Topic-level Graph Modeling of Microblogging Information Diffusion for Detecting Topical Keyphrases

Shuangyong Song[1], Yao Meng[2], Qiudan Li[3] and Haiqing Chen[1]
[1] Intelligence Innovation Centre, Alibaba Group.
No. 969, West Wenyi Road, Yuhang District, Hangzhou 311121, Zhejiang Province, China
[2] Information Technology Laboratory, Fujitsu R&D Center Co., Ltd.
Pacific Century Place, No.2A Gong Ti Bei Lu, Chaoyang District, Beijing 100027, China
[3] The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences.
No. 95, Zhongguancun Dong Road, Beijing 100190, China
[1] {shuangyong.ssy; haiqing.chenhq}@alibaba-inc.com, [2] mengyao@cn.fujitsu.com,
[3] qiudan.li@ia.ac.cn

Abstract

The rapid increasing popularity of microblogging has made it an important information seeking channel. Keyphrase extraction is an effective way for summarizing and analyzing microblogging content, which can help users gain insights into internet hotspots. Existing methods usually detect microblogging keyphrases based on their occurrence frequency or the graph modeling of their semantic relationship. However, those methods may neglect some important factors. Generally, phrases shown in influential users' microblogs are more likely to attract other users' interest. Besides, some phrases usually have similar diffusion paths and attract the attention of same population.

In this paper, we propose a novel model for detecting topical keyphrases in microblogging, which is a modified version of the general keyphrase detection models. Our proposed model considers all the above factors comprehensively. First, it detects high frequency term from abundant micro-blogs as topical candidate keyphrases, then constructs a relation graph about them with user interest and user following web for each topic. Finally, we rank those topical candidates with graph models for realizing topical keyphrases detection. Experiments show this model is highly effective for topical keyphrase extraction in microblogging.

Keywords
Microblogging, topical keyphrase extraction, user interest, user influence, graph modeling, information diffusion.

1. Introduction

Recent years have seen a rapid growth in microblogging and the rise of popular microblogging services such as Twitter. Microblogging has becoming an important source of up-to-date topics about what's happening in the world. Accordingly extraction of microblogging keyphrases, which mean words or word-groups that have high degree of diffusion and influence in a special period, becomes a useful work in this novel social network platform, which can provide users with a fast and convenient way to gain insights into internet hotspots. In microblogging, the popularity of information is impacted by not only the frequency of it but also the influential of the web users who have published microblogs about it, since information shown in the influential users' microblogs are more likely to attract other users' interest. Special unidirectional linking type help some microblogging users getting millions of followers, which makes them have important impaction of information diffusion in microblogging, and get more trust from others than users with few followers (Cha et al. 2010). Therefore, user influence should be taken as an important factor in microblogging keyphrase detection.

Existing methods usually detect microblogging keyphrases based on their occurrence frequency or the graph modelling of their semantic relationship, which are similar with the topical keyphrase ranking methods for traditional social media such as web news. Zhao et al. (2011) proposed a context-sensitive topical PageRank method for keyword ranking and a probabilistic scoring function that considers both relevance and interestingness of keyphrases for keyphrase ranking. Bellaachia and Al-Dhelaan (2012) proposed a novel unsupervised graph-based keyword ranking method, called NE-Rank, which considers word weights in addition to edge weights when calculating the ranking, and then they also proposed a method called HG-RANK (Bellaachia and Al-Dhelaan, 2014), with using more semantic features to detect keyphrases. However, all of the above mentioned methods may neglect the relationships between different phrases and the importance of user influence to further information diffusion, which are important factors in microblogging like social media.

Generally, phrases shown in influential users' microblogs are more likely to attract other

users' interest, making them more widely diffused in the near future. Besides, phrases may have relations with each other, and some phrases usually have similar diffusion paths and attract the attention of same population. For considering those factors, Song et al. (2014) proposed a novel graph model based approach for extracting general keyphrases from abundant recent microblogs, while the users' following behaviours are utilized to create link graph of candidate keyphrases. In this paper, we modify the above model into a topical keyphrase detection model, since the topical model can summarize and analyse microblogging content more detailedly (Zhao et al. 2011).

Our proposed model first detect high-frequency terms from abundant microblogs as candidate keyphrases, and perform a phrase-word topic model to discover topics. Then for each topic, we construct a relation graph about topic related candidates with the user interest and their following web. Finally we rank those candidates on each topic with graph models for realizing topical keyphrases detection and hot phrase recommendation in microblogging. The rest of this paper is organized as follows: In section 2, we provide a brief review of the related work. The introduction of our method is proposed in section 3. In section 4, some analysis of our experimental results is given. Finally we make a conclusion and discuss our plans for future work in section 5.

2. Related Work

Keyphrase extraction methods can be roughly categorized into either unsupervised or supervised (Ding et al. 2011). In this paper, we focus on the brief introduction of the unsupervised methods. Compared to supervised keyphrase extraction methods, unsupervised ones eliminate the need of training data, like our proposed model. Unsupervised methods rely solely on a collection of plain non-annotated textual data. These approaches usually select quite general sets of candidates, and use a ranking function to limit the selection to the most important candidates.

