Improved Automatic Keyword Extraction Given More Semantic Knowledge

Kai Yang[1], Zhenhong Chen[1], Yi Cai[1(✉)], DongPing Huang[1], and Ho-fung Leung[2]

[1] School of Software Engineering, South China University of Technology, Guangzhou, China
ycai@scut.edu.cn

[2] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China

Abstract. Graph-based ranking algorithm such as TextRank shows a remarkable effect on keyword extraction. However, these algorithms build graphs only considering the lexical sequence of the documents. Hence, graphs generated by these algorithm can not reflect the semantic relationships between documents. In this paper, we demonstrate that there exists an information loss in the graph-building process from textual documents to graphs. These loss will lead to the misjudgment of the algorithm. In order to solve this problem, we propose a new approach called Topic-based TextRank. Different from the traditional algorithm, our approach takes the lexical meaning of the text unit (i.e. words and phrase) into account. The result of our experiments shows that our proposed algorithm can outperform the state-of-the-art algorithms.

Keywords: Keyword extraction · Topic model · Graph-based ranking algorithm · Semantic analysis

1 Introduction

Automatic Keyword Extraction is the technology that can generate keywords from documents. This technology is also widely used in text mining and information retrieval and it becomes a popular research field these years. Many graph-based ranking algorithm, such as TextRank [18], have been proposed to extract keywords from documents. These algorithms construct the graph just according to the sequence of words in the documents, which reflects the textual structure inside the documents. However, the sequence of words can not reveal the semantic information embedded in documents. Hence, graphs built by these ranking algorithms take no account of the semantic relationship between words. This motivates us to propose an algorithm taking the semantic information into account.

Several researches have been done to extend the graph-based ranking algorithms. In [11], the importance of words is considered and they add the weight of

H. Gao et al. (Eds.): DASFAA 2016 Workshops, LNCS 9645, pp. 112–125, 2016.
DOI: 10.1007/978-3-319-32055-7_10

vertexes into the graph according to term weighting schemes such as TF and RW [22]. The main point of this approach is adding the statistical information into traditional TextRank. However, the weighting schemes can not really reflect the real semantical relationship inside the document. Liu et al. proposed an extension of TextRank, called Topical PageRank [17], basing on Latent Dirichelt Allocation, which is a popular topic model. They assigned weight to vertexes according to the topic distribution. Latent topic generated by topic model is a set of words that have similarity in semantic meaning. In this paper, we propose a different way to combine topic model and TextRank. We will compare our approach with Topical PageRank in our experiment.

In this work, we attempt to propose a new way to construct graphs for graph-based ranking algorithm. Different from the state-of-the-art algorithm, we build graphs relying not only on the local context, but also the lexical meaning of a text unit(i.e. words or phrase). Our main idea is that the connections between the vertexes having semantical similarity need to be intensified. In order to extract the semantical relationship from textual documents according to the lexical meaning of words or phrase, topic model, e.g. LDA and its variants, is applied in our work. Besides, the approach we proposed is supervised, and we learning the topical knowledge from train corpora. And then these learned knowledge is applied when we build the graphs. In general, the main contribution of us is that we find out a new way to combine semantical information into traditional graph-base keyword extraction algorithm, and we prove that this way can improve the performance of the state-of-the-art approaches.

In this paper, we firstly show the insufficiency of the state-of-the-art graph-based algorithm taking the TextRank for example in Sect. 3. Secondly, we raise our proposed approach and make its mechanism clear in Sect. 4. Finally, we conduct several experiments to verify the effectiveness of our approach and discuss the affection of the parameter setting.

2 Relative Works

2.1 Keyword Extracting Algorithm

Graph-based ranking algorithms are widely used in keyword extraction. These kind of algorithm firstly build the graphs according to the textual documents, and the important vertexes in these graphs are extracted to be the keyword. Many graph algorithms can be applied to find out the important vertexes in the graph. Mihalcea et al. apply PageRank [21], a famous websites ranking algorithm, to find out the important vertexes in [18], and they named these algorithm as TextRank. For the high performance the TextRank reach, there are many extension algorithms based on it. In [11], some term weighting schemes are added into TextRank to weight the vertexes. In [17], Liu et al. proposed Topical PageRank (TPR) where the vertexes are weighted according to the topic distribution generated by topic model. TPR has reached a high performance in

keywords extraction, but it doesn't consider the semantical information while the graph-building process. Hence, there are still large space to develop at the state-of-the-art algorithms.

