Microblog Keyphrase Extraction Based on Similarity Features

Lizi Liao[1,2], Heyan Huang[1,2]

[1]Beijing Engineering Applications Research Center of High Volume Language Information Processing and Cloud Computing (BIT), Beijing, China

[2]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

{liaolizi, hhy63}@bit.edu.cn

Abstract - This paper proposes to extract keyphrases from microblog based on similarity features. We analyze a large number of microblogs and find an interesting phenomenon that people use various nugget phrases to express the same factoid while many of these nugget phrases show similarity relationships. We propose a similarity features based context-sensitive topical PageRank method for keyphrase ranking. We evaluate our proposed methods on a large microblog dataset. Experiments show that our system is very effective for keyphrase extraction**.**

Index Terms – Keyphrase extraction, Similarity features, Graph-based model.

I . Introduction

As a broadcast medium for broadcasting short informal messages, microblog has rapidly become popular. Though microblogs are quite noisy and informal, they provide a unique compilation of first-hand information about people's opinions and feelings. Analyzing such up-to-date and tremendous amount of information can be really helpful in domain like monitoring public opinion, crisis public relations. While keyphrases are very efficient in summarizing microblog content, extracting keyphrases from microblog is starting to receive more attention.

To extract a very small amount of representative keyphrases from a large microblog set is quite challenging. Due to the numerous, changeable and noisy nature of microblogs, unsupervised approach seems to be more appropriate to analyze it. Supervised methods always need a microblogs set with human-assigned keyphrases as training set. As mentioned above, microblogs increase exponentially and change rapidly. It is unpractical to label training dataset by human time to time to meet such need. Thus, we propose an unsupervised approach in this study.

Most existing keyphrase extraction algorithms focused on popular formal domains, such as papers or web pages. When applied to microblog, an extremely informal domain, their performance drops sharply. Compared with traditional text collections, keyphrase extraction from microblog is more challenging in several aspects. At first, microblogs are much shorter than traditional texts and not all microblogs contain useful information. Secondly, microblogs are written by a wide variety of users. Thus, microblogs about the same event or even microblogs containing the same meaning may have total different form of expression.

In this paper, we analyze a large number of microblogs and find an interesting phenomenon. When an event or a topic occurs, people tend to use various nugget phrases to refer to it. Therefore, widely used features like position, Term Frequency, TFIDF could not be very efficient to extract keyphrase. At the same time, we also find that though forms of phrases are different, their contexts or head nouns are somewhat similar. In our work, we propose two kinds of features to capture this phenomenon. One is context similarity of candidate phrases. The other is inner similarity, which calculates the similarity of head nouns.

For keyphrase extraction, there are standard three steps, namely, keyword ranking, candidate keyphrase generation and keyphrase ranking. When it applied to noisy microblog, the performance is affected. We propose to directly rank candidate phrases after a preprocessing procedure. We modify the context-sensitive topical PageRank method by introducing similarity features [1]. We find that preprocessing keyphrases using similarity features before ranking can largely help boost the performance.

Section 2 surveys related literatures on keyphrase extraction. Section 3 describes the interesting phenomenon we find and outlines our proposed extraction system. Section 4 presents experiments on the effectiveness of our system compared to baselines.

II . Related Work

In keyphrase extraction, there are mainly two methods. One is that it can be viewed as a two-category classification task. The other is that it can be treated as a ranking task.

In classification task, each term in the document is tagged as a keyphrase or not. Turney carries out a pioneering achievement in keyphrase extraction from documents, GenEx [2]. It uses frequency based and part-of-speech information as features and a genetic algorithm to tune a set of parameterized heuristic rules. Frank et al. propose KEA which uses TFIDF and first occurrence as features and Naïve Bayes as the classifier [3]. Hulth explores more linguistic knowledge [4]. Medelyan and Witten use semantic information on terms and phrases gleaned from a domain-specific thesaurus to boost the performance of automatic keyphrase extraction [5].

