Enhance Social Context Understanding with Semantic Chunks

Siqiang Wen, Zhixing Li, and Juanzi Li

Dept. of Computer Sci. and Tech., Tsinghua University, China
{wensq2329,adam0730,lijuanzi2008}@gmail.com

Abstract. Social context understanding is a fundamental problem on social analysis. Social contexts are usually short, informal and incomplete and these characteristics make methods for formal texts give poor performance on social contexts. However, we discover part of relations between importance words in formal texts are helpful to understand social contexts. We propose a method that extracts semantic chunks using these relations to express social contexts. A semantic chunk is a phrase which is meaningful and significant expression describing the fist of given texts. We exploit semantic chunks by utilizing knowledge learned from semantically parsed corpora and knowledge base. Experimental results on Chinese and English data sets demonstrate that our approach improves the performance significantly.

1 Introduction

Social context understanding, whose aim is to get the main idea of the given social contents, has become more and more important and fundamental problem with the explosive growth of user generated content. However, social contexts are usually short, informal and incomplete, these characteristics make us difficult to extract phrases to express the given social contents. Thus, it is a challenging problem in social media processing.

Keyphrase extraction can be regarded as an aspect of text understanding. For formal texts, amount of effective approaches have been proposed and nice results have been achieved. Keyphrase extraction approaches can be roughly categorized into supervised [3,17] and unsupervised [15,11,7]. Supervised methods use various kinds of features to build a classifier. There are litter works using supervised methods for keyphrase extraction for social contexts, as lack of manually annotated training data. Unsupervised algorithms, regarding keyphrase extraction as a ranking task, utilize TF-IDF [15], co-occurrence[11], topic[7] and so on to rank candidates. Based on unsupervised approaches, a number of researchers have used them on twitter's texts. [18,19,1] have used and expanded TF-IDF and TextRank for keyphrase extraction. However, performance of these methods on social contexts is not as good as that on formal texts. Social contexts are usually short to record some trivia and their structure and grammar are usually incomplete. In addition, social contexts contain lots of impurities

C. Zong et al. (Eds.): NLPCC 2014, CCIS 496, pp. 251–262, 2014.

like abbreviations, misspelled words, slang words, and emoticons. These characteristics of social contexts make keyphrase extraction in difficulty to deal with social contexts. These methods can only extract words and simple phrases and the precision is very low. For example, the average number of Chinese character of Sina keywords [6,16] is about 2.08. Many meaningful and semantic phrases can't be extracted with these methods.

Social messages are casual logs of users' everyday life, which often lack of structure compared to formal text such as book and news reports; nevertheless there are some complete words and phrases in social contexts and most of these words and phrases are related to main idea of social contexts. Machine can't understand such texts, but human can capture the gist of social contexts by using part of complete words and phrases. Our method is just based on the idea that we usually do not need all relations between words like parser and only part of relations between importance words are enough to capture the main idea of the given sentence. Therefore, we can use such relations to extract semantic chunks to help to understand social contexts.

In this paper, we denote a semantic chunk as a phrase which is meaningful and significant expression describing main idea of given texts. Semantic chunk consists of semantic dependency words, which may be not consecutive words. For example, given texts *most of the passengers on board survived*, we will get semantic chunk such as *"passengers survived"*, in which *"passengers"* and *"survived"* are not consecutive words, in other word there are several words between *"passengers"* and *"survived"* .

To acquire semantic chunks, there are several problems to solve, such as how to find important relations, how to use these relations to form readable sematic dependency phrase and how to solve the limit of labeled corpora. To solve these problems, we use an corpus with annotated semantic dependency relationships for formal language to obtain dependent knowledge between words of nouns, verbs and adjectives. Then, we extract semantic chunks from social content by incorporating these dependency relationships with a knowledge base, WordNet for English and Tongyici Cilin for Chinese. Specifically, we propose a model which identifies the important words that evoke a semantic phrase in a given sentence and the dependent words which are dependent on the target words. The model is trained on semantic dependency corpora and extended by a external knowledge base since semantic dependency corpora is limited for social content. We acquire some phrases as candidates at first and we use chunk knowledge learnt by section 4.1 and some rules to expand candidates to form semantic chunks.