There are some other research working on keyword extraction. In [8], another algorithm based on LDA is proposed. Different from [17], their algorithm do not base on graph. They rank words according to its Coverage, Purity, Phraseness, Completeness, and choose the words which have most ranking score as the keywords. Tomokiyo et al. proposed a statistical way to extract keywords from long text in [23]. They use KL-divergence [16] to calculate the information gain of the phrase. In [24], the relationship between document and words are modeled by a matrix, and then the matrix factorization algorithm is used to find out the latent topic of terms in the document. Finally, they extract keywords according to these latent topic.

2.2 Topic Model

Topic models are a suite of algorithms that uncover the hidden thematic structure in documents. Latent Dirichlet Allocation (LDA) is one of a generative topic model proposed by Blei et al. [2]. As a generative model, LDA assumes that the words in a document are drawn from a set of latent variable called topic which is a distribution over terms.

Collapsed Gibbs Sampling [5,9,10] is a algorithm commonly used to estimate latent parameter in LDA. One of the parameter φ represents the distribution from topic to terms. φ_{kt} represents the probability that term t is assigned to topic k. According to [12], the equation that calculate φ according to Collapsed Gibbs Sampling is as follows:

$$\varphi_{kt} = \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}(n_{k,\neg i}^{(t)} + \beta_t)}, \tag{1}$$

where notation $n_{k,\neg i}^{(t)}$ represents the count of word t assigned to topic k excluding the i^{th} word, V is the number of terms appeared in the corpora and β is the hyperparameters of the model. Equation 1 is applied in our approach to find out the most probable terms in a specific topic.

However, standard LDA trend to have a poor performance for topics which mix unrelated or loosely-related concepts. To tackle the problem, some knowledge-based topic models have been proposed [1,6,7]. $DF-LDA$ [1] takes domain knowledge in the form of must-links and cannot-links given by the users. A must-links means that two words should be assigned to the same topic whereas a cannot-links means that two words should not be assigned to the same topic. Besides, there are several models utilizing seed words provided by user [3,14,20]. In some recent research, for example, $GKLDA$ model [7] utilizes the general knowledge such as lexical knowledge to boost the performance. $GKLDA$ can also learn the domain knowledge provided by the user. We choose $GKLDA$ as our model in our approach, and we use the training data as the domain knowledge of $GKLDA$.

3 Existing Problem of Graph-Based Ranking Algorithm

Traditional approaches use slide window algorithm to construct the graph. The slide window moves from the first word to the last word in the document, and words which occur within a window are connected by an edge in the graph. A vertex with more in-edges and out-edges has more probability to be ranked a high score by PageRank algorithm. On the contrary, it also means that a word which have a low frequency is hard to be ranked a high score. However, many keywords in the article have a low frequency. For instance, an author will raise

Table 1. Example of keywords extraction via TextRank and our approach

Document
Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions.
Keywords
Keywords assigned by human annotators:
machine learning, computer science, artificial intelligence, data-driven predictions
Keywords assigned by TextRank:
machine learning, data-driven predictions, algorithms, study
Keywords assigned by our approach:
machine learning, artificial intelligence, date-driven, computer science

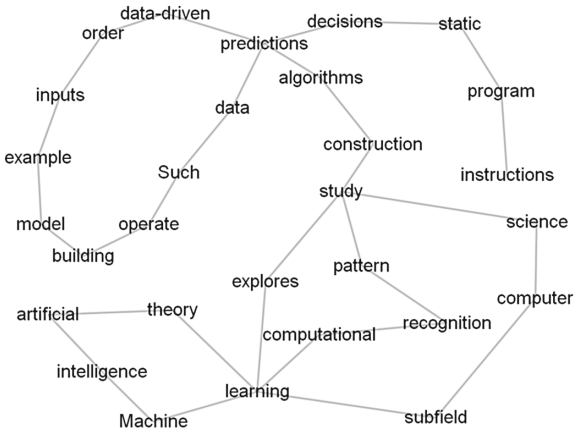


Fig. 1. Word graph of the example

a keyword in the beginning, and then he use several relevant terms to describe this keyword. We give an example in Table 1. Keywords like "computer science" and "artificial intelligence" appear once in the document, and they can not be find out by TextRank. From the graph of this document in Fig. 1, we find that word "computer", "science", "artificial" and "intelligence" have only both one in-edge and out-edge, and this will results the low ranking score of these words. The graph in Fig. 1 can only reflect the context information between words and it has nothing to do with the lexical meaning. Hence, we get a conclusion that the main problem in the state-of-the-art algorithm is the information loss in the transformation from textual document to graph. In the last of our paper, we argue that the semantic meaning must be considered when building the graph, and then we propose an algorithm overcome this problem.

