Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 15, No. 3, September 2015, pp. 153-158
http://dx.doi.org/10.5391/IJFIS.2015.15.3.153

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Latent Keyphrase Extraction Using Deep Belief Networks

Taemin Jo and Jee-Hyong Lee
Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea

Abstract

Nowadays, automatic keyphrase extraction is considered to be an important task. Most of the previous studies focused only on selecting keyphrases within the body of input documents. These studies overlooked latent keyphrases that did not appear in documents. In addition, a small number of studies on latent keyphrase extraction methods had some structural limitations. Although latent keyphrases do not appear in documents, they can still undertake an important role in text mining because they link meaningful concepts or contents of documents and can be utilized in short articles such as social network service, which rarely have explicit keyphrases. In this paper, we propose a new approach that selects qualified latent keyphrases from input documents and overcomes some structural limitations by using deep belief networks in a supervised manner. The main idea of this approach is to capture the intrinsic representations of documents and extract eligible latent keyphrases by using them. Our experimental results showed that latent keyphrases were successfully extracted using our proposed method.

Keywords: Latent keyphrase, Deep belief networks, Weighted cost function, Keyphrase extraction

1. Introduction

As the number of resources for documents is growing continuously, our need to acquire useful information from them is also growing everyday. Keyphrase, which is the smallest unit of useful information, can concisely describe the meaning of content in documents. Moreover, keyphrases can also be used in text mining applications like information retrieval, summarization, document classification, and topic detection. However, only a small portion of documents contains author-assigned keyphrases and a majority of documents do not have keyphrases. Therefore, extracting keyphrases from documents has become one of the main concerns in recent days, and there have been several studies on automatic keyphrase extraction task [1–14].

Most of the previous studies focused only on selecting keyphrases within the body of input documents. These studies overlooked latent keyphrases that did not appear in documents, extracted candidates only from the existing phrases in the document, and evaluated them under the assumption that they appear in the document. Therefore, those methods were not suitable for the extraction of latent keyphrases. In addition, a small number of studies on latent keyphrase extraction methods had some structural limitations. Although latent keyphrases do not appear in documents, they can still undertake an important role in text mining as they link meaningful concepts or contents of documents and can be utilized in short articles such as

Received: Aug. 21 2015
Revised : Sep. 22, 2015
Accepted: Sep. 24, 2015

Correspondence to: Jee-Hyong Lee
(ejohn@skku.edu)

social network service (SNS), which rarely have explicit keyphrases.

In this paper, we propose a new approach that selects reliable latent keyphrases from input documents and overcomes some structural limitations by using deep belief networks (DBNs) in a supervised manner. The main idea of this approach is to capture the intrinsic representations of documents and extract eligible latent keyphrases by using them. Additionally, a weighted cost function is suggested to handle the imbalanced environment of latent keyphrases compared to the candidates.

The remainder of this paper is organized as follows. Section 2 provides a brief description of previous methods in relation to keyphrase extraction. Section 3 provides a background on the proposed method. Section 4 introduces a method of latent keyphrase extraction. Section 5 describes the experimental environment and evaluates the result. Section 6 provides a conclusion inferred from our work and indicates the direction of future research.

2. Related Work

The algorithms for keyphrase extraction can be roughly categorized into two type: supervised and unsupervised. Initially, most of the previous extraction methods focused only on selecting the keyphrases within the body of input documents.

Supervised algorithms proposed a binary approach, that is, determine whether a candidate is a keyphrase or not. In general, supervised algorithms extracted multiple features from each candidate and applied machine learning techniques such as naive Bayes [1], support vector machine [2], and conditional random field [3]. The commonly used features were TF-IDF [4], the relative position of the first occurrence of a candidate in the document [1], and whether a candidate appeared in the title or subtitle [2]. However, these features were extracted under the assumption that the candidates appear in the document, so these algorithms are not suitable to evaluate and select latent keyphrases.

In the case of an unsupervised algorithm, a notable approach was to use a type of graph ranking model called, TextRank [5]. The major idea of this approach was that if a phrase had strong relationships with other phrases, it was an important phrase in the document. This algorithm marked the phrases of the document as vertexes and assessed each vertex with their connected links, which was called a co-occurrence relationship. Subsequently, this algorithm was expanded in a variety of ways [6, 7]. However, again, such algorithms only selected the existing phrases from documents as candidate phrases, and latent keyphrases that does not appear in documents have no likelihood of being selected under the set of final keyphrases. In addition, they evaluated the candidates with co-occurrence relationship that assuming candidate appear.