Mori et al. proposed one approach to obtain a set of terms representing a given relation by extracting terms that co-occur with an entity on the Web (Mori et al. 2007). The basic idea of their work is inputting a term to a search engine for extracting terms that co-occur frequently with the given term as related terms. Barker and Cornacchia restrict candidates to noun phrases, and rank them using heuristics based on the number of words and the frequency of a noun phrase, as well as the frequency of the head noun (Barker and Cornacchia, 2000). Ding et al. proposed a novel formulation, which presents several criteria of high quality news keyphrase and integrate those criteria into the keyphrase extraction task by converting the task to the binary integer programming problem (Ding et al. 2011).

Wan and Xiao proposed a novel single document keyphrase extraction model with utilization of neighbourhood documents knowledge instead of just utilizing the information contained in the specified document (Wan and Xiao, 2008).

In unsupervised keyphrase extraction, graph models are the state of the arts. Zhao et al. first identify topics from Twitter collection using a topic model, then for each topic, they run a topical PageRank algorithm (Brin and Page, 1998) to rank keywords and generate candidate keyphrases using the top ranked keywords. Finally, they use a probabilistic model to rank the candidate keyphrases (Zhao et al. 2011). Bellaachia and Al-Dhelaan proposed a novel unsupervised graph-based keyword ranking method, called NE-Rank, that considers word weights in addition to edge weights when calculating the ranking (Bellaachia and Al-Dhelaan, 2012). The above mentioned two graph model based keyphrase extraction methods may ignore the information diffusion path through micro-blogging users' following relationships and the importance of influential users' impact on other users' interest, since they just utilize content-based factors to create graph of candidate keyphrases. Song et al. proposed a topic graph construction method with considering micro-blogging users' following relationships (Song et al. 2014), which can well reflects users' real interest. In our proposed model, we utilize this method to construct graph of candidate keyphrases, and rank them with graph-based ranking method.

Besides, our work is also related to automatic topic labelling (Mei et al. 2007; Song and Meng, 2015b) and automatic micro-blogging user tag extraction (Wu et al. 2010; Song and Meng, 2015a). However, we focus our work on extracting topical keyphrases in microblogs, which has its own challenges. Our method can also be used to label topics in other text collections and extract personal tags of web users, products and other kinds of items automatically on micro-blogging like social networks with considering users' interest links.

3. Proposed Model for Microblogging Topical Keyphrase Detection

3.1. System Architecture of the Proposed Model

In this subsection, we present the architectural design of our proposed microblogging user summarization model. The system architecture of the proposed model is shown in Figure 1, which consists of four functional modules, namely, candidate keyphrase detection, phrase-level topic discovery, topical phrase linking web construction, and topical candidate keyphrases ranking.

In candidate keyphrase detection module, we utilize the length and frequency of noun phrases as two factors to detect candidate keyphrases, and a 'substring problem' between

phrases is also addressed. In phrase-level topic discovery module, detected candidate keyphrases will be mixed with other contextual words, and then we try to discover semantic topics from those phrases and words with topic model. In topical phrase linking web construction module, we detect implicit interest linking between candidate keyphrases with users' following links, and then build up the relational network graph of topical candidate keyphrases. In this graph, each candidate is taken as a node, and each link between two candidates represents an interest relationship and information diffusion between users who respectively published those two candidates. In topical candidate keyphrases ranking module, for each topical candidate keyphrase linking graph, we apply *PageRank* method to measure the popularity of each candidate by calculating its authority score in the graph, and top ranking candidates will be detected as final topical keyphrases. The mechanism of each functional module in our proposed model will be discussed detailedly in the following subsections.

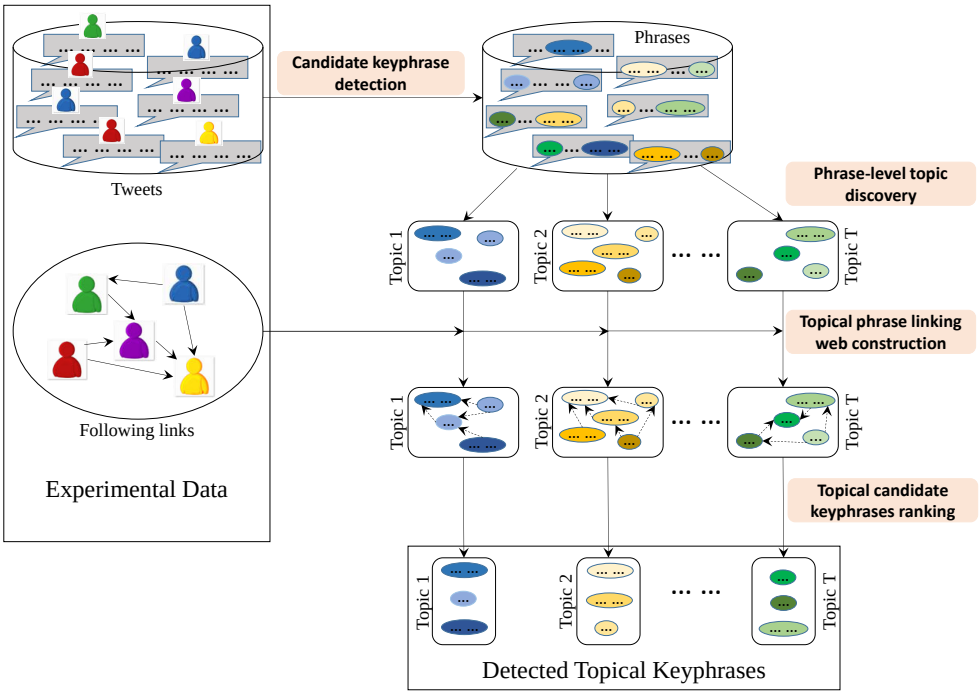

Figure 1: System architecture of the proposed model.