4 Overall Algorithm

This section introduce the proposed overall algorithm. It consists three steps: *learning topical knowledge*, *extracting keywords using the learned knowledge*. Algorithm 1 shows our proposed algorithm. The input of our algorithm is D_L which consists of document D_i and its respective manual labeled keyword $W_i(D_i, K_i \in D_L)$. Another corpora is testing corpora D_T consisting different documents. w is a variable controlling the window size while creating the word graph. The topic model controls the topic number of the result using variable K. We represent each topic using the first S most relevant words. These inputs will be applied in the following steps.

Algorithm 1. The proposed algorithm.

Input: Topic learning corpora D_L; Test corpora D_T;
Window Size w; Topic number K; The length S of each topic set.

```
1  //STEP 1: Learning topic knowledge
2  Initialize GKLDA with keywords K_i ∈ D_L
3  LK ⟵ GKLDA(D_L, K)
4  //STEP 2: generate topic-based word graph
5  for each document D_i ∈ D_T:
6        do //generate word-graph using slide windows
7           G ⟵ graph(D_i, w)
8           for (t1, t2) in all two-tuples terms of document:
9                 do if (t1, t2) ∈ LK
10                      do G.addEdge(t1, t2)
11                End IF
12          End For
13          RW ⟵ PageRank(G)
14          TopN ⟵ sort(RW, N)
15 End For
```

Step 1 (learning topical knowledge): We obtain the topical knowledge using GKLDA model, a knowledge-based LDA model using general knowledge. In this step, we use the manual labeled keywords as the prior knowledge initiating GKLDA. Then we obtain topics represented by a set of related words when the model converges after several iterations. Finally, we deal with the obtained topics and change them into the set of two-tuples, which is the topical knowledge we need. We will introduce our knowledge representation in Sect. 5.

Line 2 in Algorithm 1 initialize GKLDA model with prior knowledge K_i, which is a set of keywords in document D_i. Line 3 learn the topical knowledge KL according to the results of GKLDA model whose topic number is set to K.

Step 2 (extracting keywords using the learned knowledge): Our algorithm proposed a new way to generate word graph from documents using the learned knowledge LK. We extract keywords from text corpora D_T according to PageRank algorithm with the learned knowledge. Firstly, we initialize the nodes of word graph and create edges using traditional moving windows algorithm. Secondly, the learned knowledge LK is used and words which appear at the same topic are linked with a edge. Finally, the normal PageRank algorithm is carried out and we can obtain the Top-N words as our keywords according to the ranking. Some post-processing are hidden in Algorithm 1, such as adjacent words combination, we will detail describe these process in Sect. 6.

For Line 5–15, we shows our disposal on each document D_i in test corpora D_T to extract keywords according to learned knowledge LK. Line 7 runs the traditional algorithm to generate word graph according to the order of words. We assigned a windows side w before running our algorithm. Line 8–12 show our proposed way to generate word graph based on the semantic relation between words. We first find out all the word pairs in D_i, and then these pairs are filtered by our learned knowledge LK(Line 9). The remaining words pairs shows a semantic similarity and a new edge is added between the two words in each pairs (Line 10). Line 13 conduct a normal PageRank algorithm to rank the nodes in graph G and obtain ranked words RW. Line 14 sort the ranked words according its ranking and choose Top-N words to become the keywords of D_i.

5 Learning Topical Knowledge

This section details Step 1 in the overall algorithm, which have two steps: GKLDA initialization and topical knowledge learning.

5.1 GKLDA Initialization

We can add domain knowledge into GKLDA model in order to obtain topics with high quality. The domain knowledge of GKLDA is represented by multiple set of words, e.g. price, cheap, expensive, which represents there are semantic relation between 'price', 'cheap' and 'expensive'. In our research, the manual labeled keywords in learning corpus D_L are used to generate domain knowledge. Multi-word keywords which consist of more than two words are separated into

single words. We put the single keywords of a specific document into the same set, which means that the keywords labeling the same document can have semantic relations. These generated set is used as domain knowledge to initialize GKLDA.

