015 to February 2016. Using our approach, a set of keywords related to the hashtags words are extracted, as illustrated in Figure 10 and Figure 11. Figure 10 gives the keywords related to "Internet Plus" that has draw wide attention in Chinese microblog discussion group. China is developing the "Internet Plus" action plan to integrate mobile Internet, cloud computing, big data and the Internet of Things with modern manufacturing, to encourage the healthy development of e-commerce, industrial networks, and Internet banking. The set of important words in Figure 10 are "Web", "HTML5", "APP", "CRM", "WAP", "marketing", "collaboration", and so on.

The phenomena of "**haze** locks China" is more and more frequent. That's the norm for people in Beijing, Shanghai and many regions in China that are often covered by thick smog and haze. As shown in Figure 11, the most important keywords are "Beijing", "North of China", "Capital", "Shanghai", "PM2.5", "weather", "under control", "air", and so on. This gives us a better understanding on hot topics in current society.

Fig. 10. Here are keywords related to Hahstag "Internet Plus". Note that a larger font size means more important.

Fig. 11. Here are keywords related to Hahstag "haze". Note that a larger font size means more important.

Table 6. The user profiling using SVM for gender prediction

	radio basis	linear
Hash-LDA	0.7329	0.7605
WP-TI-H	0.7028	0.7492
Aggregation	**0.7537**	**0.7727**

6.6 Using Extracted Keywords for Predicting Microblog User Gender

All the above experimental results are judged by human evaluators. In addition, provided agreement statistics among evaluators show a moderate agreement in general. To test the effectiveness of our proposed keyword extraction, we use it in a microblog user profiling task that has data with labels as ground truth. User profile means to explore the network behavior, content preference, content output from microblog users, and then try to predict the attributes of a user, including some statistical attributes, hobby attributes, etc. Here we work on micro-blog text data to predict the gender of a user, male or female, by using SVM classifier.

The data used here is from SMP CUP 2017 [c], a competition launched by Chinese information society of China and Social media processing Specialized Committee. The data set is provided by Sina with 2500 training data and 638 testing data. We try SVM model by two kernel functions, radio basis and linear. The results are shown in Table 6. We can see from the table that different kernel functions perform differently in prediction. Aiming at our classification problem, linear kernel function is more suitable. Hash biased LDA still shows better prediction results than random walk based ranking. The highest accuracy is 0.7717 when using our aggregated keyword extraction.

6.7 Discussions

Based on the above experimental results, we have further discussions on the following two issues. One is about data for training LDA. The other is about data language.

6.7.1 LDA TrainingData

There are various ways for training data selection in real applications. The best one should be rich enough to cover as broad and diverse as possible words, concepts, and topics that are relevant to microblog posts. In the microblogging service platform side, microblog posts have already stored and accessible. The meaning of a post inputted by a user is assumed to be encoded in the whole stored posts. Therefore, if LDA is run at the microblogging service platform side, all the stored posts can be used as training data. There are also domain-specific corpora for domain-specific applications. Moreover, some public universal Web sources are available, such as ODP, Wikipedia, and so on. In this paper, our object is a small part of microblog posts related to IT topic. Those chosen posts as training data have shown improved performance, so we did not use a large-scale Web corpus.

6.7.2 Other Language Data

In this paper, experiments have been conducted on Chinese microblog posts. Our approach make uses of words extracted from natural texts as raw data for further processing, i.e., LDA

[c] https://biendata.com/competition/smpcup2017/

Table 7. An example of review text

userID	productID	score	**text**
A1QA985ULVCQOB	B000GKXY4S	5.0	enjoy scissors inspiration books collage books textures...

Table 8. Top 6 topic words for Amazon data

topic1	glue	tape	book	gun	stick	books
topic2	ink paint	water	color	pen	dye	
topic3	bought	love	daughter	gift	christmas	loves
topic4	table	box	sturdy	plastic	hold	fit
topic5	product	price	quality	amazon	item	purchase
topic6	money	buy	time	don	didn	product
topic7	paper	color	colors	yarn	quality	needles
topic8	machine	sewing	thread	easy	brother	sew
topic9	scissors	cut	punch	cutting	hole	paper
topic10	easy	kit	book	instructions	project	time

and random walk, which can be generalized to other language data. A hashtag is a keyword or a phrase used to describe a topic or a theme. It is not unique to Chinese posts and widely used in microblogging services. Microblog post is a kind of short text, so we have done some preliminary work with Amazon review texts in English.

The data from Amzaon have 35 million reviews and its size is 3.3G. An example of this review data is listed in Table 7. Here, we only use **text** subfield for topic modelling. For a same userID, all his review text are concatenated together as a virtual document, which simulates the collection of microblog posts according a hashtag. We set the number of latent topics to be 10 when applying LDA and show the top 6 words for each topic in Table 8. From Table 8, topic 1 is about stationery and book, while topic 5 is about the quality and price of products. Therefore, our LDA can be applied on English data easily.