3.2. Candidate Keyphrase Detection

In the processing of candidate keyphrase extraction, keyphrases are always single word nouns or bigram nouns empirically with lots of experiments (Paukkeri et al.

2008). In this paper, besides all single word nouns and two-gram nouns, we also extract trigram and four-gram nouns from part-of-speech results, and calculate statistics of them for ranking with frequency. Some examples of phrases in real Twitter microblogs are given in figure 2. Then a 'substring problem' is addressed and we process this problem with a simple method in (Nakagawa and Mori, 2002): in formula (1), T_{length} is the length of a phrase, which is the number of characters in it, and $T_{frequency}$ is the frequency of this phrase. T_{value} means the 'value for keeping' of a phrase, which is determined by the above two factors T_{length} and $T_{frequency}$:

$$T_{value} = T_{frequency} * T_{length} \tag{1}$$

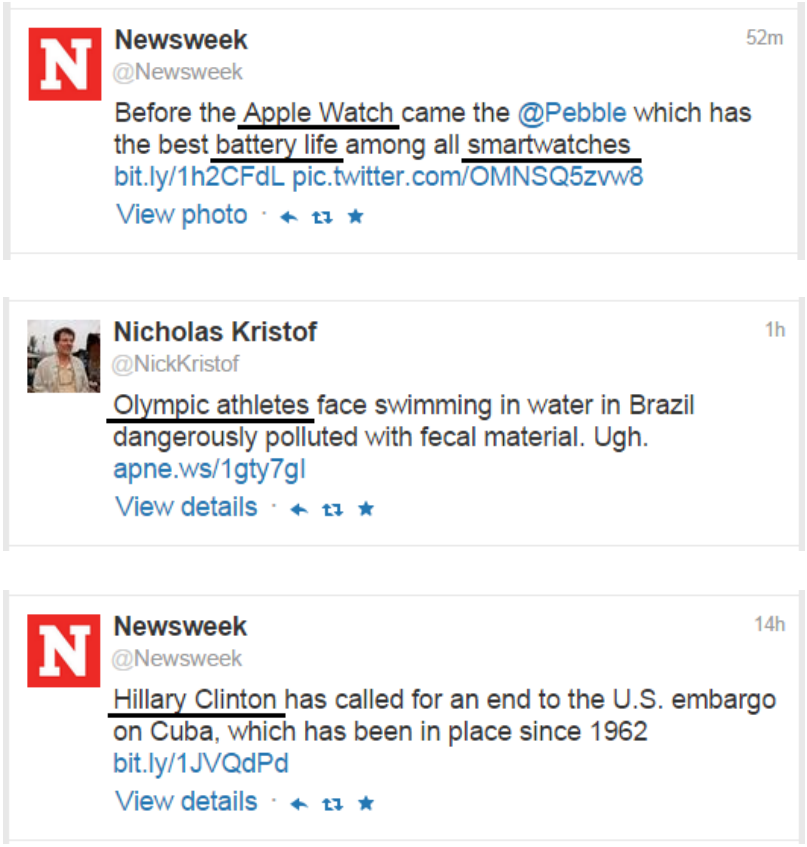

Figure 2: Examples of (key)phrases in real Twitter microblogs.

If a 'substring noun' has a larger T_{value} than the longer noun which contains it, we keep the substring noun. Otherwise, we keep the longer noun and delete the substring

noun. Finally, all the saved phrases whose frequencies are larger than a threshold number will be taken as candidate keyphrases, which in this paper is set to be 5 empirically.

3.3. Phrase-level Topic Discovery

After the detection of candidate keyphrases, we design a 'phrase-word topic model' to discover topics. Before that, we should do some preprocessing on the data. The focus of our study is English keyphrases, and the microblogs matched by our regular expressions will tend to be written in English. We therefore identify and remove non-English microblogs. The short *url*s shows in microblogs are to be removed since they will bring problems for candidate keyphrase extraction. For reducing the ambiguity and calculating pressure of topic model, we delete stopwords, punctuation marks and emoticons, which are not used in the detected phrases. The reason for us to do those preprocessing after the candidate keyphrase detection is because some phrases may contain those stopwords, punctuation marks and emoticons. For example, a candidate keyphrase "internet of things" contains the stopword 'of'.

