Predicting Abstract Keywords by Word Vectors

Qing Li[1], Wenhao Zhu[1], and Zhiguo Lu[2](✉)

[1] School of Computer Engineering and Science,
Shanghai University, Shanghai, China
[2] Shanghai University Library, Shanghai University, Shanghai, China
luzg@staff.shu.edu.cn

Abstract. The continuous development of the information technology leads to the explosive growth of many information domains. Obtaining the required information from a large-scale text in a quick and accurate way has become a great challenge. Keyword extraction is a kind of effective method to solve these problems. It is one of the core technologies in the research area of text mining, and plays a very important role. Currently, the keywords of most text information have not been provided. Some keywords of a text are not contained in the text content. There is not any elegant solution, offered by the existing algorithms, for this problem yet. To solve it, this paper proposes a keyword extraction method based on word vectors. The concept of a text turns into computer understandable space by training word vectors using a word2vec algorithm. This method trains all the words and keywords which appear in the text into vector sets through the word2vec training method, and then the words in the test text will be replaced by word term vectors. The Euclidean distances between every candidate words and every text words are calculated to find out the top-N-closest keywords as the automatic text extraction keywords. The experiment uses computer field papers as a training text. The results show that the method can improve the accuracy of the phrase keyword extraction and find the keywords not appearing in the text.

Keywords: Keyword extraction · Semantic analysis · Word vector · Word2vec

1 Introduction

With the rapid development of the information technology, huge amounts of text information is electronical. How to get useful information from these digital resources quickly and accurately is becoming an important topic. Text data mining, a branch of data mining, is a computer processing technology which aims to extract valuable information and knowledge from a text. The main methods include but are not limited to text classification, clustering, and information extraction. A keyword automatic extraction technology is an important branch in the field of text data mining. It is the most effective way to solve the problem of massive text retrieval. It is also the basic work of document retrieval, document comparison, summarization generation, document classification and clustering. Keywords summarize the theme of the article

J. Xie et al. (Eds.): HPCA 2015, LNCS 9576, pp. 185–195, 2016.
DOI: 10.1007/978-3-319-32557-6_20

information, and help the reader quickly grasp the gist. It has obvious practical significance for its great improvement of the efficiency of information access. However, most text information has not yet provided keywords. The traditional manual method has high accuracy, but its efficiency is low. On the other hand, the computer automatic extracting keywords method can have very high efficiency with low accuracy. At the same time, the existing automatic keyword extraction algorithms are still faced with some problems, such as redundant expression, polysemy, synonyms thesaurus updating dynamically, and interdisciplinary content complexity.

There is another problem in automatic keyword extraction. In the actual research and application, quite a part of keywords includes phrases which are difficult to extract. Phrases have more generalization capability than words and contain more abundant information, so the extraction of keyword phrases is more meaningful. In most of the keywords extraction algorithms, such as in [2, 4] in which consecutive sequences of a few words in the text, as a candidate for a keyword phrase, are highly regarded. But a problem that a sequence of these words are not in accordance with the approved phrase form is not fully considered. In [8], a separation model is described. The method of a separating process for the keywords and phrases and designing different characteristics to improve the accuracy of extraction is also introduced. The promotion effect of keyword phrases is obvious, while the effect of the whole keyword extraction is less than a traditional keyword extraction algorithm.

Most of the preceding automatic keyword extraction algorithms rely on the manual feature selection. The classical features are given in a comprehensive introduction in [7]. However, the process of heuristic feature selection needs prior knowledge, and it is the most time-consuming part of the whole system. Deep Learning, also called Unsupervised Learning, is a new field of machine learning, and the motive [17] is to simulate a human brain mechanism to interpret and analyze data to study a neural network. The distributed data characteristics is found by a combination of the bottom-layer feature and a more abstract high-layer category or feature. The method constructs a machine learning model which has a number of hidden layers and vast amounts of training data to learn more useful features automatically, and then to eliminate manual feature selection process. In the age of big data, all that needed is to put huge amounts of data in an algorithm directly. Let data speak for themselves, and then the system will automatic study from the data. The biggest breakthrough of Deep Learning is in the field of voice and image recognition. In 2013, in Google's open source word2vec tools [9], using the ideas of deep learning through the training, the text processing is simplified to K-dimensional vector operations. The similarity of a vector space is used to represent the text semantic similarity by regarding the words as features. Word2vec can map those features to a K-dimensional vector space and seek deeper-layer features for the text data.