However, to the best of our knowledge, there have been four studies that handle latent keyphrases. Wang et al. [8] considered latent keyphrases as abstractive keyphrases. Their algorithm conjugated single word embeddings as an external knowledge to select semantically similar word embeddings with the document embedding as abstractive keyphrases. Cho et al. [9] extracted primitive words that are important single words in the document and combined the two contextually similar primitive words as latent keyphrases. However, both methods had a limitation on length. They could only select single word keyphrases or two-word keyphrases. Liu et al. [10] tackled the translation problem between title/abstract and body so as to evaluate the importance of a single word and assessed candidates by summing up the importance of their components. However, this algorithm had a problem when it came to overlapping candidates, which means one candidate included in the other candidate. Similarly, Cho and Lee [11] used latent dirichlet allocation (LDA) to evaluate the importance of single words by considering topics of document and assessed candidates by calculating the harmonic mean of their component's importance. By averaging, this method alleviated the overlapping problem; however, it did not consider the relationship between components during averaging. As a result, the performance deteriorated with a little bit much of candidates.

As described above, previous studies on latent keyphrases had some structural limitations. To handle these problems, this study considers a candidate as one complete element, and not separate words. With the one complete element perspective, we can have varying lengths of candidates and avoid the relationship issue.

3. Background

3.1 Latent Keyphrase

In this paper, a latent keyphrase is defined as a keyphrase that does not appear in the document. Most previous works gave little consideration to latent keyphrases. In addition, those studies treated latent keyphrases as missing or inappropriate keyphrases, thereby eliminating the latent keyphrases from the answer set or excluding when evaluating.

Figure 1 shows three documents that have $w_1 w_2$, a two-word phrase(e.g. This is just an example. Latent keyphrases can be

composed with one word or more than two words like normal keyphrases), as keyphrases. In document Figure 1(a), w_1 and w_2 appear together, however, in documents Figure 1(b), w_1 and w_2 are seperate from each other, and in document Figure 1(c), only w_1 appears. For the latter cases, we call $w_1 w_2$ as a latent keyphrase.

Although latent keyphrases do not appear in documents, they can still undertake an important role in text summarization and information retrieval because they link meaningful concepts or contents of documents. Latent keyphrases cover more than one-fourth of the keyphrases in real-world datasets [14–16] and can be utilized in short articles such as SNS, which rarely have explicit keyphrases.

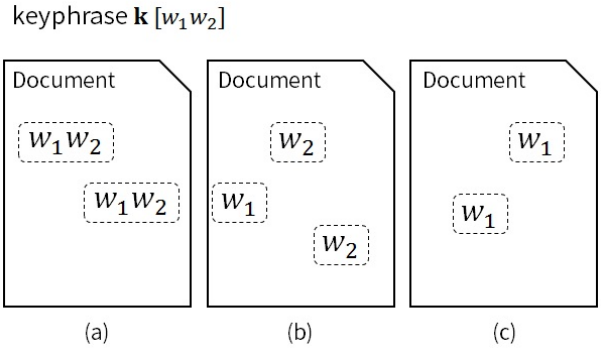

Figure 1. Comparison of explicit keyphrase and latent keyphrase.

3.2 Deep Belief Networks (DBNs)

Hinton et al. [17] introduced a greedy layer-wise unsupervised learning algorithm for DBNs. This training strategy for deep networks is an important ingredient for effective optimization and training of deep networks. While lower layers of a DBN extract low-level factors from the inputs, the upper layers are considered to represent more abstract concepts that explain the inputs.

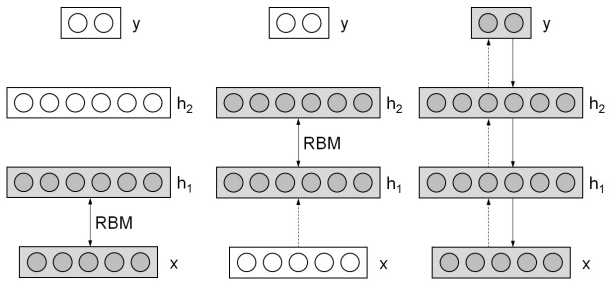

Figure 2. Deep belief network (DBN) training procedure.

DBNs are pre-trained by multiple restricted boltzmann machine (RBM) layers, and then, fine-tuned, which is similar to back-propagating networks. The entire procedure of training DBNs is illustrated in Figure 2.