The main contributions of our paper are summarized as follows.

– We propose the method to use semantic chunks to solve the problem of social content understanding. We utilize part of semantic dependency relations between importance words learned from semantic dependency corpora and knowledge base to extract semantic chunks to capture the gist of the given document. The method does not need all relations between words like parser.

- We learn word knowledge, relation knowledge and distance knowledge from semantic dependency corpora. With these knowledge, we can extract long distance dependency phrases to form semantic chunks.
- To verify the effectiveness and efficiency of our method, we evaluated our method over Chinese and English social contexts, and the experimental results show that our method significantly outperforms the baseline methods.

2 Related Work

As this paper mainly studies social context understanding by extraction semantic chunks from them, we focus our literature review for approaches about keyphrase extraction and dependency parsing.

Keyphrase extraction is the process that identify a few meaningful and significant terms that can best express contexts. Generally speaking, keyphrase extraction approaches can be roughly categorized into two principled approaches: supervised and unsupervised. Supervised algorithms consider keyphrase extraction as a classification problem to classify a candidate phrase into either keyphrase of not. [3,17] have used some features, such as frequency, location, statistical association and other linguistic knowledge, to classify a candidate phrase. Due to lack of manually annotated training data, researchers hardly use supervised methods for keyphrase extraction for social contexts.

Unsupervised algorithms usually regard keyphrase extraction as a ranking task, which assigns a score to each candidate phrases by various methods then picks out the top-ranked terms as keyphrases. [15] has proposed a simple unsupervised algorithms using TF-IDF to extract keypharse. Graph-based ranking methods become popular after TextRank model [11] proposed by Mihalcea and Tarau. Some approaches have been proposed to improve TextRank. Liu proposed other unsupervised algorithms, including clustering-based [8] and topic-based [7] methods.

While existing research mainly focuses on formal articles, the rapid growth of social network raises the needs of research on informal language. Informal language contexts are much shorter than formal articles. It is more difficult to extract keyphrase from informal contexts than from traditional articles. [18] uses TFIDF and TextRank, two standard keyword ranking techniques, to extract keyword. NE-Rank [1] proposes an enhanced PageRank to extract keyphrase for Twitter. [19] modifies Topical PageRank [7] to find topic keyphrase. PolyU [14] extracts core words and expands the identified core words to the target keyphrases by a word expansion approach. These unsupervised methods can extract words and simple phrases, but can not identify more meaningful and semantic phrases. Words and simple phrases can't accurately cover the mean of texts.

Dependency parsing can provide a representation of lexical or semantic relations between words in a sentence and have been designed to be easily extract textual relations. Stanford Dependencies [9] provides English and Chinese dependent relations between words. But these parser can't perform nice on social

contexts. Factually, we usually do not need all relations between words in a given sentence, and we only want the relations between importance words. In this paper, we use dependent knowledge to get the relations between importance words, such nouns, verbs and adjectives. Parser need full structure information of sentence, but we need part of structure information.

3 Framework

The method in this paper is mainly inspired by the idea that semantic chunks in social contexts are helpful to understand social texts. We find candidate phrases and then rank them to select semantic chunks. Not all words in a document are fit to be selected as candidate phrases. In [17], candidate phrases were found using n-gram. Liu [8] used exemplars to extract multi-word candidate phrases. Mihalcea and Tarau in [11] used n-grams as a post-processing step to form phrases. However, in this paper, we use chunk knowledge to extract candidate phrases rather than words and then select semantic chunks. Our framework consists of two processes, including

- **Dependency Knowledge Learning** We learn semantic dependency knowledge from annotated semantic dependency corpus for formal texts. The knowledge contains word dependency knowledge, relation dependency knowledge and distance knowledge.
- **Semantic Chunking** Given a sentence, a set of semantic chunk is generated using learned knowledge and external knowledge bases. The step contains target words identification, semantic pair discovery and semantic chunk generation.