5.2 Topical Knowledge Learning

In this step, we run the GKLDA model and generate K topics. Each topic is represented by S most probable words according to topic-word distribution φ. From the generated K topic sets, we pair every two words in the set together and form the topical knowledge pairs, i.e. {"hardware","software"}. The topical knowledge is represented by the set of these pairs. Algorithm 2 details the generation of topical knowledge, where w_i^k represents the i^{th} words in topic set k.

Algorithm 2. Generate topical knowledge

```
1  for topic k in K topics:
2      do for i from 0 to S:
3          do for j from i + 1 to S:
4              do add pair (t_i^k, t_j^k) into knowledge set LK
5          End For
6      End For
7  End For
```

In general, two words of a pair in learned topical knowledge have some semantic relations because they belong to the same topic. However, research shows that some topics generated by topic model have a inferior quality [4]. In these bad topics, the semantic relation between each words is weak. Adding words in these bad topic into the topical knowledge will reduce the performance of our approach. There are some metric evaluating the quality of topics generated by topic model. Topic Coherent [19] is a metric commonly used to evaluate the performance of topic model, because it shows a well consistence with the judgement of human beings. We set a threshold δ to filter the bad topics. Words in topics whose topic coherence is less than δ will not be learned in topical knowledge. The topic coherence is calculated as

$$C(k, W^{(k)}) = \sum_{s=2}^{S} \sum_{l=1}^{s-1} log \frac{D(w_s^{(k)}, w_l^{(k)}) + 1}{D(w_s^{(k)})}, \tag{2}$$

Where $W^t = (w_1^t, ..., w_S^t)$ is a list of the first S most probable words in topic t and $D(v, v\prime)$ is the co-document frequency of word v and $v\prime$. Threshold δ is based on empirical value and is set before the algorithm.

6 Extracting Keywords Using the Learned Knowledge

Traditional way to extracting keywords is based on the order of words in a document. However, the order of words can just reflect the structure of the document, but can not show the opinion, emotion and implication behind the text. In our proposed approach, we transform documents into word graphs taking the topical knowledge into account. The topical knowledge learned in last step reflects the semantic information behind the documents.

Some post-processing are carried out in our approach. In most case, the keywords of the document are multi-words rather than single words. We combine the adjacent words which have high ranking in PageRank algorithm and form the multi-keywords.

We use the example showed in Table 1 again to demonstrate the usage of topical knowledge in a keywords extraction task. The traditional TextRank algorithm can not extract keywords such as "artificial intelligence" and "computer science". However, in our approach, we find that these two words are appeared at the same topic in topic models. Hence, the topical knowledge is extracted and showed as follows:

$$\{\{artificial, computer\}, \{artificial, science\},$$
$$\{intelligence, computer\}, \{intelligence, science\}\}$$

We add new edges into the graph showed in Fig. 1 and generate a new graph showed in Fig. 2. Finally, we find out the importance nodes in the graph to be our keywords using PageRank algorithm. The result of our approach is as below: machine learning, artificial intelligence, date-driven, computer science. Our approach can extract keywords precisely in this example.

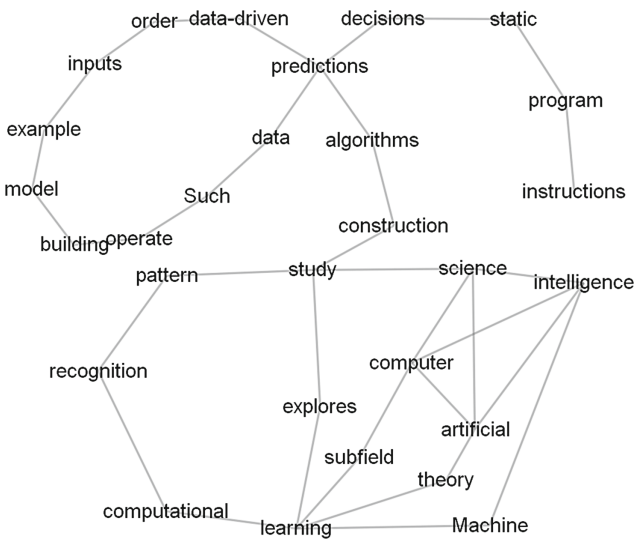


Fig. 2. Word graph of the example

7 Experiments

We prepared two datasets in our experiments. These two datasets are a collection of academic papers which contain the abstracts and the corresponding manually assigned keywords. One is the dataset used in the keyword extraction experiments in [13] and we denote it as $D1$. The academic papers in $D1$ are journal papers from Computer Science and Information Technology. We used 500 aspects and its corresponding keywords for learning topical knowledge, and 500 for extracting keywords. Another dataset is the abstracts of academic paper from ACM, we call it D2. The keywords in D2 assigned by the author himself. For our experiments, 500 papers of D2 is selected to complete the knowledge learning process, and 500 of them is used to the keyword extraction process.