Based on the above methods and encountered difficulties, we believe that the words which contain large amounts of semantic information will be extracted as candidate keywords for the article. Therefore, with the help of the deep learning method, we use word2vec tools to train the term vectors. The smaller the Euclidean distance between two words means the closer semantic meaning. Through vector calculation between test words and a keywords set, we can choose the keywords most of which are representative of the full text semantic information. The method we employ can be a very good solution to the polysemy and synonyms problem. Meanwhile, our experiment adopts the distributed method, and it can achieve good results in a relatively short time. The following sections are arranged as below. The second section discusses the related research to automatic keyword extraction and the word2vec method. The third section introduces our algorithm and experimental method. The fourth section describes the comparison between the method proposed in this paper and other two classic keyword extraction algorithms, and then analyzes the experimental results. Finally, conclusions and prospective discussions are given in Sect. 5.

2 Related Work

Automatic keyword extraction is a process which analyze the article and extract the keywords that can express the main idea of the article according to a certain proportion. The researchers have already obtained a lot of achievements. Literature [18] is the first paper which describes the study of text annotation. Since then, researchers has been working on the automatic keyword extraction technology, which is based on the technology of text annotation for 50 years. And in recent years, there are mainly 3 directions to the study of the area of automatic keyword extraction: 1. statistical methods; 2. Machine learning method 3. Semantic analysis method.

Word2vec [9], whose source code has been released by Google in 2013, is an efficient tool that can convert text words to real value vector. Based on deep learning, Word2vec can analyze the nature language and convert words to vectors. The method can get vector representation terms by building a vocabulary from training text data. These vectors can be used in many natural language processing and machine learning researches. In this way, we can transform text content space to vector space and conduct vector operations [10]. The Euclidean distance between vectors can be calculated easily, which can represent the text semantic similarity.

2.1 Automatic Keyword Extraction Method

As mentioned in the [19], automatic keyword extraction method is mainly divide into the following three categories:

1. Statistics methods, including frequency, TF-IDF and other statistical information. Literature [3] put forward a kind of improved tf-idf extraction method. The method combines high similarity words with paragraph annotation technology and selects the candidate keywords with higher weight by using the word inverse frequency tf-iwf algorithm. The extraction accuracy of the method tends to be low, though it could be more applicable and feasible.
2. Researchers have made some achievements in the area of machine learning. The KEA system, mentioned in Literature [4], is a kind of supervised machine learning methods. Using simple Bayesian technique, the method train and set up a predicting model with candidate phrases and eigenvalue gained before. And it can extract keywords from documents with the model. Other methods like random model and maximum entropy model [5], etc. can also get the same result. However, there are still problems like imbalance between labeled samples in different level, as

well as the limitation of extraction accuracy caused by the over-fitting problem which is existed in the training process of classifier constructing.

3. The study of semantic analysis methods, including speech, grammar and semantic dependency has attracted broadly attention. The methods mentioned in [6], composes the semantic chain according to the semantic similarity, and obtain the keywords with semantic feature analysis. Compared to the former methods, it can mine the potential semantic information in a deeper layer. The quality of keyword extraction is also higher.

2.2 Word Vector

Word vector is used for presenting the digitized terms in the natural language so as to process the natural language.

The original representation method of word vector is one-hot representation. The method suggests that each word can be regard as a length vector. The dimension of the vector is equal to the vocabulary size. Number 1 shows on the corresponding position. But this method cannot capture the similarity between words, and it can easily lead to the curse of dimensionality.

Distributed representation is put forward in Literature [14]. It maps each word to a K-dimensional real vector, and the semantic similarity is can be determined by word distance (cos similarity, Euclidean distance).

There are mainly 3 categories of the word vector generation model:

1. The statistical language model, including context-independent model, n-gram model, decision tree based language model, etc. The neural network language model is described in Literature [11]. The model convert each word to a floating point vector with Distributed representation;
2. Hierarchical language model, including hierarchical probabilistic language model [15] and hierarchical Log-bilinear model [14], is a binary decision-making tree whose leaf node is word.

Word2vec is based on the new Log-Linear Model.

2.3 Word2vec Model

Word2vec proposes two important models: CBOW(continuous bag-of-words model) and skip-gram model.