We now introduce semantic dependency corpora which are annotated with dependency grammar and some related denotations for they frequently are used below. We use Chinese and English semantic dependency corpora(SND [13] and TreeBank) to learn chunk knowledge. SDN is Chinese semantic dependency corpus built by Tsinghua University. We gain English semantic dependency corpus by Penn TreeBank with Stanford Dependencies. Table 1 shows some import figures about two semantic dependency sets we use in this paper. Pair knowledge is the number of word pairs whose two word have semantic dependency relation each other. Relation knowledge is the number of pairs (r_a and r_b).

Table 1. Statistics on Semantic Dependency corpora

Data Set	*Sentences*	*Words*	*Vocabulary*	*Pair knowledge*	*Relation knowledge*
SDN	132396	908048	34539	550536	619084
TreeBank	240873	2514549	62632	1316311	1846158

In semantic dependency datasets, each sentence is represented by a dependency tree. For example, Figure 1 shows the dependency tree of the sentence

The sales of newly launched IPhone_5s disappoint investor. The root is "disappoint". Two words have semantic dependency relation if there is a directed edge between the two words. For example, *"investor"* is dependent on *"disappoint"* and the two words have relation of *"dobj"* for an edge links the two words.

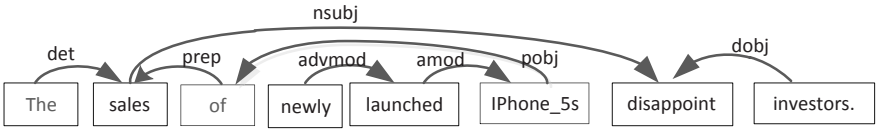

Fig. 1. A semantic dependency tree

We now give some related denotations used in this paper. Table 2 defines some variables used in our approach.

Table 2. Notation of some frequently occurring variables

Symbol	Description
d	a document of social contexts, a sentence or several sentences, which can be represented as a sequence of words $(w_1, w_2, \cdots, w_{n_d})$, where n_d means the number of words in the document d
W_K	word vocabulary of semantic dependency corpora
Kno_w	Word knowledge of semantic dependency corpora, which is represented as the set of $\{(w_i, w_j, r_a)\}$, where w_i and w_j has relationship of r_a
Kno_r	Relation knowledge of semantic dependency corpora, which is represented as the set of relation sequential knowledge($\{(r_a, r_b)\}$), which means r_a and r_b are two sequential co-occurring relationships.

4 Method

4.1 Dependency Knowledge Learning

In this part, we learn semantic dependency knowledge from semantic dependency datasets. The knowledge contains lexical collocations, and distance value between two dependent words and three words's semantic relations.

For each sentence, if two words(w_i, w_j) have dependent relation(r_a), we put (w_i, w_j, r_a) into Kno_w. There is a special case, for example, in Figure 1, "sales" and "$IPhone\_5s$" have semantic dependency relation. "of" is in the dependent path("$IPhone\_5s$" $\rightarrow$ "of" $\rightarrow$ "sales"), but it is a red word. The red words are functional words and we remove the pair relations about functional words, because they are useless to capture meaning of content. So "sales" and "$IPhone\_5s$" are dependent words due to transitivity. We use transitivity to remove form words and get expanded dependent words.

We extract words dependency knowledge($C_w(w_i, w_j)$), and distance knowledge($C_d(w_i, w_j)$). $C_w(w_i, w_j)$ is defined as follows:

$$\begin{aligned} C_w(w_i, w_j) &= p(w_i w_j) \lg \frac{p(w_i w_j)}{p(w_i)p(w_j)} \\ &= \frac{\#N(w_i, w_j)}{\#Tp} \lg \frac{\#N(w_i, w_j)(\#Tw)^2}{\#N(w_i)\#N(w_j)\#Tp} \end{aligned} \tag{1}$$

where $\#N(w_i, w_j)$ is the number of pairs(w_i, w_j)) occurred in the corpus, $\#N(w_i)$ is the number of w_i, $\#Tw$ is the number of words and $\#Tp$ is the number of pairs in the annotated corpus. w_i is word's prototype and part-of-speech. $C_d(w_i, w_j)$ is the average distance between two dependent words in the corpus.