Our 'phrase-word topic model' is based on the following assumptions: all the candidate keyphrases are taken as linguistic units mixed with other single words, and the topics are represented by the distribution on those candidate keyphrases and single words. Besides, each user has her topic interests modelled by a distribution over the topics. In this paper, we utilize *JGibbLDA* version topic model (Phan et al. 2008) to realize this modelling step. We empirically set the number of topics to be 30, and set the max iteration of Gibbs sampling to be 500, based on the discussion in paper of Zhao et al. (2011).

3.4. Topical Phrase Linking Web Construction

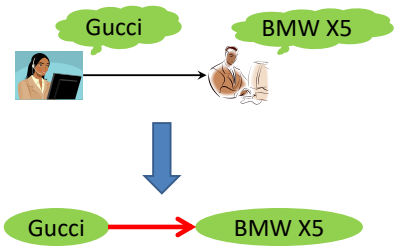

Figure 3: Creating links between candidate keyphrases with users' following actions.

For each topic, we construct the topical phrase linking web with the method in (Song et al. 2014). For constructing phrase links with user following relationship, we first need

to detect interest of users for the candidate keyphrases. We count the frequency of each candidate keyphrases existing in a user's microblogs within a week, and then up to three most frequency candidates are confirmed as his favourite ones, empirically. After this, links between candidates in each topic can be constructed. We give an example in Figure 3 to explain this: if following relation exists between two users, and the follower is interest in Gucci while the being-followed user is interested in BMW X5, we create a link from Gucci to BMW X5, which means that 'a user is interested in Gucci, and she also is interested in BMW X5'. Meanwhile, this link also shows the diffusion path of 'BMW X5' in microblogging, which gives a vivid description of the important impact of influential users for information diffusion, since the phrases shown in influential users' microblogs can get many in-links.

Weights of link between candidate keyphrases are calculated with the weight of following link between users. In the following link of users, weight of those link are not all equal. In this paper, we get the weight of users' following links by considering the being-followed users' influence, which is determined with three factors: follower number, retweeted number and mentioned number (Cha et al. 2010). Those three factors can all reflect a user's influence in microblogging platform. Detailedly, follower number is the number of a user's followers, retweeted number is the number of retweet action for a user's microblogs, and mentioned number is the how many times a user has been mentioned in others' microblogs. Different from (Cha et al. 2010), we consider those three factors together to get a user's topical influence, and the formula is given below:

$$A(u) = \alpha \frac{N_F(u)}{Max(N_F)} + \beta \frac{N_R(u)}{Max(N_R)} + \gamma \frac{N_M(u)}{Max(N_M)}, (\alpha + \beta + \gamma = 1) \tag{2}$$

where $A(u)$ means the topical influence of user u. $N_F(u)$ is number of u's followers, and $Max(N_F)$ is the largest $N_F(u)$ of all users. $N_R(u)$ is the number of retweet for u in recent three days, and $Max(N_R)$ is the largest $N_R(u)$ of all users. $N_M(u)$ is the mentioned number of u in recent three days, and $Max(N_M)$ is the largest $N_M(u)$ of all users. Parameters α, β, γ are used to adjust the importance of the three mentioned factors, and $\alpha + \beta + \gamma = 1$. The setting of those three factors will be shown in the experiments section.

Since the link between candidates can be created from more than one links between users, we calculate weight of links between candidates with formula below:

$$W_{P-link} = \sum W_{U-link} = \sum A(u) \tag{3}$$

where $W_{P\text{-}link}$ means the weight of a link between candidate keyphrases, and $W_{U\text{-}link}$ means the weight of links between users which are used to create *P-link*. $W_{U\text{-}link}$ between two users is equal to $A(u)$ of the being-followed user, as mentioned before.

3.5. Topical Candidate Keyphrases Ranking

Based on topical phrase linking web mentioned in above section, *PageRank* algorithm (Brin and Page, 1998) is applied to measure the popularity of candidate keyphrases with Popularity Score (P_{score}). The in-links and out-links of each candidate can be detected by the creation of the relationships among them. The Popularity Score of each candidate is calculated by the equation given below:

$$PopularityScore(x_i) = \frac{1-d}{N} + d \times \sum_{x_j \in IN(x_i)} \frac{PopularityScore(x_j) * W_{ji}}{O(x_j)} \tag{4}$$

where $PopularityScore(x_i)$ is the Popularity Score of candidate x_i , $PopularityScore(x_j)$ is the Popularity Score of candidate x_j that links to x_i , W_{ji} is the weight value W_{P-link} of edge from x_j to x_i, N is the total number of the candidates, $IN(x_i)$ is the set of candidates that link to x_i, $O(x_j)$ is the number of out-links of x_j, 'd' is a damping factor, ranging between 0 and 1. In this work, we still use the damping factor of 0.85, which was used in the original PageRank paper (Brin and Page, 1998). Initially, the Popularity Score for each candidate is set to 1, and the computation ends when for some small ε:

$$\left| [PopularityScore(x_i, \mathrm{r})] - [PopularityScore(x_i, \mathrm{r}-1)] \right| < \varepsilon \tag{5}$$

i.e., when convergence is assumed. In Eq. (4), $PopularityScore(x_i, \mathrm{r})$ means the $PopularityScore(x_i)$ in the r_{th} iteration, where $1 \le r \le R$, and R is the maximal iteration number set up in experiments.