In unsupervised learning, keyphrase extraction can be treated as a ranking task. Candidate phrases are ranked by a saliency score using various features. A small set of top ranked

phrases are selected as keyphrases. Graph-based ranking methods are the state-of-the-art [6]. At first, these methods use some kind of relationship within documents, like word co-occurrence, to build up a graph. Then, random walk techniques are applied to measure word importance. At last, top ranked words are used in selecting keyphrases. Liu et al. propose to decompose the traditional random walk into topical PageRank considering topic information[7]. Zhao et al. extent the topical PageRank into context-sensitive topical PageRank. They apply it to extract topical keyphrases from Twitter and consider relevance and interestingness while ranking. Our model is mainly based on [1]. The difference is that we directly rank candidate keyphrases after a preprocessing procedure based on similarity features, which can easily dig out keyphrases which have various forms, and lower the effect of noise in microblog.

Our work also gets inspiration from the findings of Vahed Qazvinian and Dragomir R. Radev [8]. They design summary generation systems which use distributional similarity to build a network of words and capture the diversity caused by lexical choice by detecting communities in the network. Another inspiration comes from the study of Barker and Cornacchia [9]. They propose a simple method that whether a noun phrase can be a keyphrase or not is chosen by its length, frequency and the frequency of its head noun. In our work, we use head noun to measure the inner similarity between different phrases.

III . Keyphrase extraction

A. Nugget phrases and factoid phrases

A nugget phrase is defined to be a phrasal information unit. Different nugget phrases may represent the same atomic information community, which we call as a factoid phrase. In this paper, we analyze 15 sets of microblogs. Each set consists of a number of unique microblogs about the same event written by different people. The datasets used in our experiments are collected from weibo.com. Table I lists some of the sets and the number of microblogs in them.

TABLE I Microblog sets and the number of microblogs

ID	Keyphrase	Number
1	Red Cross Society	112
2	Shenzhou IX	86
…	…	…
15	Jobs's death	103

After analyzing these sets, we find that when more microblogs analyzed, the number of nugget phrases increases rapidly while the number of factoid phrases increases much slower. The results are showed in Fig.1. This can be observed when some other event or topic occurs (a movie is released, a social problem is uncovered, a scientific finding is published, etc.). People tend to refer to the same thing or write about it from various aspects using different phrases. Though forms of these phrases are different, context sets of them always show similarity relationship to some extent.

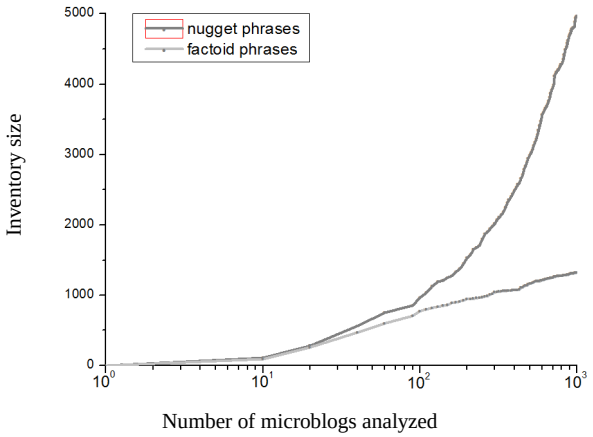

Fig. 1 The number of unique factoids and nuggets observed by analyzing n microblogs.

B. Preprocessing based on Similarity Features

For each topic, we use similarity features to find those candidate phrases which talk about the same factoid but appear as various nuggets. The features used are context similarity which tries to capture helpful information concealed in neighbor phrases, and inner similarity which tries to find useful information within these phrases themselves. For example, there are two candidate phrases k_i and k_j. If the similarity value of them caculated from the two similarity features is larger than the threshold θ, phrase k_i and phrase k_j will be merged into aphrase set S. If there are more other phrases similar to k_i and k_j, they all are added to the set S. In the keyphrase ranking part, each set is treated as a single node in the constructed graph. The co-occurrence relationships of those phrases in the set is preserved except those co-occurrence relationships within the same set.