We expect that semantic chunk contains not only two words but also three or more words. If r_a and r_b have the same word in the dependency tree, we put (r_a, r_b) into Kno_r. For example, in Figure 1, nsubj and dobj both have word "disappoint", then we put $(nsubj, dobj)$ into Kno_r. In addition, we learn $(h(r_a, r_b))$ and other probability $(p(r|(w_i, w_j)))$ for forming semantic chunk whose number of word is more than two. $p(r|(w_i, w_j))$ is the probability of relation r when given words w_i and w_j. $h(r_a, r_b)$ is mutual information co-occurrence of two relations (r_a, r_b). In section 4.2, we use this knowledge and rule to get semantic chunks which are more than two words.

4.2 Semantic Chunking

We denote S_d as semantic chunks, which can be denoted as $(s_{d_1}, s_{d_2}, \cdots, s_{d_{n_s}})$, and denote s_{d_i} as a word or phrase, which can be represented as a sequence of words $(w_{d_{i_1}}, w_{d_{i_2}}, \cdots, w_{d_{i_n}} | d_{i_1} < d_{i_2} < \cdots < d_{i_n})$ where n is among 1 and 3. A semantic chunk consists of target words and their relate words. Target word which can evoke a semantic phrase in a given sentence represents the dominant concepts in the social content. We denote T_d as target word set of d. s_{d_i} can be represented as follow.

$$s_{d_i} = \begin{cases} (w_i), & w_i \in T_d \\ (w_i, w_j), \exists r_a(w_i, w_j, r_a) \in Kno_w \\ (w_i, w_j, w_k), \exists r_a(w_i, w_j, r_a) \in Kno_w, \\ \qquad \exists r_b(w_j, w_k, r_b) \in Kno_w, \\ \qquad (r_a, r_b) \in Kno_r \end{cases} \tag{2}$$

Words(w_i, w_j, w_k) can be expended by knowledge base, then determine whether (w_i, w_j, r_a) and (w_j, w_k, r_b) are in Kno_w. For example, w_i and w_j are expanded to w_{ik} and w_{jl}(4.2), where w_{ik} and w_{jl} are in W_K, if there is r_a and (w_{ik}, w_{jl}, r_a) is in Kno_w, then (w_i, w_j, r_a) is in Kno_w.

The task of semantic chunk extraction is to find a set of semantic chunks S_d when given a item of social contexts d. In other word, we find semantic dependency phrases from lexical collocations of all words in d. We denote the

number of lexical collocations is C_{all}. Semantic chunk candidates can be obtained with three steps, including target words identification semantic chunk discovery and semantic chunk generation.

Target Words Identification. The size of C_{all} is 2^n, where n is the length(the number of words) of social text d. The size of C_{all} is too large. C_{all} contains all combinatorial words. We use target word to reduce candidates without affecting the result. What's more, we think important target words are usually in semantic chunks. Generally speaking, verbs, nouns, adjectives, and even prepositions can evoke phrases under certain conditions. In [2], target words are identified by rules followed Johansson and Nugues [4], they select verbs, nouns, adjectives, and even prepositions as target words. Given a text, verbs, adjectives, adverbs and prepositions usually depend on nouns or be depended by noun. In other word, we only select noun and extract their related dependent relations, and thus we can get relevant verbs, adjectives and so on. So we pick out nouns as potential target words. Especially, entity is more important to dominant concept than other words. Therefore, we prefer proper nouns as goal words and give three bonus to proper nouns. We also consider frequency and position as features. We denote $score(w_i)$ as the score of target value. If w_i is target word, $score(w_i)$ is calculated by above strategy. If not, $score(w_i)$ equals one.

Semantic Pair Discovery. Given a sentence, in order to get semantic phrases, we first get target words in 4.2, then use these words, knowledge base and chunk knowledge to get related words set of target words. The chunk knowledge learnt from dependency corpus is limited. If we only use some part-of-speech patterns to select relate words of word that is not in W_K, most of the results are not readable and meaningless. Therefore, we utilize knowledge base, lexical collocations, part-of-speech knowledge and distance knowledge to get semantic chunk. We use **Tongyici Cilin** (A Dictionary of Syn-onyms) as knowledge base for Chinese and **WordNet** [12] for English.

