The enhancement of TextRank algorithm by using word2vec and its application on topic extraction

Xiaolei Zuo[1], Silan Zhang[1, 2] and Jingbo Xia[2, 3,4*]

[1]College of Science, Huazhong Agricultural University, Wuhan, China
[2]Institute of Applied Mathematics, Huazhong Agricultural University, Wuhan, China
[3]College of Informatics, Huazhong Agricultural University, Wuhan, China
[4]Hubei Key Laboratory of Agricultural Bioinformatics, Wuhan, China
*Corresponding author, mailto: xiajingbo.math@gmail.com

Abstract. TextRank is a traditional method for keyword matching and topic extraction, while its drawback stems from the ignoring of the semantic similarity among texts. By using word embedding technique, Word2Vec was incorporated into traditional TextRank and four simulation tests were carried on for model comparison. The results showed that the hybrid combination of Word2Vec and TextRank algorithms achieved better keyword/topic extraction towards our testing text dataset.

1. Introduction

With the booming development of new media and the Internet, the text data of unstructured or semi-structured news is proliferating. Extracting effective information from the complex and irregular texts, which can certainly improve daily reading efficiency, is of great significance.

PageRank is a sorting algorithm for webpages [1]. The algorithm assigns a heavier weight to webpage which is more frequently cited by other webpages. In another word, the importance metric of a webpage relies on the amount of the linking resources. Assume $V=\{v_1,v_2,\cdots v_n\}$ is the set of webpages, m_j is the weight for each webpage, $In(V_i)$ refers to set of webpages that link to V_i, and the importance metric of webpage V_i is regarded as $p(v_i)$:

$$p\left(v_i\right) = \frac{1-d}{n} + d \sum_{v_j \in In\left(V_i\right)} m_j \times p\left(v_j\right) \tag{1}$$

where d is a factor, which normally is set as 0.85. A drawback of this algorithm is that an outlier will draw significant effect to the result, e.g., a dramatic change of one webpage in $In(V_i)$ will bring dramatic value change. Henceforth, several enhancement were made to enhance PageRank [2, 3].

In light of PageRank, Tarau and Mihalcea proposed TextRank in 2004 [4]. In TextRank, article is divided into basic text units, i.e., words or phrases. As treated as webpage in PageRank, text unit maps to vertex in graph, and edge between vertexes refers to the link between text units.

The research of this paper is to introducing semantic similarity into text unit of sentences so as to achieve better topic representation of a text. Word2Vec [5] are used for the purpose of semantic embedding. Four simulation tests are carried on after the algorithm design and coding. The data and codes are available in GitHub, https://github.com/zuoxiaolei/TextRankPlus.

2. Material and Method

2.1. Material

Literature abstracts are extracted from www.sciencedirect.com. Four type of topics are chosen randomly for keyword evaluation. As shown in Table 1.

Table 1. Literature sampling for evaluating the keywordk extraction

Topic for literature	Filed	Amount of the literature	Keywords from expertise's view
Biomedical natural language processing	Computer science [6]	5	NLP, text mining
Intrusion detection system	Computer science [7]	5	IDS, classifier, evaluation metric
Bioinformatics method for predicting thermophilic proteins.	Bioinformatics [8]	5	Prediction, Pseudo AAC, thermophilic
Thue equation	Mathematics [9]	5	Thue equation

The reason of the topics selection comes from authors' expertise in these wide-range disciplinary fields, based on which we released sufficient amount publications, thus made it accurate to define the relevant keywords/topics for each texts. From the view of our expertise, the keywords for each topic are safely pre-set, as shown in the last column in Table 1, and all of the topics will be tested in our simulation test in the Result section. The text data for evaluation is with proper length that make it sufficient to test the accuracy of keyword/topic extraction. The data is also delivered in Github for free downloading (https://github.com/JingboXia/Enhancement_of_TextRank).