Considering the diversity in phrase forms, we use context similarity of phrases to capture the nuggets of equivalent information units. We represent each phrase by its context set and calculate the similarity of such context sets. By this method, we indirectly introduce syntactic relation between phrases. Syntactic relation assumes that terms at a certain distance have syntactic or semantic relationship. Lyon et al. find that 70% of syntactic dependencies are between neighboring terms, and 17% at a distance of 2 [10]. In our method, each phrase k_i is represented by a bag of words or phrases l_i, which are those 6 (or less) closest words or phrases to k_i. This bag of words or phrases representation of phrases enables us to find the phrase-pair similarities. The similarity between phrases k_i and k_j is calculated by the similarities of the corresponding two bag of words or phrases which are decided by their conceptual approximation using HowNet[11], which is an on-line common-sense knowledge base unveiling inter-conceptual relations and inter-attribute relations of concepts as connoting in lexicons of the Chinese and their English equivalents.

We consider the similarity relationship between head nouns of phrases as inner similarity. Our decision is based on two observations. One is that hrases with more pre-modifiers are always more specific and tend to be more relevant to a particular part of an event or topic. The other is that when

writting about a given topic or event, people may use various pre-modifiers due to their different feelings or stands toward it. Since Chinese phrases' head nouns always are the nouns located in the last, we simply identify the last noun as head noun. Then, we calculate the inner similarity value of two phrases by the similarity of their head noun using HowNet.

C. *Topical PageRank for Keyphrase Ranking*

Zhao et al. introduce context-sensitive topical PageRank to identify keywords for keyphrase extraction. Topic-biased PageRank runs separately for each topic. We denote this context-sensitive topical PageRank as cTPR. The cTPR can avoid context-free propagation which may cause the rank to be off-topic.

$$R_t(w_i)=\lambda \sum_{j:w_j \to w_i} \frac{e_t(w_j,w_i)}{O_t(w_j)} R_t(w_j)+(1-\lambda)P_t(w_i) \qquad (1)$$

where the edge weight from w_j to w_i is parameterized by t. They compute edge weight $e_t(w_j,w_i)$ by counting the number of co-occurrences of these two words in tweets assigned to topic t. After keyword ranking using cTPR, they select the top S keywords for each topic, and then look for combinations of these keywords that occur as frequent phrases.

However, cTPR ignores the injection of noise in dividing phrases into separate words when selecting some top keywords for each topic. Taking the topic of "economy" as an example, the word "bank" always tends to rank high in TPR and cTPR. There might be many phrases containing the word "bank" but none of them is a proper keyphrase, since there are so many bank names and specialized terms containing "bank". We denote those words like "bank" as pseudo-keywords. It can be quite possible that there are too many pseudo-keywords that real keywords are expelled to rank relatively low. Therefore, the combinations of those high ranked keywords might not be the keyphrases we want.

So in this paper, we propose to use a similarity features based context-sensitive topical PageRank method. Formally, we have

$$R_t(S_i)=\lambda \sum_{\substack{k_s \in S_j \\ k_e \in S_i}}^{k_s \to k_e} \frac{e_t(k_s,k_e)}{\sum_{k \in S_j} O_t(k)} R_t(S_j)+(1-\lambda)P_t(S_i) \qquad (2)$$

where Rt(S) is the similarity features based topic-specific PageRank score of node S in topic t. S contains similar candidate phrases k given by similarity features. For each set, we choose the most frequent candidate phrase in it to represent the whole set after ranking. We denote this similarity features based context-sensitive topical PageRank as scTPR.

IV . Experiments

A. *Dataset*

Since there is no existing standard dataset for keyphrase extraction from Microblog, we constructed a dataset. The majority of the microblogs collected were published in a year period from June 20, 2011 to June 20, 2012. These microblogs were randomly collected from weibo.com. We removed common stopwords like particle and words appeared in fewer than 10 microblogs. We also removed all microblogs which were discussed by less than 10 users or written by users who had fewer than 10 microblogs. To capture the reposting behavior in Microblog, we simply duplicate the microblog when it is reposted. Some statistics of this dataset after cleaning are shown in Table II.