Formally, we define wd as a window's size whose furthest word is distance target words as wd, where wd can be set the length of sentence. We find accompaniment words of t_i in a certain range of window(wd). We denote $Expand(w_i) = \{w_{i1}, w_{i2}, \cdots, w_{ie}\}$ as a set expanded w_i by knowledge base with semantic category. Any word in $Expand(w_i)$ is in W_K. Let Sim_{ij} as similarity of w_i and w_{ij}. Sim_{ij} is calculated through knowledge base.

We define $R_w(w_i, w_j)$ as the word value and $R_d(w_i, w_j)$ as the distance value of w_i and w_j. $R_w(w_i, w_j)$ will be calculated as follow:

$$R_w(w_i, w_j) = \begin{cases} C_w(w_i, w_j), & w_i, w_j \in W_K \\ \sum_{k=1}^{e} \sum_{l=1}^{e} p_{kl} S(w_{ik}, w_{jl}), & else \end{cases} \tag{3}$$

where w_i is a target word. If w_i and w_j both are not in W_K, we will expand w_i and w_j by knowledge base. e is amount of expanded words, and $S(w_{ik}, w_{jl})$ is,

$$S(w_{ik}, w_{jl}) = C(w_{ik}, w_{jk}) \times Sim_{ik} \times Sim_{jl} \tag{4}$$

p_{kl} is,

$$p_{kl} = \frac{\#N(w_{ik}, w_{jk})}{\sum_{k=1}^{e}\sum_{l=1}^{e} \#N(w_{ik}, w_{jk})} \tag{5}$$

Suppose the distance between w_i and w_j is d_{ij}. $R_d(w_i, w_j)$ is the average distance in the corpus. If w_i and w_j both are not in W_K, we will expand w_i and w_j by knowledge base and use method like $R_w(w_i, w_j)$ to calculate $R_d(w_i, w_j)$. In order to determine wether w_i and w_j can form semantic chunk candidate, we then model the probability of candidate d as a function incorporating words knowledge and distance knowledge.

$$R(w_i, w_j) = \alpha R_w(w_i, w_j) + (1-\alpha)(|R_d(w_i, w_j) - d_{ij}|)^{-1} \tag{6}$$

Semantic Chunk Generation. For each target words, we select top-m related words to form candidate set(EW), which is a sub set of C_{all}. We remove many unimportant candidates form C_{all} through above two steps. We can get semantic chunk candidates whose size is two. In this part, we use chunk knowledge learnt by section 4.1 and some rules to expand candidates.

Given three words(w_i, w_j, w_k), w_i and w_j have relation r_m, w_j and w_k have relation r_n. Then

$$C(w_i, w_j, w_k) = p(r_m|(w_i, w_j))p(r_n|(w_i, w_j))h(r_m, r_n) \tag{7}$$

We select top-v of all $C(w_i, w_j, w_k)$. Then selected phase (w_i, w_j, w_k) as new candidates replaces (w_i, w_j) and (w_j, w_k). $R(w_i, w_j, w_k)$ is gotten by

$$R(w_i, w_j, w_k) = (R(w_i, w_j) + R(w_j, w_k))/2 \tag{8}$$

Then we select top n_s as semantic chunks from all candidates by standard logistic regression model. We use two features. One feature is score of phrases($R(w_i, w_j, w_k)$ or $R(w_i, w_j)$). Another is target value of phrases(section 4.2).

Through above steps, We get some two and triple phrases. Then we add some expanded words by heuristic rules to form phrases. Take *I have a beautify house in the woods* for example, we can extract *"house woods"*, but it's not nice phrase. We need to make some change. We add preposition(*in*) to phrase *"house woods"*. *"house in woods"* is better than *"house woods"*.

5 Experiments

5.1 Datasets

As far as we know there is no existing benchmark dataset and no gold standard answers for semantic chunks on social contexts. To evaluate the performance

of our method, we carry out our experiments on two real world datasets, microblogs(Chinese) crawled from Sina Weibo(China) and news comments(English) from Yahoo!. The blogs cotain blog posts that cover a diverse range of subjects. The statistics of the datasets is shown in Table 3, where $\mid D \mid$, $\mid W \mid$, $\mid V \mid$, $\mid N_s \mid$, $\mid N_w \mid$ are the number of document, the number of words, the vocabulary of contexts, the average number of sentences in each document and the average number of words in each sentence, respectively.