Before testing our proposed algorithm, some preprocessing must be done in the datasets. Firstly, words were converted into lower case, so the words with upper case or lower were treated as the same words in our experiment. Secondly, all the punctuation in the document were eliminated and only the retain the alphabetic and numeric character. Thirdly, we performed text stemming and filter words with useless tagging. In this step, we used WordNet, a large lexical database of English, to perform POS tagging. In this work, we only used nouns, adjective and adverb in modeling, and other words were removed.

The parameters of $GKLDA$ were given default values in our experiments. The total Gibbs sampling iterations was set to 2500 with an initial burn-in of 100 iterations. For the reason that small changes of α and β will not affect the results much [15], we set $\alpha = 1$ and $\beta = 0.1$ as [7] do. Topic number K is a influence factor of our approach, so we individually discuss it in Sect. 7.2 and set $K = 20$ in other experiments.

In our experiments, we firstly compared our approach with some baseline algorithms in Sect. 7.1. Then we discussed the affection of parameters in our models. In Sect. 7.2, we looked into the performance our approach given different topic number K. In Sect. 7.3, the effect of topic-set size S was considered. Furthermore, the size of the slide window can also affect the result of the algorithm, and we discussed it in Sect. 7.4. The evaluation metric we used in all the experiments is F-measure which have also been used in [18].

7.1 Evaluation of Our Proposed Algorithm

In this section, we conducted several experiment to evaluate and compare the proposed algorithm with three baseline models i.e. TextRank, Topical TextRank and N-gram [13]. The first two algorithms are unsupervised and we use the testing part of both two datasets in our experiment. Besides, N-gram is a supervised keyword extraction algorithm. Hence, we carried out N-gram model with both training and testing data in two datasets.

We set the size of slide window w to 2, which showed the highest performance in [13]. The word number of each topic S is set to 10. The threshold σ was set to -1530 and 10 % of the topic was filtered.

Table 2. Topical knowledge of the example

Datasets	Assigned		Correct		Precision	Recall	$F-measure$
	Total	Mean	Total	Mean			
$D1$							
Topical graph TextRank	2,231	4.5	6,761	13.5	33.3	45.9	38.6
TextRank	2,003	4.0	6,803	13.6	29.4	40.9	34.2
Topical PageRank	2,124	4.2	6,728	13.4	31.6	43.3	36.5
N-Gram	1,952	3.9	7,644	15.3	25.5	39.8	31.1
$D2$							
Topical graph TextRank	1,302	2.61	5,044	10.1	25.8	37.7	30.7
TextRank	1,049	2.1	4,911	9.8	21.3	30.4	25.1
Topical PageRank	1,269	2.5	4,832	9.6	26.3	36.8	30.6
N-Gram	1,104	2.2	6,451	12.9	17.1	32.0	22.3

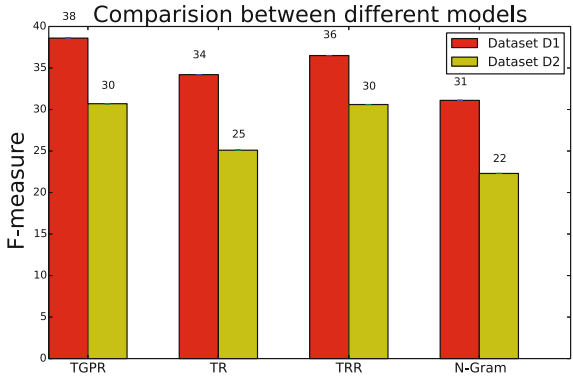


Fig. 3. Comparison of Topical graph PageRank (TGPR) with Topical PageRank (PR), Topical PageRank (TPR) and N-Gram

The detail results of our experiments are showed in Table 2, and we can compare different approach intuitively in Fig. 3. The result shows that the performance of our approach in both datasets $D1$ and $D2$ is better than others. Our approach can promote the traditional TextRank by approximately 15 %. The F-measure results of Topical PageRank are slightly surpassed by our approach. However, our approach is totally different from Topical PageRank, and the combination of these two approach may reach a more better result. We will extend our approach to combine with other approaches in our future works.