4. Proposed Method

In this section, we introduce our proposed method for latent keyphrase extraction that uses the DBNs and a logistic regression layer. The main idea of this approach is to capture the intrinsic representations of documents and extract eligible latent keyphrases by using them. The inputs of the DBNs are 0 or 1 of bag-of-words representation of the input document and the outputs of the logistic regression layer are candidate phrases. Figure 3 shows a simple structure of the algorithm.

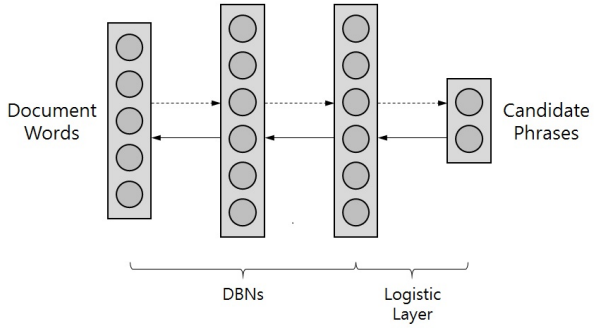

Figure 3. Deep belief network (DBN) training procedure.

For inputs, we do not use all of the words in a document set. As the corpus is generally composed of the same type of documents, they share words that are commonly used but have meaningless information. These common words, similar to stopwords, may act as noise and disturb the DBNs from capturing the intrinsic representations. Therefore, a number of r frequently occurring words are eliminated and the remaining ones become 0 or 1 bag-of-words representation of inputs of the DBNs.

The outputs of the logistic regression layer are candidate phrases. The candidate phrase indicates a phrase that has the possibility of becoming a final keyphrase. Therefore, for phrases that do not appear in the document to become final keyphrases, candidate set must have phrases that do not appear in the input document. However, the input document has limited information to provide various form of phrases, so there is a need to utilize other information beyond the input document. In this study, all of the answer keyphrases of the corpus are used as candidate phrases for outputs of the logistic regression layer. And each candidate is considered as one complete element to

overcome some structural limitations like varying length of candidates and the relationship issue.

The pre-training process of the DBNs are similar to the method developed by Hinton et al. [17]; however, the fine-tuning process is slightly different. Because the number of answer keyphrases is less than that of the candidates, we require the DBNs to train more dependent on the answer keyphrases. Therefore, we apply the weighted cost function shown in Eq. (1). This equation is a variation of mean squared error. In Eq. (1), py denotes predicted vector; y, answer vector; λ, damping factor; and D, the document set. When $\lambda > 0.5$, the DBNs can be trained more dependent on the answer latent keyphrases.

$$L = \sum^{D} \lambda((1-py)y)^2 + (1-\lambda)(py(1-y))^2 \qquad (1)$$

5. Experiments

5.1 Experimental Environment

5.1.1 Dataset description

The INSPEC database stores abstracts of journal papers belonging to computer science and information technology fields. Hulth [14] built the dataset using English journal papers from the years 1998 to 2002. Each document has two kinds of keyphrases: controlled keyphrases, which are restricted to a given dictionary, and uncontrolled keyphrases, which are freely assigned by the experts. Because uncontrolled keyphrases have lots of appearing only once keyphrases and these keyphrases cannot be found by supervised method, controlled keyphrases are used for this experiment. The following Table 1 shows the distribution of controlled keyphrases.

Table 1. Distribution of controlled keyphrases (%)

No. of words	1	2	3	4	5	Total
Explicit	12.08	10.13	1.64	0.07	0.00	23.93
Latent	9.75	48.20	16.38	1.71	0.03	76.07
Total	21.84	58.33	18.02	1.78	0.03	100.00

The entire dataset has 1,500 training and 500 testing documents; however, we exclude small documents that are composed of less than 70 words. Therefore, 1,165 training and 376 testing documents are used for the experiment. Each document has 3.68 latent keyphrases in average.

5.1.2 Text Preprocessing

The purpose of preprocessing is to normalize the documents. The following modifications are applied to the entire corpus:

- Stopwords such as prepositions, pronouns, and articles are eliminated by referencing the commonly used stopword list [18].
- Only the tagged words NN, NNS, NNP, NNPS and JJ by the Stanford log-linear POS tagger [19] are used by following Hulth [14] that the majority of the keyphrases was noun phrases with adjective or noun modifier.
- Porter stemmer [20], which is commonly used in keyphrase extraction field, is used for the experiment.