Table 3. Statistics on DataSets

Dataset	$\mid D \mid$	$\mid W \mid$	$\mid V \mid$	$\mid N_s \mid$	$\mid N_w \mid$
Sina(cn)	1000	129304	15318	7.06	18.32
Yahoo!(en)	1000	97392	10821	5.96	16.34

We use some heuristic rules to filter out the noisy words in advance. Firstly we remove emoticons and URL. Secondly, English documents are tokenized and tagged, while Chinese documents are segmented into words and then tagged. Finally, for both datasets, we identify stop words. We do not remove stop words in the original texts for stop words may be used as expanded words.

5.2 Experiments Setup

Evaluation Methods. To guarantee the low noise of the manual annotated data, we totally employ 1000 blogs posts(Chinese) and 1000 news comments (English), and for each document we ask at least 3 different annotators to rate the corresponding semantic chunks and keyphrases. Finally, each document is rated by averaging ratings from annotators. For each document, annotators were asked to rate the labels based on the following ordinal scale [5]:

3: Very good phrase, completely capturing gist of the document
2: Reasonable and readable phrase, but not completely capturing gist
1: Phrase is related to the contexts, but not readable
0: Phrase is completely inappropriate

Baseline Methods. We use **SmanC** to denote semantic chunk with knowledge base and **SmanC-KB** to denote semantic chunk without knowledge base. We select two major types of baseline methods for comparison: unsupervised keyphrase extraction methods and parsing which is the process of analysing contexts.

Unsupervised Keyphrase Extraction Method: Unsupervised keyphrase extraction methods assigns a score to each candidate phrases by various methods then picks out the top-ranked terms as result. There are some unsupervised keyphrase extraction methods proposed for social contexts. In this paper, we use **TextRank** [11] as baselines. TextRank build a word graph based on the

co-occurrence between words, then execute PageRank on the graph to give score for each keyphrase candidate. [18,1] have used TextRank for extraction with social contexts. To build the graph for TextRank, we will select noun phrases, verb phrases and adjectives as candidates. According to our experiments, TextRank's best score is achieved when vertices' co-occur within a window of two words for microblog documents and three words for news comments. Damping factor of 0.85, uniform prior of 1.0 and 50 iterations are set.

Parsing: There are many parser proposed. In this paper, we use **Stanford Dependencies** [9] for English documents and **MSTParser**(Minimum-Spanning Tree Parser) [10] for Chinese documents as baselines. MSTParser is trained by SDN, a labeled semantic dependency corpora using in section 4.1. In our experiment, we will extract phrases whose one word is noun or verb and other words semantically depend or be depended by the word. We will extract noun phrases and verb phrases. Then final result is selected by frequency and position information, such as head or tail of sentence.

5.3 Results and Analysis

Table 4 gives the performance results on two datasets, and the best performances in the comparisons are highlighted in bold. m and v both are set to 2 according to experiment performance. As the length of each document is short, we select top-3 as final phrases for each methods. Our proposed method **SmanC** outperforms the baseline methods TextRank and Parsing on both datasets. Our method utilizes semantic dependency knowledge, and don't care the complete structure of sentence. We can see textRank's performance is not good , because document is very short and the co-occurrence relation can not reflect the meaning of whole document. In addition, textRank can only extract some simple words and phrases. These reasons make textRank worse than other methods.

Table 4. Overall results of various methods for social contexts

	SmanC	*SmanC-KB*	*parser*	*textRank*
Sina(cn)	**2.237**	2.156	1.918	1.424
Yahoo!(en)	**1.871**	1.859	1.548	1.213

Results of SmanC-KB is higher than that of parser for microblogs posts and news comments. The performance of SmanC is higher than SmanC-KB for two datasets. The reason is that SmanC utilizes knowledge base to expand candidate set and can extract more phrases. But we can find some expanded phrases not readable. SmanC can extract longer phrases than the other methods. The average length of phrases of SmanC is 4.46 for Chinese documents and 2.53 for English document. However, a small part of long phrases is not reasonable. The performance of Chinese documents is better than that of English documents, because we deal with words in English documents while through Chinese word

segmentation Chinese word are not only a word but a simple phrase. For example, the phrase "New York" has two words in English, but a word after segmentation in Chinese. Although we make use of simple rules to combine some nouns and verbs to form nice phrases in order to get better performance, the results of news comments are not as good as microblogs'.

