Chinese Text Summarization Algorithm Based on Word2vec

To cite this article: Xu Chengzhang and Liu Dan 2018 *J. Phys.: Conf. Ser.* **976** 012006

Chinese Text Summarization Algorithm Based on Word2vec

Xu Chengzhang and Liu Dan

Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China

15208171819@163.com

Abstract. In order to extract some sentences that can cover the topic of a Chinese article, a Chinese text summarization algorithm based on Word2vec is used in this paper. Words in an article are represented as vectors trained by Word2vec, the weight of each word, the sentence vector and the weight of each sentence are calculated by combining word-sentence relationship with graph-based ranking model. Finally the summary is generated on the basis of the final sentence vector and the final weight of the sentence. The experimental results on real datasets show that the proposed algorithm has a better summarization quality compared with TF-IDF and TextRank.

1. Introduction

With the advent of the era of big data, the volume of data on the Internet is growing rapidly. This makes artificial text summarization methods unable to meet the needs of users. Text summarization technology is one of the effective tools to solve this problem. Text summarization technology is a method that can extract or generate the summarizations under the help of computers [1].

There are two main methods to get the summary, the extracted summarization method and the generative summarization method.

The extracted summarization method assumes that the topic can be summed up into some sentences or a few words in the text, so the text summarization problem can be transformed into a sentence representation and sorting problem. In the literature [2], author proposed a statistical summarization method. TF-IDF algorithm was used to calculate the weight of words and sentences, and the summary was extracted from the text after sorting the sentences by weight. In the literature [3], author proposed a summarization method based on similarity. From the multi-document model, a probability distribution of words that indicates the relationship between words and topics was generated. The weight of sentence was calculated based on the similarity between words. Thus, the summary was extracted from the text after sorting the sentences. Moreover, extracted summarization methods based on machine learning [4] and graph model [5] were also effective in representing and sorting the sentences. In most of the above methods, the frequency statistics for each word were used to calculate the weight of the sentence and the sentences were sorted on the basis of the weight, the graph model and the text feature.

The generative summarization method is a method to generate the natural language that can cover the main topic of the article after the computer understands the content. In the literature [6], author proposed a data-driven approach. This approach utilized an attention-based model to generate each word of the summary. The model could easily be trained end-to-end and scaled to a large amount of training data. However, this method was limited by natural language processing technology and did not work well in long articles.

The text summarization algorithm used in this paper belongs to the extracted summarization method.

2. Text summarization algorithm based on Word2vec

FIG. 1 is the flowchart of the summarization algorithm based on Word2vec. The algorithm firstly divides the text into sentences and divides each sentence into words, and each word is represented as a vector trained by Word2vec. Then the algorithm calculates the weight of each word, the weight of each sentence and the sentence vector iteratively on the basis of the word vector, word-sentence relationship and graph-based ranking model. Finally, a summary is generated on the basis of the final sentence vector and final weight of the sentence.

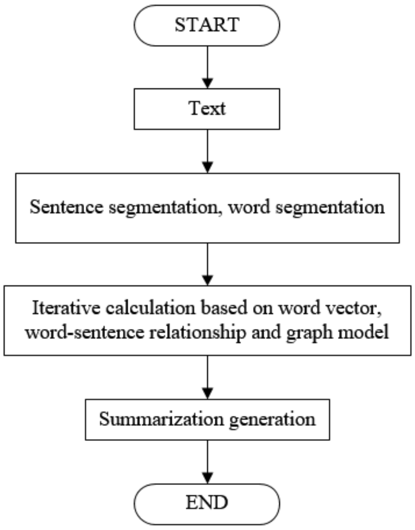
START
Text
Sentence segmentation, word segmentation
Iterative calculation based on word vector, word-sentence relationship and graph model
Summarization generation
END

Figure 1. Flowchart of the summarization algorithm

2.1. Word2vec

Word2vec [7] is an open source toolkit to produce word vectors. Vectors trained by Word2vec have the following characteristics: the correlation between words can be measured by the vector distance between word vectors. The higher the semantic relevance is, the shorter the distance between two word vectors will be. Word2vec is now widely used in natural language processing tasks such as clustering, synonyms, Sentiment Analysis and text classification.