5.1.3 Parameter Setting

As the DBNs has various parameters to determine, it is very hard to test all the cases of parameter settings and find the best one. Therefore, the guidelines by Hinton [21] are initially adopted for the experiments and later some modifications are done. Finally, the DBNs have three hidden layers with nodes of 1,300 each. The epochs of pre-training and fine-tuning are equally set to 150, learning rate is set to 0.01 and 0.1 each. The batch size is set to 10. Additionally, the number of input and output nodes is set to 9,784 and 1,921, respectively, following the corpus. The number of eliminated common words is 100. Explicit keyphrases are excluded for the valid evaluation after the training step, as the proposed method only targets latent keyphrases. Theano [22] based DBNs for classifying the MNIST digits are modified for the experiments.

5.2 Experimental Results

This section gives an evaluation of the proposed method, latent keyphrase extraction. The results are presented on the Figure 4 with the paper of Cho and Lee [11], which is for baseline. They proposed a latent keyphrase extraction method using LDA. The main ideas of the baseline are extracts candidate phrases by referencing neighbor documents and evaluates words of each candidate by considering topic.

Figure 4 shows the result with varying λ. If λ is high, the proposed method is mainly trained on answer latent keyphrases than other candidates in fine-tuning stage. We can see the proposed method performed better than the baseline in feasible cases with λ ranging from 0.5 to 0.9. The F1 score of the best matching latent keyphrase is 0.108, when λ is 0.9. These results

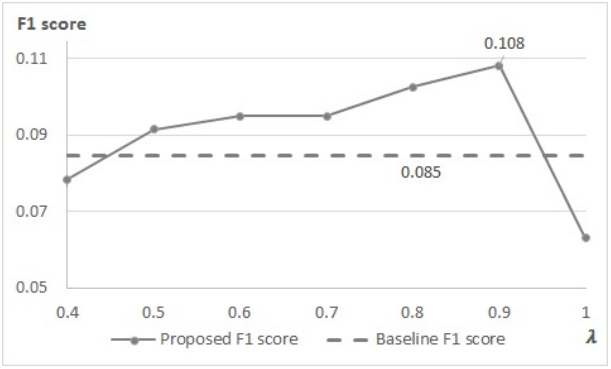

Figure 4. Performance of latent keyphrase extraction.

show that latent keyphrases can be extracted by the proposed method at a reasonable level.

6. Conclusion

This study focused on selecting qualified latent keyphrases of documents using DBNs in a supervised manner. The main idea of this approach was to capture the intrinsic representations of documents and extract eligible latent keyphrases by using them. Our experimental results showed that latent keyphrases can be extracted using our proposed method. Additionally, a weighted cost function is suggested to handle the imbalanced environment of latent keyphrases compared to the candidates. A more complex structure of deep learning with word embeddings is presumed to deliver better performance. This can be a part of future work.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This work was supported by the ICT R&D program of MSIP/IITP (B0101-15-0559, Developing On-line Open Platform to Provide Local-business Strategy Analysis and User-targeting Visual Advertisement Materials for Micro-enterprise Managers). Also, this research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014M3C4A7030503).

References

[1] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-specific keyphrase extraction," *Proceedings of the 16th international joint conference on artificial intelligence*, 1999, pp. 668-673. http://researchcommons.waikato.ac.nz/handle/10289/1508

[2] K. Zhang, H. Xu, J. Tang, and J. Li, "Keyword extraction using support vector machine," *Proceedings of the 7th international conference on web-age information management*, 2006, pp. 86-96. http://link.springer.com/chapter/10.1007/11775300_8

[3] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao, and B. Wang, "Automatic keyword extraction from documents using conditional random fields," *Journal of Computational Information System*, vol. 4, no. 3, 2008, pp. 1169-1180. http://eprints.rclis.org/handle/10760/12305

[4] G. Salton, and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, 1988, pp. 513-523. http://www.sciencedirect.com/science/article/pii/0306457388900210

[5] R. Mihalcea, and P. Tarau, "Textrank: bringing order into texts," *Association for Computational Linguistics*, 2004. http://digital.library.unt.edu/ark:/67531/metadc30962/

[6] X. Wan, and J. Xiao, "Single Document Keyphrase Extraction Using Neighborhood Knowledge," *Association for the Advancement of Artificial Intelligence*, vol. 8, 2008. http://www.aaai.org/Papers/AAAI/2008/AAAI08-136.pdf