CBOW [9] is a model to predict the probability of $P\left(w_t | w_{t-c}, w_{t-(c-1)} \cdots, w_{t-1}, w_{t+1}, w_{t+2} \ldots, w_{t+c}\right)$. Using hierarchical training strategy, every word is represented by c words selected from its context, c is the size of primary window. The structure of the method consists of hidden layer, input layer and output layer. The input layer is used for initializing the term vectors (obtain with one-hot representation method). The sum of vector accumulation is calculated in the hidden layer. And the output layer display the results with Huffman binary tree. The left children represent the probability of the vector of the word is in front of its parent node, and the right child of the parent node represent the probability of this word behind its parent node. All the non-leaf nodes in the output layer is connected with the node in the hidden layer, and the leaf node is the output vector.

Skip-gram model [9] and CBOW model are just at the opposite side. It predict the probability of $p(w_i|w_t)(t - c \leq i \leq t + c,\ i \neq t)$, c is the size of the context window. The input layer of this method is a single word, which is connected with the Huffman tree. The method has no hidden layer.

The text can be convert to word vector with Word2vec through training. The output vectors can be used for clustering, synonyms, speech analysis and so on. In Literature [11], the authors conduct addition operation using word vector obtained through training. For example, vector('king') − vector('man') + vector('woman') ≈ vector ('queen'), which can reflect the fact that vector space can be used to indicate the similarity of text semantic.

3 Methodology

This section describes the main structure of our algorithm including two parts: First, the procedure of text library; Second, the procedure of testing text. The filter of Punctuation, numbers, stop words and part of speech are included in both of the mentioned parts. The overall process is shown in Fig. 1.

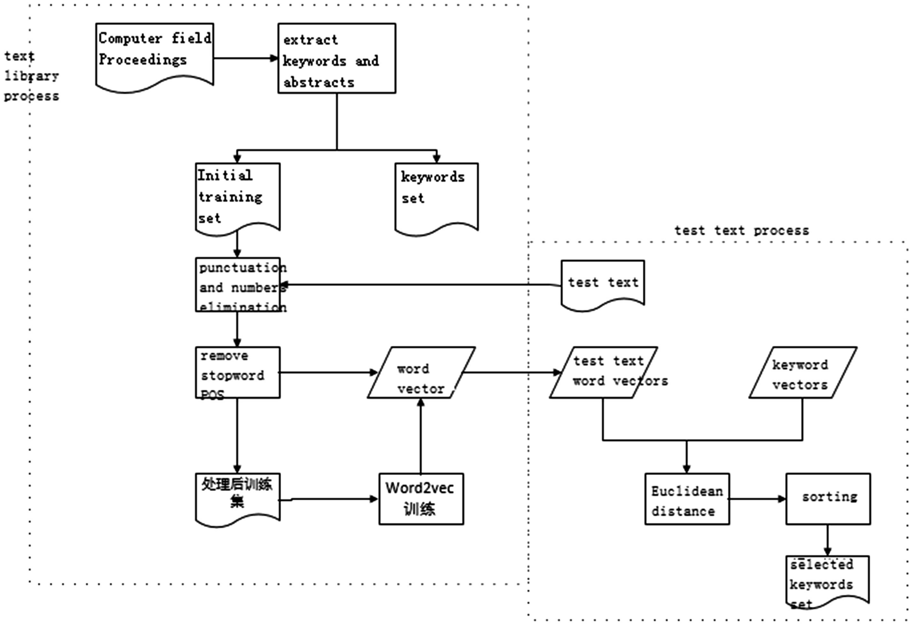

Fig. 1. Overall process

3.1 Text Library Process

3.1.1 Preprocessing of Dataset

The dataset used in the experiment comes from 8064 computer-related English paper records, where each one has five fields including title, paper source, authors, keywords and abstract.

Firstly, we should establish the keyword set: check keywords field, extract all the keywords, then filter out the keywords which appear more than five times in all records (namely, the word is selected by more than five papers as keywords) as keyword set. Meanwhile, in the word2vec training process, each word vector is automatically segmented by a blank. Therefore, regard keyword phrases as a whole, represent the blanks between each word in the phrase with "-" without thinking about the single-word keyword. Then, make the statistics of keyword set, choose the total 4429 keywords, the ratio of keyword phrase and single keyword is 0.679, keyword phrase accounting for about two-thirds. This proportion will be used as the standard to guide the whole process in the latter procedure of testing keyword extraction, it denotes by $\partial = 0.679$.