Word2vec can utilize Continuous Bag-of-Words Model (CBOW) and Skip-gram Model to produce word vectors. The main idea of CBOW is to predict the probability of the centre word on the basis of the context around the word, and Skip-gram Model is to "skip some symbols" around the centre word to predict the context [8]. While CBOW Model is time-consuming, and the training effect is limited by the size of the sliding window, Skip-gram Model has higher semantic accuracy, but the computational complexity is high too and takes a long time for the training process.

In Chinese text summarization, there are some algorithms which can consider the meaning of words and similarity between words [9] [10] [11], and most of these methods sort the sentences by text feature so that they are confined to certain texts and have no universal applicability. On the basis of Word2vec, the text summarization algorithm can to some extent preserve the semantic relevance between sentences.

2.2. Iterative calculation process based on word-sentence relationship and graph model

The basic idea of iterative calculation process based on word-sentence relationship and graph model is that on the one hand the more key words a sentence consists of, the higher weight this sentence will have. On the other hand, the more frequently a word appears in a high weight sentence, the higher weight this word will have.

The algorithm first defines the variables. Assuming that the set of sentences is *S*, the number of sentences is m, and the weights of the sentences are expressed as a m-dimensional vector $T = [T_1, T_2, ..., T_m]$, in which T_i represents the weights of the i-th sentence; the set of words is *W*, the number of words is *n*, and the weights of words are expressed as a n-dimensional vector $Y = [Y_1, Y_2, ..., Y_n]$, Y_i is the weight of the i-th word, furthermore, the initial weight of each word is calculated by TF-IDF [12] or assigned equally; Word2vec vector corresponding to *W* is $V = [V_1, V_2, ..., V_n]$; p_{ij} represents the number of occurrences of the word j in sentence i.

D represents words in a sentence, and the sentence is represented as a sentence vector *x*:

$$x = \sum\nolimits_{i \in D} V_i \times Y_i \qquad (1)$$

The similarity between sentences is calculated by using the cosine value of the angle between them.

$$sim(X, Y) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|} \qquad (2)$$

An undirected graph G is established in which the sentence is treated as node and the similarity between sentences is treated as edge. After several iterative computations combined with word-sentence relationship and PageRank [13] on G, the final weight of each word, the final weight of each sentence and the final sentence vector can be calculated. Experiments in the literature [5] proved that after several iterations of the weights, the weight of sentence converged. The terminating condition is a threshold. When the threshold is 0.0001, after 20-30 iterations, the weight value converges [14]. To get the summary, sentences which have a higher similarity to the sentence with the highest weight than a threshold are regarded as repetitions and omitted. Ultimately, summaries can be generated in the top K sentences, and the K can be determined according to some conditions like the number of summaries and the number of words in summary /the number of text words.

2.2.1. *Iterative computation* .

The iterative computation is a combination of word-sentence weight-propagation and PageRank. Based on the word frequency in a sentence, there is a weight-propagation between words and sentences. Word2vec is used to further update the vector representation of the sentence.

The iterative computation firstly runs a sentence weight calculation algorithm like PageRank on G:

$$T_i = \frac{1-d}{m} + d \times \sum\nolimits_{d \in In(d_i)} T_j \times \frac{sim(d_i, d_j)}{\sum\nolimits_{d_k \in Out(d_j)} sim(d_j, d_k)} \qquad (3)$$

In where, d is the damping coefficient (default is 0.85), m can be 1 or the number of the sentences in the article. The impact of other sentences on the current sentence is controlled by d and m. d_i is a sentence node in graph G, which represents the i-th sentence in S. *In*(d_i) represents the set of nodes pointing to the sentence node d_i and *Out*(d_j) represents the set of nodes pointed to by d_j . T_j is the weight of the sentence node d_j in this round. In (3), the weight of the current sentence is calculated on the basis of the similarities to other sentences and the weight of other sentences. Perform such a calculation for each sentence until the weight converges and then we can get the initial weight of the sentence.