2.2. Method

2.2.1. TextRank algorithm for keywords ranking. TextRank [4] is built in a graph-based unsupervised learning frame, and it has been widely used in keywords extraction and automatic abstracting. The core of TextRank come from vertex voting, where the voting action equals to an edge between two vertexes. The keyword is mapped with higher value, if the vertex it represents has higher relevance with the rest vertexes. In our research, the idea of TextRank is used for keywords ranking among four types of scientific abstracts.

PageRank algorithm for page ranking.

In the beginning, each vertex is assigned with equal weight, and afterwards a recurrent calculation update the weight thought voting. Here, $G=(V,E)$ is the graph, with V being the vertex set, E being the edge set. The importance metric of each vertex is as shown in the formula:

$$W(V_i)=(1-d)+d*\sum_{j\in In(V_i)}\frac{w_{ji}}{\sum_{V_k\in out(V_j)}w_{jk}}W(V_j) \tag{2}$$

where $In(V_i)$ is the set of subscripts for vertexes (text units) which share a common window with V_i in linear order in sentence, $Out(V_j)$ is the set of vertexes which share a common window with V_i, d is a damping factor, and its default value is 0.85. Normally, weight assigned to V_j to V_i, w_{ji}, is counted by calculating the chances of two text units co-occurred in a text window with fixed size, and the usual size equals to 2.

In the initialization step, weight of each text unit is one, and all of the weights reach consistency after recurrent calculation by formula (2). The text units in the top ranking list are considered to be keywords of the text.

The flowchart of the algorithm is shown in Figure 1.

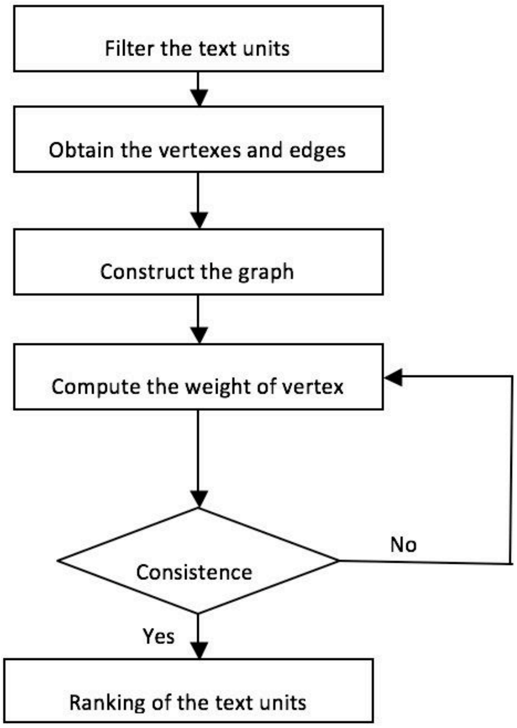
Filter the text units
Obtain the vertexes and edges
Construct the graph
Compute the weight of vertex
Consistence
No
Yes
Ranking of the text units

Figure 1. Classic TextRank algorithm workflow

The advantage of TextRank is that it is an unsupervised learning algorithm in no need of huge corpus for training. It make it easy to be adopted for handling other text resources in an efficient way.

2.2.2. Word embedding (CBOW and Skip-gram). The disadvantage of TextRank is that it omit the keywords which has lower chance to appear though being meaningful in context. The natural way to enhance TextRank is to use the semantic similarity of words and avoid the miss selection of vital keywords.

Word2vec is a word embedding algorithm proposed by Google in 2013, which have two varities: CBOW and Skip-Gram. The main idea of Word2vec is to find numerical vector representation of word by using neural networks.