In this paper, for realizing a simpler calculation instead of the loop iteration, we formulate *PageRank* algorithm with the Markov chain model (Steward, 1994). We treat the candidate graph G as a Markov chain, each candidate as a state and each link as a transition from one state to another. Considering the adjacency relation among the states in the Markov chain, we use *N*N* adjacency matrix ***M*** to denote the transition probability matrix of the chain. Using $\sum w_{i*}$ to represent the summation of weight of links from candidate x_i to other candidates, we define each cell of ***M*** as the following.

$$M_{ij} = \begin{cases} \dfrac{W_{ij}}{\sum W_{i*}} & if\ (i,j) \in E \\ 0 & otherwise \end{cases} \tag{6}$$

Candidates which have no links with others were deleted, because it means that authors of the microblogs about those candidates has no links with other users, so it has little influence in the whole social network (Wasserman and Faust, 1994). Overall, the matrix ***M*** is a transition probability, i.e., column-stochastic with no columns consisting of just zeroes. If *P* is a column vector, and it can meet the following equation:

$$P = \hat{M}P \qquad (\hat{M} = dM^T + \frac{1-d}{N}E) \tag{7}$$

where *P*, the principal eigenvector of $\hat{M}$ with eigenvalue of 1, is in fact the P_{score} column vector containing the P_{score} of all the candidates (Li et al. 2008). Thus, after the calculation of *P*, we can easily get the popularity ranking of candidates, and top ranking candidates in each topic will be detected as the final topical keyphrases.

4. Experiments

4.1. Experimental Data and Pre-processing

A dataset used in our experiments is constructed by data from Twitter, which is a subset of the dataset collected by Choudhury et al. (2010). The chosen one week dataset contains 510,761 microblogs published by 54,307 users from Sep 07 to Sep 13, 2009, and following links between those users are downloaded. Besides, we also use a dataset from another microblogging website FriendFeed in the same one week period, which is a subset of the dataset collected by Celli et al. (2010). The dataset contains 4,114,263 microblogs published by 88,444 users, and also the following links. A statistic description of the dataset is shown in Fig. 4. The upper plot of figure 4 shows that a

large number of users tag only a few microblogs in Twitter, while a few users have published an amount of microblogs. The bottom plot of Fig. 4 shows that in FriendFeed the situation is same with that in Twitter.

After the preprocessing steps mentioned in above section, we got the final datasets with 456,324 microblogs from Twitter and 2,721,494 microblogs from FriendFeed.

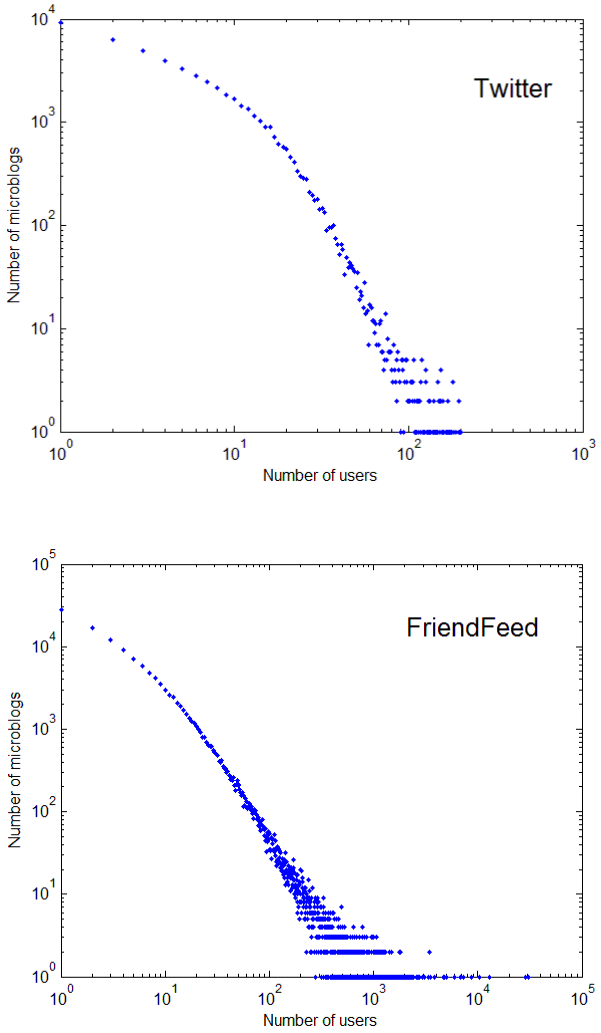

Figure 4: Upper: the log–log plot of the number of users to the number of microblogs (Twitter). Bottom: the log–log plot of the number of users to the number of microblogs (FriendFeed).

4.2. Baseline Methods

In this subsection, to evaluate the performance of the proposed model, we compare it with some baseline methods. Besides the semantic web (***S-web***) based method in (Zhao et al. 2011), we choose three other methods as baselines:

• Term frequency ranking model (***T-fre***): Hussey et al. (2012) give a comparison of term frequency based methods on the task of keyphrase detection, and on all given datasets *T-fre* algorithm had the highest percentage. Therefore, we choose *T-fre* as one comparative method.