Based on the weights of sentences in (3), the weights of words can be updated:

$$Y_j = (\sum_{i=1}^{m} p_{ij} \times T_{\mathrm{i}}) / (\sum_{i=1}^{m} p_{ij}) \qquad (4)$$

After the weight of each word is obtained, the weight of each sentence can be recalculated and the sentence vector can be recalculated by (1):

$$T_i^{'} = (\sum_{j=1}^{n} p_{ij} \times Y_j) / (\sum_{j=1}^{n} p_{ij}) \tag{5}$$

Finally, after a combination of formula (3), (4) and (5), the weight of words and sentences, and sentence vector can be the initial value in the next iteration, α is the adjustment factor, the sentence update method is:

$$T_i = a \times T_i + (1 - a) \times T_i^{'} \tag{6}$$

After several iterations, algorithm will get the final weight of sentence and the final sentence vector.

2.2.2. Summarization filter. Based on the final weight of each sentence and the final sentence vector, sentences which have a higher similarity to the sentence with the highest weight than a threshold are regarded as repetitions and omitted. To generate the summary, number of words in summary / number of text words is considered as the precondition for K, and the summary is the remaining top-k sentences.

3. Experiment

3.1. Experimental corpus

There is lack of authoritative test data in the field of Chinese text summarization. By using web crawler, some news from Ifeng that belongs to different subject like finance, entertainment, culture and sports were fetched. On the fetched data a Word2vec model was trained too. The experiment was conducted based on 100 news data from all areas mentioned above. The statistical characteristics of our experimental data are shown in Table 1. In our experimental data, there were 2722 sentences and 149,668 words in total. The minimum number of words in an article was 365, the maximum number of words was 6425; the minimum number of sentences was 4, and the maximum number of sentences was 149. To generate the summary, number of words in summary / number of text words was considered as the precondition for K. The weight of the sentence was sorted in advance artificially, and the top-K sentences were selected as the expert summaries. Edmundson[15] was used to evaluate the results. Edmundson was defined as:

$$\text{Coincidence degree} = \frac{\text{Number of matched sentences}}{\text{Number of expert summaries}} \times 100\% \tag{7}$$

Table 1. Statistical characteristics of news corpus

Number of news	100
Total number of sentences	2722
The maximum number of sentences in an article	149
The minimum number of sentences in an article	4
Total number of words	149668
The maximum number of words in an article	6425
The minimum number of words in an article	365

3.2. Analysis of experimental result

In the experiment, the precondition for K was that number of words in summary / number of text words was close but no more than 10%. The experimental result is shown in Table 2. The coincidence degree in financial news was 34.06%, in entertainment news was 31.23%, in sports news was 38.29%, and in culture news was 27.44%. Respectively, the average was 31.54%. The result shows that the effect of Chinese text summarization algorithm based on Word2vec is better in financial, entertainment, sports news, and is average in cultural news. In the financial, entertainment and sports test data, the number of words contained in a sentence is smaller, so the Word2vec-based feature representation method can better retain the relevance between the sentences. In addition, compared with entertainment and culture news, there are more repetitive keywords and stronger semantic relevance in financial and sports news, which make the weights of sentences more accurate.

Table 2. Experimental results

Category	**Average of K**	**Coincidence degree /%**
Finance	4.3	34.06
Entertainment	1.82	31.23
Sports	2.46	38.29
Culture	6.75	27.44
Average	3.14	31.54

3.3. Comparison with the original method

On the same experimental corpus, the proposed method was compared with TF-IDF and TextRank, and the evaluation result is shown in Figure 2. Compared with TF-IDF and TextRank, the method proposed in this paper had the highest coincidence degree in financial, entertainment and sports news, while in cultural news the coincidence degree was only almost 1.6% lower than that of TextRank. On average, the coincidence degree of our method was 6-7% higher than TF-IDF and TextRank. The result shows that method proposed in this paper has a better summarization quality in news from most subjects compared with TF-IDF and TextRank.