TABLE II Some statistics of the dataset

user	microblog	candidate phrase	token
1,280	25,660	69,520	184,421

B. *Experiment Results*

We compare our method with the context-sensitive topical PageRank method (cTPR), which represents the state-of-the-art.

We run those methods separately. Their results are filtered. We eliminate phrases with less meaning and phrases which are quite similar to those appeared before. We compute the nDCG score for those methods. Results are listed on Table III.

TABLE III Comparisons of scTPR and baseline

	cTPR	scTPR
nDCG@5	0.5264	0.5279
nDCG@10	0.5211	0.5233
nDCG@25	0.4974	0.5110
nDCG@50	0.4836	0.4703

As shown on Table III, we can observe that our approach scTPR performs better than baseline. To investigate the reason behind, we conducted a futher experiment on the 15 microblog sets. The 15 datasets are manually selected, thus little noisy information is included. The results are listed in table IV.

TABLE IV Comparisons of scTPR and cTPR on 15 datasets

		@5	@10	@25
Shenzhou IX	cTPR	0.766	0.753	0.749
	scTPR	0.821	0.813	0.799
Red Cross	cTPR	0.793	0.790	0.785
	scTPR	0.811	0.800	0.784
…	…	…	…	…
Jobs's death	cTPR	0.823	0.814	0.791
	scTPR	0.816	0.807	0.795

When we compare the results on those selected datasets separately, we can see that the performance of scTPR on Shenzhou IX set is especially better than cTPR. In Chinese, (Shenzhou IX) has a popular alternative form (S IX). Their context sets are quite similar which are always about Temple I, shake hands or congratulations. The context similarity feature might have been of great use in this part.

The performance of scTPR on Red Cross Society set is slightly better than cTPR. The reason for this better performance might be that Red Cross Society is the head noun of many forms like (Red Cross Society of China), (Red Cross Society of China Beijing Branch).

V . Conclusion

In this paper, we propose two context features for Microblog topical keyphrase extraction. They capture the phenomenon that people always express things in various forms while those forms have similarity relationships. In our experiments, those two combined features are shown to be effective, especially in digging out keyphrases which submerged by various forms.

In the future, we will further exploit the representation of the two proposed features. These features are set to be discrete and combined linearly. There should be other potential ways to simulate the real relationship between them.

VI . Acknowledgment

This work was supported by the National Basic Research Program of China (No. 2013CB329300).

References

[1] Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-peng Lim and Xiaoming Li. Topical keyphrase extraction from Twitter. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 379-388, Portland, Oregon, June 2011. Association for Computational Linguistics.

[2] Peter D. Turney. Learning to extract keyphrases from text. Technical Report, National Research Council Canada, Institute for Information Technology , 1999.

[3] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In Proceedings of the International Joint Conference on Artificial Intelligence, pages 668–673, Stockholm, Sweden ,1999.

[4] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of EMNLP, pages 216-223, 2003.

[5] Olena Medelyan, Ian H. Witten. Thesaurus based automatic keyphrase indexing. In: Proceddings of JCADL, 2006.

[6] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. In Proceedings of EMNLP, pages 404-411, 2004.

[7] Zhiyuan Liu, Wenyi Huang, Yabin Zheng and Maosong Sun. Automatic keyphrase ex-traction via topic decomposition. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 366-376, Massachusetts, USA, Octorber 2010. Association for Computational Linguistics.

[8] Vahed Qazvinian and Dragomir R. Radev. Learning from collective human behavior to introduce diversity in lexical choice. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Pages 1098-1108, Portland, Oregon, June 2011. Association for Computational Linguistics.

[9] Ken Barker and Nadia Cornacchia.: Using noun phrase heads to extract document keyp-hrases. In Canadian Conference on AI , 2000.

[10] Lyon, C., Nehaniv, C., and Dickerson, B. Entropy indicators for Investigating early language process. In AISB'05: Proc. of EELC'05, Pages 143-150, 2005.

[11] HowNet, http://www.keenage.com/