• User frequency ranking model (***U-fre***): we design a user frequency model as another baseline. *U-fre* consider the number of users who have published microblogs about a candidate keyphrase as the ranking factor for detecting final keyphrases.

• Retweet graph ranking model (***R-gra***): Yang et al. (2012) take the retweet number of microblogs as their most important evidence on popularity, and propose a retweet graph analysis method for finding interesting microblogs. In this paper, we take it as a baseline method.

4.3. Gold Standard Generation

Since users always just care about the top rank keyphrases, we just evaluate top 20 results of each model on each topic. For each dataset, we put results of each model in the same day together with deduping, and finally we get an approximately average 35 keyphrase results in each day on each topic. Then, a list with random order of them will be provided to tagging volunteers.

Three graduate students were recruited to annotate the candidate keyphrases. Each annotator should give each candidate keyphrase an integral ‘Keyphrase Probability’ from 1 to 5. In the process of scoring, annotators can search the information related to candidate keyphrases from web search engine, or asking friends around to get their opinions, or just considering own understanding and interest on those candidates. We then used the average value as the final score. Fleiss' Kappa is adopted to verify the degree of agreement among the three annotators, which is 0.73, indicating substantial agreements.

4.4. Evaluation Metrics

We measure the effects of our model using Normalized Discounted Cumulative Gain at top k (*NDCG@k*) and Mean Average Precision (*MAP*), which are both fit for evaluating top results sensitive ranking problems such as web search, keyphrase detection and information recommendation, etc. The Discounted Cumulative Gain at

top *k* (*DCG@k*) for keyphrase detection task is defined as:

$$DCG@k(y) = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log(1+i)} \tag{8}$$

where *y* means candidate keyphrase ranking list in our experiments, and rel_i means real keyphrase score of the i^{th} result in our result. *NDCG@k* is accordingly defined as:

$$NDCG@k(y) = \frac{DCG@k(y)}{DCG@k(y^*)} \tag{9}$$

where *y** is a perfect ranking result which corresponds to any perfect ordering based on the manual tagging scores.

MAP is defined as the average precision on each day a real keyphrase is detected, which is given in Equation (8):

$$MAP = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{j=1}^{i} r(j)}{i} * r(i) \tag{10}$$

where *i* is the position of the keyphrase in ranking list and we want to evaluate top *N* results. The *r*(*i*) is a binary value: when the suggested keyphrase at position *i* is one of the top *N* ones, *r*(*i*) is set to be 1 and 0 otherwise.

In this paper, we set the *k* in *NDCG@k* to be 5, 10, 15 and 20 respectively, and the *N* in *MAP* is set to be 20 according to the generating method of gold standard. After obtaining the *NDCG@k* and *MAP* of experimental results about topics on each day, we further calculate the average value as our final evaluation metrics.

4.5. Setting of Topical Influence Parameters

In the subsection 3.4, three parameters α, β and γ are used for adjusting the significance of three factors in formula (2): follower number, retweeted number and mentioned number. In our previous work (Song et al. 2014), we empirically set $\alpha = \beta = \gamma = 1/3$ for balancing the importance of the three factors. Actually, this setting method is without enough consideration. In this paper, for choosing the most effective α, β and γ in formula (2), we design a setting experiment of those three parameters, with the evaluation metric *MAP* of top 20 keyphrases detected by our model in each topic. Since running the program of our model once needs about a dozen minutes, we cannot test all possible combinations of α, β and γ. Therefore, we firstly set α = 1, and test the performance of β and γ as different values of 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, which totally contains 400 different combinations of β and γ. The *MAP* results of those different combinations of β and γ (while α = 1) for our

proposed model on topical keyphrases detection is given in figure 5, and the maximum *MAP* value 0.882 shows when $\beta = 4$ and $\gamma = 0.3$. Finally, with the restriction of $\alpha + \beta + \gamma = 1$, we normalize the values of them ($\alpha = 1$, $\beta = 4$ and $\gamma = 0.3$) as $\alpha = 0.19$, $\beta = 0.75$ and $\gamma = 0.06$.