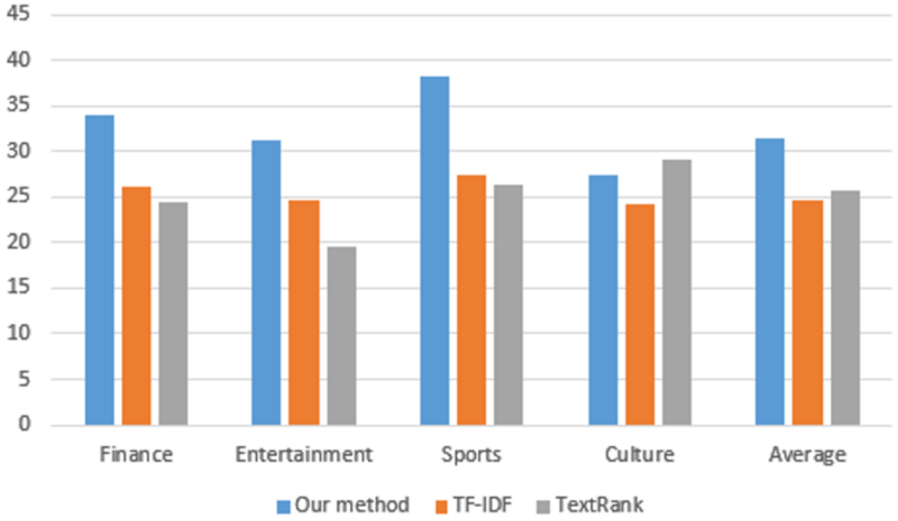

Figure 2. Result of comparative Experiment

4. Conclusion

Focusing on the issue that Chinese text summarization needs to extract some sentences to cover the topic of an article, we proposed a Chinese text summarization algorithm based on Word2vec. After converting words into vectors, the algorithm updates the weight of words and sentences iteratively on the basis of the combination of word-sentence relationship and the graph ranking model. Experimental results on real datasets shows that the text summary algorithm has a better summarization quality compared with TF-IDF and TextRank.

References

[1] Tan Chong, Chen YueXin. Literature Review of Automatic Summarization Method[J]. Journal of The China Society for Scientific and Technical Information, 2008, 27(1):12-12.
[2] Cheng QianQian, Tian DaGang. Automatic Chinese Summarization Model Based on Basic Elements Method[J]. New Technology of Library and Information Service, 2010, 26(2):74-78.
[3] Wang Hongling, Zhang MingHui, Zhou GuoDong. Chinese multi-document summarization system based on topic information[J]. Computer Engineering and Application, 2012, 48(25):132-136.
[4] Cao Yang, Cheng Ying, Pei Lei.A Review on Machine Learning Oriented Automatic Summarization[J]. LIBRARY AND INFORMATION SERVICE, 2014, 58(18):122-130.
[5] Mihalcea R, Tarau P. TextRank: Bringing Order into Texts[J]. Unt Scholarly Works, 2004:404-411.
[6] Rush A M, Chopra S, Weston J. A Neural Attention Model for Abstractive Sentence Summarization[J]. Computer Science, 2015.
[7] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.
[8] Xiong FuLin, Deng YiHao, Tang XiaoCheng. The Architecture of Word2vec and Its Applications[J]. Journal of Nanjing Normal University, 2015(1):43-48.
[9] Huang ChengHui, Ying Jian, Hou Fang. A Text Similarity Measurement Combining Word Semantic Information with TF-IDF Method[J]. Chinese Journal of Computers, 2011, 34(5):856-864.
[10] Cheng Yuan, Wushouer SILAMU, Maimaitiyiming HASIMUA. Automatic Text Summarization Based on Comprehensive Charateristics of Sentence[J]. Computer Science, 2015, 42(4):226-229.
[11] Jiang XiaoYu. Automatic Summarization Algorithm Based On Keyword Extraction[J]. Computer Engineering, 2012, 38(03):183-186.
[12] Tang Ming,Zhu Lei, Zou XianChun. Document Vector Representation Based on Word2Vec[J]. Computer Science, 2016, 43(6):214-217.
[13] Page L. The PageRank citation ranking : Bringing order to the web[J]. Stanford Digital Libraries Working Paper, 1998, 9(1):1-14.
[14] Luhn H P. The automatic creation of literature abstracts[M]. IBM Corp. 1958.
[15] Edmundson H P. New Methods in Automatic Extracting.[J]. Journal of the Acm, 1969, 16(2):264-285.