Secondly, establish the initial training set: for each record, check its keyword field. If the number of keywords which exist in the previously established keyword set is greater than 2, extract the abstract and keyword field of this record. Then, we select out 749 records. Next, we use three cycles: cycle one shows each word in abstract, cycle two represents the appeared keyword phrase, Cycle three means that, for each phrase which appears in the abstract, replace the blank to "-", or for the separate words in the phrase which appears in the abstract, it is automatically extended to the corresponding phrase, remove the middle blank and use "-" instead, Finally, after processing the abstract, put all the corresponding keywords at the last of abstract, add the initial training set, until all the records processing finished.

After the initial training set is set up, some preprocessing are needed including the filter of punctuation, numbers, stop words and part of speech. It is taken into account that the keywords are usually contain noun phrase or single word, rarely containing punctuation or stop words. Stop words include function words (such virtual words as "and", "the", and "of", modal particles, conjunctions, adverbs) or other words with minimal lexical meaning. In most cases, because these words appear so frequently and is widely used in the user analysis and searching mission that they are considered uninformative or meaningless in general. Thus, they are excluded by most of the information extraction and text analysis system.

This paper use the stop list which contains 891 English words. At first, remove the punctuation, numbers, and stop words in the initial training set. Then, the nltk Toolkit is used to filter the part of speech, remaining all the noun terms and phrases and eliminating most of the adjectives, verbs, adverbs and other parts of speech. Nltk is a python toolkit for managing the natural language processing related works including tokenization, part-of-speech (POS), text classification and so on. Nltk of POS tagging tool is used in this paper, tagging words with part of speech and filtering out the terms and phrases.

3.1.2 Generation of Word Vectors

After preprocessing, the post-processing training set trains the documents with the help of word2vec tool. This paper selects the CBOW model and uses the hierarchical training policy, in which the text window is set to 5. Through training, the text is convert into a word vector set of 100-dimensional vector space, and is saved in the file of vectors.bin. The file includes 8,322 words (including the connection phrase), followed by a 100-dimension float vector, which is used as an input to the next process of text testing.

3.2 Test Text Process

For the text to be tested, the first thing to do is preprocessing, whose method is the same as training set, arranging text to a collection of words. According to vectors.bin file and the keywords set, we get the under test word vectors and keywords vectors, calculate the Euclidean distance between each keyword and the word to be tested. Then we sort the keyword, according to ∂ value obtained before and the number of word in the processed text, select the top $\mathrm{T} * \partial$ keyword phrases and top $\mathrm{T} * (1 - \partial)$. Single keywords as the eventual keywords, according to [16], We select as 1/4 of the totally number of the text.

Since the experiment involves complex calculation of large sample float vectors arithmetic Euclidean distance, stand-alone operation execution speed is so slow that it impacts the experimental efficiency. Therefore this paper is based on the hadoop distributed systems infrastructure, using the distributed file system HDFS and MapReduce programming model, applying 7 computing nodes, which greatly enhances the efficiency of the experiments.

3.3 Analysis of Experimental Result

In this paper, the experimental framework is based on Hadoop. Data input includes four parts including word vector file which obtained from the preprocessing(named “vectors.bin”), keywords set(named “keywords.txt”), testing text(named “testfile.txt”), manual selection keywords of text testing(named “output.txt”). Data Outputs are keywords and their distance which are automatically extracted, in which the distance is selected manually.

This paper uses three evaluation criteria which are commonly used in the field of information retrieval. They are Precision(denoted by P), Recall(denoted by R) and F-measure to analyze the experimental result. Three standard formula are as formula 1, 2, 3 follows:

$$P = \frac{The\ number\ of\ correctly\ extracted\ keywords}{the\ number\ of\ extracted\ keywords} \tag{1}$$

$$R = \frac{The\ number\ of\ correctly\ extracted\ keywords}{the\ number\ of\ manual\ selection\ keywords} \tag{2}$$

$$F - measure = \frac{2 * PR}{P + R} \tag{3}$$

Table 1 lists comparison between the keywords we extracted and the manual assigned keywords in one of the paper record. After preprocessing, it has 24 words, so T = 6, we select the minimum distance of 4 keyword phrases and two single keywords.

From the table above we know that for the phrase keyword, in the automatically extracted four keyword phrases, the correct extracted number is three, P is 75 % while the manual selection number of keyword phrases is five, R equals to 60 %; Based on P and R, the obtained F-measure is 67 %. For the single keywords, in the two automatically extracted words, the correct number is one, so P is 50 % while the manual selection of single keywords is one, the recall rate R is 100 %; Based on P and R, the obtained F-measure also reaches 67 %. For the overall keywords, P, R and F-measure are all 67 %.