7.2 Effect of Topic Number K

For topic models such LDA, the topic number K can be adjusted in order to obtain higher performance. The value of K with which LDA obtain the best

performance is various when we deal with other corpora. In this section, we discuss how K affect our algorithm. We set the topic size K = 10, 20, 40, 60. The setting of other parameters was the same as Sect. 7.1.

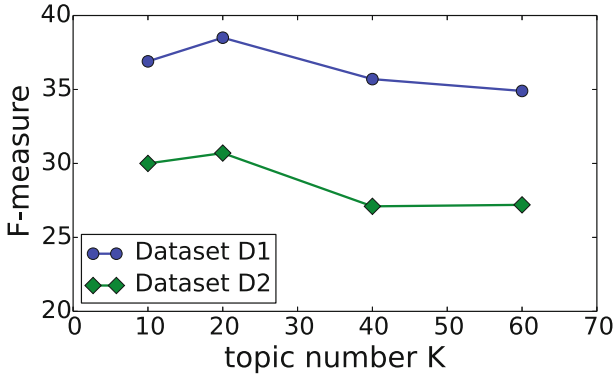
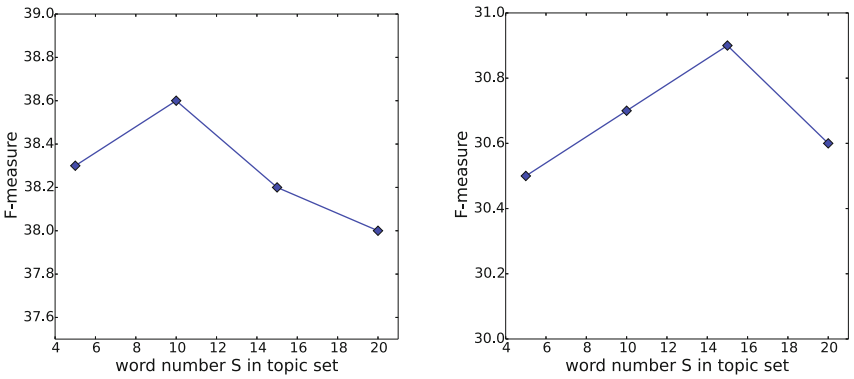


Fig. 4. Effect of different K value.

Figure 4 shows the F-measure of different K setting in two datasets. We observe that when $K = 20$, our approach reach a best performance. Another observation is that when K is larger than 40, the result of our approach become converge. These results are caused by the losing of topical information when the latent topic are divided too finely. Hence, we get a conclusion that lower or larger setting of topic number will both reduce the performance of our approach.

7.3 Effect of Size of Topic Set S

In the proposed approach, we choose the first S most probable words from each topics to represent the corresponding topic. Variable S is also an important factor



(a) Dataset $D1$ (b) Dataset $D2$

Fig. 5. The effect of different topic set length S

in our approach, because the topic information is incomplete when S is too low, while the topic will mix with irrelevant words when it is set a high value. S was set to 5, 10, 15, 20 in this experiment, and the other parameters were set to the default value as Sect. 7.1.

Figure 5 shows the results of different setting S in two datasets. We find that the result performed in dataset $D1$ and $D2$ is different. Figure 5(a) shows the best setting value is 10, while Fig. 5(b) is 15. We have the conclusion that the setting of topic size S is vary from different datasets.

7.4 Effect of the Window Size w

Window size w control the slide window size when constructing the word graph. In [18], the experiment shows that the model reach the best performance when w is set to 2. The objective of this experiment is to test whether our approach still work in different window size. We give w for $w = 1, 2, 3, 4$ in our experiment. In this experiment, we just use dataset $D1$, and we compare our approach with traditional TextRank.

Figure 6 shows the variation of different window size. We observed that both approach reach its best performance when the window size is set to 2. When the window size was larger than 8, both approach converge into a F-measure value near 32. It shows that our approach can not work will when the window size is too large. In order to find out the reason about that, we looked into the situation when the window size is very large. In that situation, the number of edges will become very large. However, the number of edge added to the graph according to our approach is limited. Hence, our adding edge will not dominate the PageRank algorithm when the window size is too large.