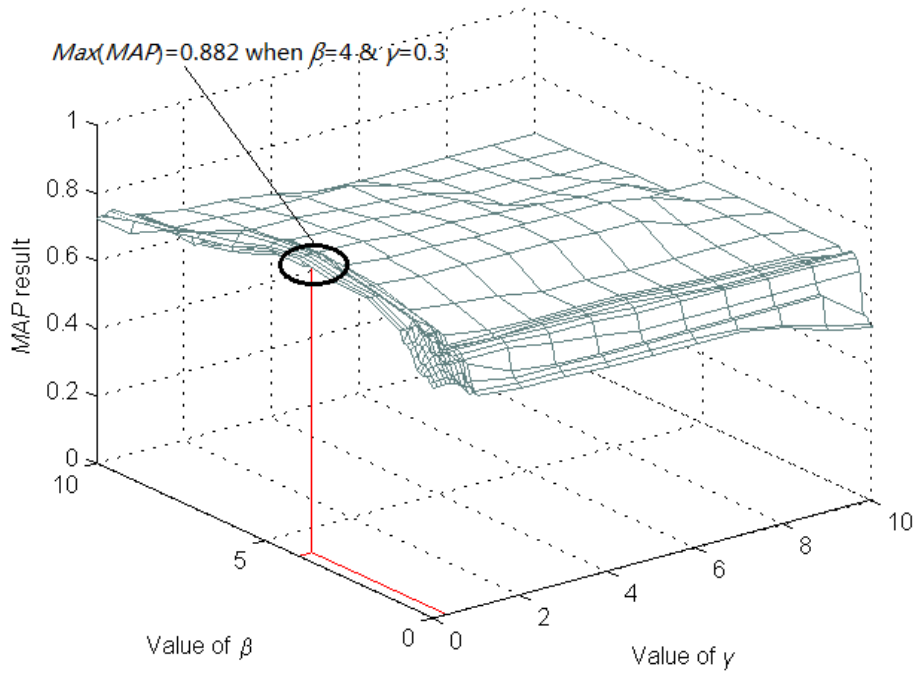

Figure 5: The *MAP* (Mean Average Precision) results of different values of β and γ (while α = 1) for our proposed model on topical keyphrases detection

4.6. Experimental Evaluations

Figure 6 demonstrates the *NDCG@k* and *MAP* evaluation results of the five models on experimental datasets, from which we can see our model, *R-gra*, *U-fre* and *S-web* all outperform *T-fre* model. This is because *T-fre* model neither consider user interest factor nor semantic relation among phrases. In *T-fre* model, phrases such as 'social media', 'blog post' and 'real estate' are usually detected as keyphrases in many days, which indeed represent the character of microblogging that it is a platform for users to discuss daily life. However, for providing users microblogging keyphrases in each day, some phrases with high timeliness should be given more attention. For example, on 2009-09-10, 'health care speech' and 'president Obama' are given highest ranks in a

same topic by other models, which refer to the news "As the debate on health care in the U.S. continues, President Obama detailed his vision for health insurance reform in his second address to a joint session of Congress on Wednesday September 9, 2009.", and this event caused a hot discussion on the following day.

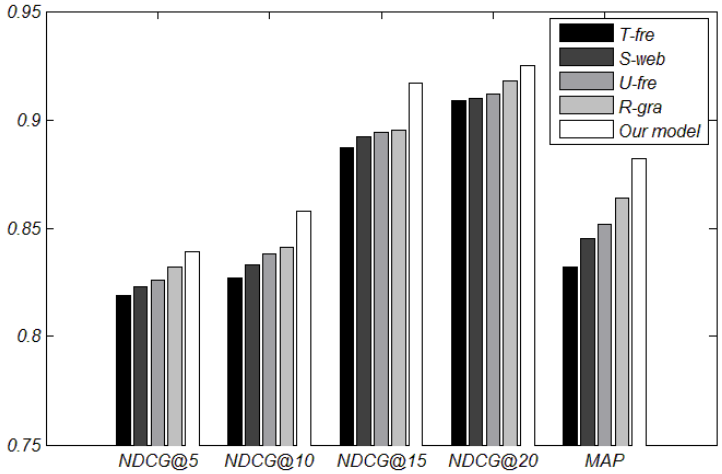

Figure 6: Result comparison with *NDCG@k* and *MAP*

Besides, *S-web* model performs better than *T-fre* model, because some phrases with frequent but meaningless words can be removed with *PageRank* in *S-web* model. *S-web* model can perform well on topical keyphrase detection task since topic model can help ranking topic related words and choosing meaningful words on topical keyphrase detection task. Furthermore, *U-fre* and *R-gra* also perform better than *T-fre* since the user interest factor and user retweet factor can both partly reflect the impact of influential users on information diffusion. Our model performs better than *S-web*, *R-gra* and *U-fre* models, because our model highlights influential users' impact on information diffusion, and those influential users include movie stars, politicians and news media. etc., and many intraday popular information are derived from their microblogs. In particular, our model just show minor improvement over *R-gra*, this result indicates that user retweet actions are very important and veritable for reflecting users' interest, so the *R-gra* can perform so well based on just one retweeting factor.

We further discuss the result consistency of those five models on Twitter data and FriendFeed data. A set of top 20 keyphrases detected by model i ($1 \leq i \leq 5$) on Twitter

data is defined as S_{Ti} and same meaning on FriendFeed data is defined as S_{Fi}, then we calculate $|S_{Ti} \cap S_{Fi}|/20$ on each topic and each day in our experimental dataset, and the mean value of them is taken as the result consistency of model *i* on two different experimental datasets. However, not all topics on two datasets have same meanings with each other, so we choose several general topics such as 'food', 'sports', 'entertainment' to compare the model results. Finally, we get the result consistency of *T-fre*, *S-web*, *U-fre* and *R-gra* and our model as 0.8914, 0.8886, 0.8600, 0.8857 and 0.8743 respectively, which shows that many top rank keyphrases are same on different web platform, and users on different social network websites have similar interest. For example, top keyphrases about *sports* topic on Sunday September 13, 2009 on both datasets all contain 'LeBron James', 'Championship', 'Football game', 'Premier league', '2009 NBA Hall', 'NFL' and 'Lakers'.