4 Experiment

In order to test the performance of the extraction method based on the word2vec, we compare this method with TextRank [1] and RAKE [2]. We randomly take 10 % of the previous 749 papers, 75 papers exactly, as the test set. And then we verify the results of those three methods using P, R and F-measure indexes.

Mihalcea and Tarau (2004) describe a system which applies a series of syntactic filters to identify POS tags for selecting keywords [1]. It is based on PageRank algorithm, and the basic idea of PageRank is: the importance of a web page depends on the quantity of the backlinks and the importance of these pages. PageRank algorithm regards the whole World Wide Web pages as a directed graph, and the node of the graph is a web page. If there is a link from A to B, then there is a directed edge from the A to the B in the directed graph. Hence, the TestRank splits the text into sentences, and the sentences are split into words. Then it sets up the window size, Co-occurrences of the selected words within a fixed-size sliding window are accumulated within a word co-occurrence graph. A graph-based ranking algorithm (TextRank) is applied to rank words based on their associations in the graph, and then top ranking words are taken as the keywords. Keywords which are adjacent in the document are combined to form multi-word keywords.

RAKE algorithm is raised in the paper [2] proposed by Rose S and Engel D in 2010, and it is an unsupervised, domain-independent and language-independent method for extracting keywords from individual documents. The basic idea of this algorithm is: RAKE is based on our observation that keywords frequently contain multiple words but rarely contain standard punctuation or stop word. The input parameters for RAKE comprise a list of stop words (or stop list), a set of phrase delimiters, and a set of word delimiters. The score which is based on the degree and frequency of word vertices in the graph is calculated for each candidate keyword and defined as the sum of its member word scores. Finally the top T scoring candidates are selected as keywords for the document, and it computes T as one-third the numbers of words in the graph.

Table 1. Result of our method

Extracted by word2vec	Manually assigned
Decision-theoretic	Information-retrieval-system
Decision-making	Evaluation
Information-retrieval-system	Information-search-process
Analytic-hierarchy-process	Decision-making
Evaluation	Multi-criteria-model
Retrieval	Analytic-hierarchy-process

Table 2 shows the comparison result using word2vec method, TestRank method and the RAKE method to extract keywords automatically from 75 abstracts.

Table 2. Comparison result

Method keyword pattern	Precision P	Recall R	F-measure
word2vec method keyword phrases	**66 %**	**70 %**	**68 %**
word2vec method single keywords	19 %	31 %	24 %
word2vec method keywords	**51 %**	**61 %**	**51 %**
TestRank keyword phrases	23 %	20 %	21 %
TestRank single keywords	27 %	30 %	28 %
TestRank keywords	24 %	25 %	24 %
RAKE keyword phrases	42 %	18 %	25 %
RAKE single keywords	**41 %**	**38 %**	**39 %**
RAKE keywords	42 %	22 %	29 %

In the table above, the top three evaluation values of keyword phrases, single keyword, and keywords are respectively marked in bold. Obviously, our method can obtain a great enhanced performance in the keyword phrase extraction, and it can get the keywords which don't exist in the abstract but may become a candidate keywords from keywords set via to train a large number of samples. Although our method does not have a very satisfactory performance for the single keywords extraction, due to the practical application, keyword phrases appear to be higher frequency. So the overall performance of the keyword extraction is better than other two ways with a more significant improvement.

5 Conclusion

This paper proposes an automatic keyword extraction method which based on word2vec tools. We obtain the word vectors through training the large-scale samples, the text conceptual space is then convert into computable space. It means that the text information knowledge equals to computer-readable knowledge and facilitate people to perform data extraction or keyword match. By calculating the Euclidean distance between text words and keywords, using the prior knowledge of proportion of single keywords and keyword phrases, we can extract the shortest distance ones respectively. This experiments using the computer field proceedings as the training set and extract the corresponding domain keywords set. The result shows that this method improves the performance of the keyword phrase extraction and find the keywords which are not include in the text at the same time, so that the overall performance of automatic keywords extraction is greatly improved compared to the previous method.