7 Conclusions

In this paper, we propose a hashtag enhanced keyword extraction approach for microblog posts by treating hashtags as topical indicators. Our approach is the combination of two hashtag based algorithms. One is a LDA based keyword extraction algorithm with hashtag constraints. As hashtags are user-generated topic words, we treat hashtags as constraints in estimating the latent topics from a collection of microblog posts. The experimental results show that our Hash-LDA algorithm improves the quality of keyword extraction and outperforms traditional K-means and LDA algorithms. The other is a novel word to post bipartite graph based ranking by adopting a hashtag biased random walk. The proposed ranking algorithm can extract important keywords from a collection of microblog posts. Experimental result show that our random walk 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 random walk. The combination of our proposed algorithms shows better performance than each individual one. 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. A. Ahmed and E. P. Xing. Staying informed: Supervised and semi-supervised multi-view topical analysis of ideological perspective. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1140–1150, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
2. 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, New York, NY, USA, 2014. ACM.
3. D. M. Blei and J. D. McAuliffe. Supervised topic models. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, 2007.
4. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
5. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms. McGraw-Hill, 1990.
6. M. Dredze, H. M. Wallach, D. Puller, and F. Pereira. Generating summary keywords for emails using topics. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, pages 199–206, New York, NY, USA, 2008. ACM.
7. D. Gao, W. Li, O. You, and R. Zhang. Lda-based topic formation and topic-sentence reinforcement for graph-based multi-document summarization. In *Information Retrieval Technology, 8th Asia Information Retrieval Societies Conference, AIRS 2012, Tianjin, China, December 17-19, 2012. Proceedings*, pages 376–385, 2012.
8. 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, New York, NY, USA, 2009. ACM.
9. M. Habibi and A. Popescu-Belis. Diverse keyword extraction from conversations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 651–657, 2013.
10. 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.
11. 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, New York, NY, USA, 2014. ACM.
12. 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, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
13. G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48(1):96–129, Jan. 1998.
14. 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.
15. L. Li, L. Qi, F. Deng, S. Xiong, and J. Yuan. Enhancing keyword suggestion of web search by leveraging microblog data. *J. Web Eng.*, 15(3&4):181–202, 2016.
16. L. Li, C. Su, Y. Sun, S. Xiong, and G. Xu. Hashtag biased ranking for keyword extraction from microblog posts. In *Knowledge Science, Engineering and Management - 8th International Conference, KSEM 2015, Chongqing, China, October 28-30, 2015, Proceedings*, pages 348–359, 2015.
17. 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, New York, NY, USA, 2010. ACM.

18. S. Liu, M. X. Zhou, S. Pan, W. Qian, W. Cai, and X. Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 543–552, New York, NY, USA, 2009. ACM.
19. 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.
20. 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, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
21. C. D. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval, 2008.
22. R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In D. Lin and D. Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.
23. R. Nallapati and W. W. Cohen. Link-plsa-lda: A new unsupervised model for topics and influence of blogs. In *Proceedings of the Second International Conference on Weblogs and Social Media, ICWSM 2008, Seattle, Washington, USA, March 30 - April 2, 2008*, 2008.
24. 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.
25. A. Qamra, B. Tseng, and E. Y. Chang. Mining blog stories using community-based and temporal clustering. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 58–67, New York, NY, USA, 2006. ACM.
26. 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, New York, NY, USA, 2013. ACM.
27. J. Radelaar, A. Boor, D. Vandic, J. van Dam, and F. Frasincar. Improving search and exploration in tag spaces using automated tag clustering. *J. Web Eng.*, 13(3&4):277–301, 2014.
28. S. Siegel and N. J. C. Jr. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill Book Co, 1998.
29. Y. Song, S. Pan, S. Liu, M. X. Zhou, and W. Qian. Topic and keyword re-ranking for lda-based topic modeling. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1757–1760, New York, NY, USA, 2009. ACM.
30. 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, Washington, DC, USA, 2006. IEEE Computer Society.
31. P. D. Turney. Learning algorithms for keyphrase extraction. *Inf. Retr.*, 2(4):303–336, May 2000.
32. 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, New York, NY, USA, 2014. ACM.
33. X. Wan and J. Xiao. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 969–976, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.
34. 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.
35. 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, Berlin, Heidelberg, 2007. Springer-Verlag.
36. 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 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part II*, pages 608–619. Springer, 2014.

37. 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, New York, NY, USA, 2012. ACM.
38. 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, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
39. T. Yano, W. W. Cohen, and N. A. Smith. Predicting response to political blog posts with topic models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 477–485, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
40. 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.
41. 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, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
42. 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.
43. D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.