Furthermore, we investigate the results and discover that our method failed to find and appropriate phrases when encountered wrong tag of words. For example, give a sentence *The/DT injured/JJ included/VBD two/CD FBI/NP agents/NNS* , and we only get *FBI agents*. In fact, the tag of word *injured* is NN. With the right tags, we will get better phrase *injured included FBI agents*. Sometimes, we will not get reasonable and readable phrase without proper tags.

6 Conclusion and Future Work

We extract semantic chunks from social contexts to solve the problem of social context understanding. There're many future directions of this work such as automatical labelling. We will explore our method on other languages and on other test data to investigate and validate the robustness of our approach.

Acknowledgment. The work is supported by 973 Program (No. 2014CB340504), NSFC (No. 61035004), NSFC-ANR (No. 61261130588), European Union 7th Framework Project FP7-288342 and THU-NUS NExT Co-Lab.

References

1. Bellaachia, A., Al-Dhelaan, M.: Ne-rank: A novel graph-based keyphrase extraction in twitter. In: Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, WI-IAT 2012, vol. 01, pp. 372–379. IEEE Computer Society, Washington, DC (2012)
2. Das, D., Chen, D., Martins, A.F., Schneider, N., Smith, N.A.: Frame-semantic parsing (2013)
3. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, pp. 216–223. Association for Computational Linguistics, Stroudsburg (2003)
4. Johansson, R., Nugues, P.: Lth: Semantic structure extraction using nonprojective dependency trees. In: Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval 2007, pp. 227–230. Association for Computational Linguistics, Stroudsburg (2007)
5. Lau, J.H., Grieser, K., Newman, D., Baldwin, T.: Automatic labelling of topic models. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT 2011, vol. 1, pp. 1536–1545. Association for Computational Linguistics, Stroudsburg (2011)
6. Liu, Z., Chen, X., Sun, M.: Mining the interests of chinese microbloggers via keyword extraction. Frontiers of Computer Science 6(1), 76–87 (2012)

7. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, pp. 366–376. Association for Computational Linguistics, Stroudsburg (2010)
8. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, vol. 1, pp. 257–266. Association for Computational Linguistics, Stroudsburg (2009)
9. de Marneffe, M.C., Manning, C.D.: The stanford typed dependencies representation. In: Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, CrossParser 2008, pp. 1–8. Association for Computational Linguistics, Stroudsburg (2008)
10. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 523–530. Association for Computational Linguistics, Stroudsburg (2005)
11. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: EMNLP 2004, pp. 404–411 (2004)
12. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM 38(11), 39–41 (1995)
13. Mingqin, L., Juanzi, L., Zhendong, D., Zuoying, W., Dajin, L.: Building a large chinese corpus annotated with semantic dependency. In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, SIGHAN 2003, vol. 17, pp. 84–91. Association for Computational Linguistics, Stroudsburg (2003)
14. Ouyang, Y., Li, W., Zhang, R.: 273. task 5. keyphrase extraction based on core word identification and word expansion. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval 2010, pp. 142–145. Association for Computational Linguistics, Stroudsburg (2010)
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage., 513–523 (1988)
16. Liu, Z., Tu, C., Sun, M.: Tag dispatch model with social network regularization for microblog user tag suggestion (2012)
17. Turney, P.D.: Learning algorithms for keyphrase extraction. Inf. Retr., 303–336 (2000)
18. Vu, T., Perez, V.: Interest mining from user tweets. In: Proceedings of the 22nd ACM International Conference on Conference on Information Knowledge Management, CIKM 2013, pp. 1869–1872. ACM, New York (2013)
19. Zhao, W.X., Jiang, J., He, J., Song, Y., Achananuparp, P., Lim, E.P., Li, X.: Topical keyphrase extraction from twitter. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT 2011, vol. 1, pp. 379–388. Association for Computational Linguistics, Stroudsburg (2011)