5. Conclusion and Future Work

In this paper, we propose a new model for the problem of topical keyphrase extraction in microblogging, which is an important way for summarizing and analysing microblogging content. The proposed model first extract multi-gram nouns with high frequency as candidate keyphrases, and then consider the user influence and user interest relationship as key factors for ranking those candidates with graph model. In our experiments, this method is shown to be very effective to boost the performance of topical keyphrase extraction in microblogging.

In our future work, for detecting the evolution of popular topical information which is due to the changing of microblogging users' interest, we plan to propose a time-sensitive model for phrase ranking not only in microblogging but also in other kinds of social media, such as social networking sites and Vertical BBS. Besides, the user-oriented phrase ranking based on users' interest are also part of our future work.

6. References

K. Barker, N. Cornacchia, 2000. Using Noun Phrase Heads to Extract Document Keyphrases, In *Proceedings of the 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, 40-52.

A. Bellaachia, and M. Al-Dhelaan, NE-Rank: A Novel Graph-Based Keyphrase Extraction in Twitter. 2012. In *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, pp. 372-379.

A. Bellaachia, and M. Al-Dhelaan, HG-RANK: A Hypergraph-based Keyphrase Extraction

for Short Documents in Dynamic Genre. In *Microposts*, 2014, pp. 42-49.

S. Brin, and L. Page, 1998. The anatomy of a large-scale hypertextual Web search en-gine. In *Proceedings of the Seventh International Conference on World Wide Web*, pp. 107-117.

F. Celli, F. M. L. Di Lascio, M. Magnani, B. Pacelli, and L. Rossi, 2010. Social network data and practices: the case of FriendFeed. In *Proceedings of the third International Conference on Social Computing, Behavioral Modeling and Prediction*, pp. 346-353.

M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, 2010. Measuring user influence in Twitter: the million follower fallacy. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, pp. 10–17.

M. D. Choudhury, Y-R. Lin, and H. Sundaram, 2010. How Does the Data Sampling Strategy Impact the Discovery of Information Diffusion in Social Media? In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, pp. 34-41.

Z. Ding, Q. Zhang, and X. Huang, 2011. Keyphrase Extraction from Online News Us-ing Binary Integer Programming. In *Proceedings of the Fifth International Joint Conference on Natural Language Processing*, pp. 165–173.

R. Hussey, S. Williams, R. Mitchell, I. Field, 2012. A Comparison of Automated Keyphrase Extraction Techniques and of Automatic Evaluation vs. Human Evaluation. *International Journal on Advances in Life Sciences*, vol 4 no 3 & 4, 136–153.

X. Li, B. Liu, and P. Yu, 2008. Time sensitive ranking with application to publication search, In *Proceedings of the 2008 IEEE International Conference on Data Mining*, pp. 893–898.

Q. Mei, X. Shen, C. Zhai, 2007. Automatic labeling of multinomial topic model, In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 490–499.

J. Mori, M. Ishizuka, and Y. Matsuo, 2007. Extracting Keyphrases to Represent Rela-tions in Social Networks from Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2820–2827.

H. Nakagawa, T. Mori, 2002. A simple but powerful automatic term extraction method, In *Proceedings of second international workshop on computational terminology*, pp. 1-7.

M. Paukkeri, I. Nieminen, M. Pöllä, and T. Honkela, 2008. A Language-independent Approach to Keyphrase Extraction and Evaluation. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 237–252.

X-H. Phan, L-M. Nguyen, and S. Horiguchi, 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, pp. 91-100.

S. Song, Y. Meng, and J. Sun, 2014. Detecting Keyphrases in Microblogging with Graph

Modeling of Information Diffusion. In *Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence*, pp. 26-38.

S. Song, Y. Meng, 2015a. Detecting representative tweets of micro-blogging users, In *Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering*, pp. 110-112.

S. Song, Y. Meng, 2015b. Classifying and ranking microblogging hashtags with news categories, In *Proceedings of the 9th IEEE International Conference on Research Challenges in Information Science*, pp. 540–541.

W. Steward, 1994. Introduction to the numerical solution of Markov chains, published by *Princeton University Press*.

X. Wan, and J. Xiao, 2008. Single Document Keyphrase Extraction Using Neighbor-hood Knowledge. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pp. 855-860.

W. Wu, B. Zhang, M. Ostendorf, 2010. Automatic generation of personalized annotation tags for twitter users, In *Proceedings of the 2010 Conference of the North American Chapter of the Association of Computational Linguistics*, pp. 689–692.

S. Wasserman, K. Faust, 1994. Social Network Analysis, published by *Cambridge University Press.*

M. Yang, J. Lee, S. Lee, and H. Rim, 2012. Finding interesting posts in Twitter based on retweet graph analysis. In *Proceedings of the 35th International ACM SIGIR conference on research and development in Information Retrieval*, pp. 1073-1074.

W. Zhao, J. Jiang, J. He, Y. Song, P. Achananuparp, E.P. Lim, and X. Li, 2011. Topical Keyphrase Extraction from Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 379–388.