Indeed, our method has some flaws. On one hand, the extraction performance of single keyword doesn't increase significantly compared to the previous methods. However, according to prior obtained domain keywords set, the proportion of single keywords in the overall keywords is not high, the overall performance of the automatic keyword extraction is still enhanced in a large rate because most of the keywords are existed in the form of phrase. On the other hand, the training number of samples is too large, it leads to an increase of the execution time and a decrease of the efficiency. However, the original intention of our method is applied in the large-scale sample problems. For the word2vec tool, the larger the sample size is, the higher accuracy of the semantic meaning of the word vectors would be. And it takes advantages of keyword automatic extraction. Meanwhile most of the previous work of automatic keyword extraction algorithm can only extract the keywords which exist in the text, but our method can extract the rest of other keywords from domain keywords set, those keywords can also summarize the text topic. Hadoop cluster architecture used in this paper can also speed up the overall operating efficiency to increase the integrated performance.

Based on word2vec, we have shown that our automatic keyword extraction technology achieves higher precision in comparison to the existing techniques. However, this method still has some improvements and enhancements in the future work. Firstly, the strong domain dependence. In general, we establish the keywords set in the certain domain according to the training samples, then the test text must be the same domain as the keywords set. So we intend to set up the adaptive model and adjust parameters to select keyword sets in different areas; secondly, the performance of single keywords extraction is relatively low. We can combine our method with the mature automatic keyword extraction algorithm like TF-IDF, etc., to focus on improving the accuracy of single keyword extraction in future work; finally, in order to calculate the distance between two word vectors, in addition to the Euclidean distance, other methods like cosine can also be performed. We will use different methods to calculate vectors distance and compare them to determine the best performance calculation method of keyword extraction.

Acknowledgements. The work of this paper is partially supported by National Natural Science Foundation of China (No. 61303097) and Ph.D. Programs Foundation of Ministry of Education of China (No. 20123108120026).

References

1. Mihalcea, T.P.: TextRank: bringing order into texts. Association for Computational Linguistics (2004)
2. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. In: Berry, M.W., Kogan, J. (eds.) Text Mining: Theory and Applications. Wiley, Hoboken (2010)
3. Xiaolin, W., Lin, Y., Dong, W., Lihua, Z.: Improved TF-IDF keyword extraction algorithm. Comput. Sci. Appl. **3**, 64–68 (2013)
4. Witten, I.H., Paynter, G.W., Frank, E., et al.: KEA: practical automatic keyphrase extraction. In: Proceedings of the 4th ACM Conference on Digital Libraries, Berkeley, California, US, pp. 254–256. ACM (1999)
5. SuJian, L., HouFeng, W., ShiWen, Y., ChengSheng, X.: Research on maximum entropy model for keyword indexing. Chin. J. Comput. **27**(9), 1192–1197 (2004)
6. Gonenc, E., Ilyas, C.: Using lexical chains for keyword extraction. Inf. Process. Manage. **43** (6), 1705–1714 (2007)
7. Yih, W., Goodman, J., Carbalho, V.: Finding advertising keywords on web pages. In: International World Wide Web Conference Committee (IW3C2), May 23-26 (2006)
8. Zhunchen, L., Ting, W.: Research on the chinese keyword extraction algorithm based on separate models. J. Chin. Inf. Process. **23**(1), 63–70 (2009)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representation in vector space. Cornell University Library, 7 September 2013 (2013)
10. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics. Human Language Technologies (2013)
11. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. Citeseer, October 2001
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. Cornell Unicersity Library, 16 October 2013 (2013)
13. Morin, F., Bengio, Y.: Hierarchical probabilistic neural network language model. In: AISTATS (2005)
14. Hinton, G.E.: Learning distributed representations of concepts. In: Proceeding of the Eighth Annual Conference of the Cognitive Science Society (1986)
15. Mnih, A., Hinton, G.: There new graphical models for statistical language modeling. In: Proceedings of the 24th International Conference on Machine learning, pp. 641–648 (2007)
16. Hulth, A.: Combining machine learning and natural language processing for automatic keyword extraction. Stockholm University, Faculty of Social Science, Department of Computer and System Science (together with KTH) (2015)
17. Hiton, G.E., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
18. Luhn, H.P.: A statistical approach to the mechanized encoding and searching of literary information. IBM J. Res. Dev. **1**(4), 309–317 (1957)
19. Wang, L.: The research of keywords extraction algorithm in text mining. College of Computer Science and Technology, Zhejiang University of Technology, Zhejiang (2013)