[7] Z. Liu, W. Huang, Y. Zheng, and M. Sun, "Automatic keyphrase extraction via topic decomposition," *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, 2010. http://dl.acm.org/citation.cfm?id=1870694

[8] R. Wang, W. Liu, and C. McDonald, "Using Word Embeddings to Enhance Keyword Identification for Scientific Publications," *Databases Theory and Applications*, 2015, pp. 257-268. http://link.springer.com/chapter/10.1007/978-3-319-19548-3_21

[9] T. Cho, H. Cho, J. Lee, and J. H. Lee, "Latent keyphrase generation by combining contextually similar primitive words," *Joint 7th International Conference on Soft Computing and Intelligent Systems and The 15th International*

Symposium on Advanced Intelligent Systems, 2014, pp. 600-604. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7044871

[10] Z. Liu, X. Chen, Y. Zheng, and M. Sun, "Automatic keyphrase extraction by bridging vocabulary gap," *Proceedings of the Fifteenth Conference on Computational Natural Language Learning. Association for Computational Linguistics*, 2011. http://dl.acm.org/citation.cfm?id=2018952

[11] T. Cho, and J. H. Lee, "Latent Keyphrase Extraction Using LDA Model," *Journal of The Korean Institute of Intelligent Systems*, vol. 25, no. 2, 2015, pp. 180-185. http://www.dbpia.co.kr/Journal/ArticleDetail/NODE06277944

[12] J. H. Kim, Q. Gao, and Y. I. Cho, "A Context-Awareness Modeling User Profile Construction Method for Personalized Information Retrieval System," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 14, no. 2, 2014, pp. 122-129. http://www.dbpia.co.kr/Journal/ArticleDetail/3468702

[13] S. Rho, B. Kim, and N. Huh, "Representative Keyword Extraction from Few Documents through Fuzzy Inference," *Journal of The Korean Institute of Intelligent Systems*, vol. 11, no. 9, 2001, pp. 837-843. http://www.dbpia.co.kr/Journal/ArticleDetail/NODE01008078

[14] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," *Proceedings of the 2003 conference on Empirical methods in natural language processing. Association for Computational Linguistics*, 2003. http://dl.acm.org/citation.cfm?id=1119383

[15] M. Krapivin, A. Autaeu, and M. Marchese, "Large dataset for keyphrases extraction," *Technical Report DISI-09-055*, 2009. http://eprints.biblio.unitn.it/1671/

[16] S. N. Kim, O. Medelyan, M. K. Kan, and T. Baldwin, "Semeval-2010 task 5: automatic keyphrase extraction from scientific articles," *Proceedings of the 5th International Workshop on Semantic Evaluation. Association for Computational Linguistics*, 2010. http://dl.acm.org/citation.cfm?id=1859668

[17] G. E. Hinton, and O. Simon, and T. Y. The, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, 2006, pp. 1527-1554. http://www.mitpressjournals.org/doi/abs/10.1162/neco.2006.18.7.1527

[18] "Stop Word List 1," Available http://www.lextek.com/manuals/onix/stopwords1.html

[19] K. Toutanova, and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," *Association for Computational Linguistics*, vol. 13, 2000. http://dl.acm.org/citation.cfm?id=1117802

[20] M. F. Porter, "An algorithm for suffix stripping. Program: electronic library and information systems," vol. 14, no. 3, 1980, pp. 130-137. http://www.emeraldinsight.com/doi/abs/10.1108/eb046814

[21] G. E. Hinton, "A practical guide to training restricted boltzmann machines," *Neural Networks: Tricks of the Trade*, 2012, 599-619. http://link.springer.com/chapter/10.1007/978-3-642-35289-8_32

[22] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," *Proceedings of the Python for scientific computing conference*, vol. 4, 2010. https://projects.scipy.org/scipy2010/slides/james_bergstra_theano.pdf

Taemin Jo received his B.S. in computer engineering from Sungkyunkwan University, Korea in 2014. He is currently pursuing his M.S. in computer engineering at Sungkyunkwan University. His research interests include text mining and machine learning.
E-mail: tmchojo@skku.edu

Jee-Hyong Lee received his B.S., M.S., and Ph.D. in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1993, 1995, and 1999, respectively. From 2000 to 2002, he was an international fellow at SRI International, USA. He joined Sungkyunkwan University, Suwon, Korea, as a faculty member in 2002. His research interests include fuzzy theory and application, intelligent systems, and machine learning.
E-mail: john@skku.edu