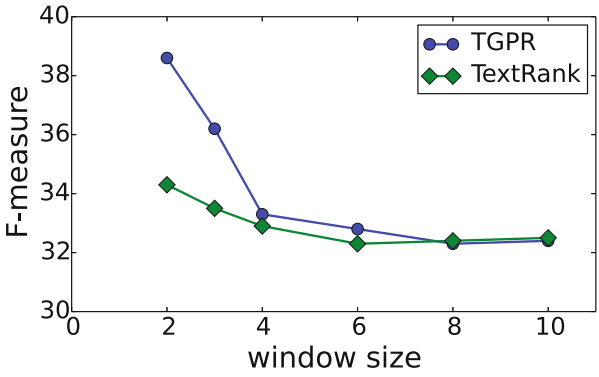


Fig. 6. Effect of window size

8 Conclusions

The objective of our research is to promote the performance of graph-based algorithm in keyword extraction. We solve the problem that the semantic information

is ignored during constructing the graphs. We raise a topic based approach to find out the latent semantic relationship between words. Finally, we have conduct a series of experiments to evaluate the effect of our proposed algorithm. It shows that our approach have a outstanding performance comparing to the state-of-the-art algorithm. The results also shows that the semantic information is necessary to be considered while building graphs of documents.

9 Future Works

In future work, we would like to extend our approach to the variants of TextRank or other graph-based algorithms. There are many variants which can be considered such as Topical PageRank. We also want to apply our algorithm for summarization extraction. It needs some further enhancements to fit with this task.

Acknowledgement. This work is supported by National Natural Science Foundation of China (project no. 61300137), and NEMODE Network Pilot Study: A Computational Taxonomy of Business Models of the Digital Economy, P55805.

References

1. Andrzejewski, D., Zhu, X., Craven, M.: Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 25–32. ACM (2009)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
3. Burns, N., Bi, Y., Wang, H., Anderson, T.: Extended twofold-LDA model for two aspects in one sentence. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R. (eds.) Advances in Computational Intelligence. CCIS, vol. 298, pp. 265–275. Springer, Heidelberg (2012)
4. Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J.L., Blei, D.M.: Reading tea leaves: how humans interpret topic models. In: Advances in Neural Information Processing Systems, pp. 288–296 (2009)
5. Chatterji, S., Pachter, L.: Multiple organism gene finding by collapsed Gibbs sampling. In: Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology, pp. 187–193. ACM (2004)
6. Chen, Z., Mukherjee, A., Liu, B.: Aspect extraction with automated prior knowledge learning. In: Proceedings of ACL, pp. 347–358 (2014)
7. Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., Ghosh, R.: Discovering coherent topics using general knowledge. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 209–218. ACM (2013)
8. Danilevsky, M., Wang, C., Desai, N., Ren, X., Guo, J., Han, J.: Automatic construction and ranking of topical keyphrases on collections of short documents. In: Proceedings of the SIAM International Conference on Data Mining, 2014 (2014)

9. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 721–741 (1984)
10. Griffiths, T.: Gibbs sampling in the generative model of latent Dirichlet allocation. Technical report, Stanford University (2002)
11. Hassan, S., Mihalcea, R., Banea, C.: Random walk term weighting for improved text classification. Int. J. Semant. Comput. **1**(04), 421–439 (2007)
12. Heinrich, G.: Parameter estimation for text analysis. Technical report (2005)
13. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 216–223. Association for Computational Linguistics (2003)
14. Jagarlamudi, J., Daumé III, H., Udupa, R.: Incorporating lexical priors into topic models. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 204–213. Association for Computational Linguistics (2012)
15. Jo, Y., Oh, A.H.: Aspect and sentiment unification model for online review analysis. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 815–824. ACM (2011)
16. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**, 79–86 (1951)
17. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 366–376. Association for Computational Linguistics (2010)
18. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain, pp. 404–411 (2004)
19. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 262–272. Association for Computational Linguistics (2011)
20. Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers. vol. 1, pp. 339–348. Association for Computational Linguistics (2012)
21. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120
22. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Documentation **28**(1), 11–21 (1972)
23. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, vol. 18, pp. 33–40. Association for Computational Linguistics (2003)
24. Yan, X., Guo, J., Liu, S., Cheng, X., Wang, Y.: Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: Proceedings of the SIAM International Conference on Data Mining (2013)