The idea of Word2vec comes from the probability calculation of Bayesian occurrence estimation, Let $T = w_1, w_2, \cdots w_n$ be sentences including n words, the probability of occurrence of the sentence T is:

$$P(T) = \prod_{i=1}^{n} p(w_i \mid w_{i-n-1} w_{i-n-2} \cdots w_{i-1}). \tag{3}$$

Similarly, the Bayesian estimation of the occurrence chance of the i-th word is:

$$p(w_i \mid w_{i-n-1} w_{i-n-2} \cdots w_{i-1}) = \frac{C(w_1 w_2 \cdots w_n)}{C(w_1 w_2 \cdots w_{n-1})}$$

where $C(w_1 w_2 \cdots w_n)$ is the probability of the sentence $w_1 w_2 \cdots w_n$ in the corpus.

In CBOW, context information of each word $w \in W$ is concerned within a ws - width window. The purpose of CBOW training is to maximize the probability of $(w, C_w) \in T$ and minimize the probability of $(w, C_w) \notin T$. Here, the probability of the occurrence of *w* based on C_w is:

$$p(w \mid c_w) = \frac{1}{1 + e^{-v_{c_w} W_w}}$$

where W_w is the weights matrix connecting the hidden layer and softmax layer, v_{c_w} refers to the sum of the numerical vectors which are in the flank side of the target word:

$$v_{c_w} = \sum_{w \in c_w} v_w$$

The likelihood function of the model is

$$OBJ_{cbow} = arg \max_{v_w, W_1} \left(\prod_{(w, c_w) \in T} log(p(w \mid c_w)) \cdot \prod_{(w, c_w) \notin T} (1 - log(p(w \mid c_w))) \right) \tag{4}$$

Negative sampling strategy is used to obtain $(w, c_w) \notin T$, that make a quicker implementation. And gradient descent algorithm is used for parameter optimization as well.

Similar as CBOW, Skip-gram is to predict the neighbor words, and its objective function is to calculate the greatest average logarithm:

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} log p(w_{t+j} \mid w_t) \tag{5}$$

where *c* refers to the window width. As shown in Figure 2, CBOW and Skip-Gram own similar ideas and both were considered in our research.

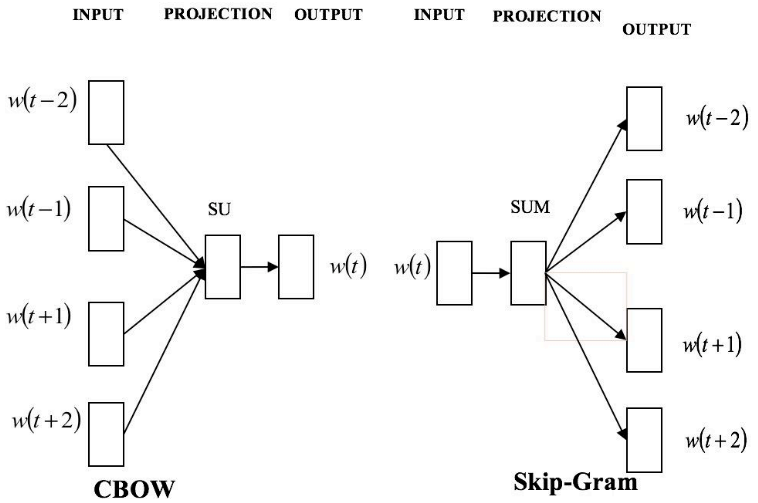

Figure 2. CBOW and Skip-Gram model structures

2.2.3. Enhanced TextRank algorithm. The classic TextRank algorithm only considered the occurrence info of text units in sentences, but omit the semantic understanding of them. Furthermore, the same initial weights is assigned for each text unit, without differentiating the semantic representation of text unit. Henceforth, the word appeared more often is readily be selected to be keyword, regardless its importance. In order to solve this drawback, Word2Vec and TextRank are combined so as to form our proposed algorithm. In this way, lower-rank numerical vector is assigned for each word, and semantic similarity between text units are remained.

For implementation, popular python package Gensim is used for Word2Vec training and model construction (https://radimrehurek.com/gensim/), and Wikipedia Corpus is selected as training corpus. Here, Wikipedia corpus is a domain-free corpus which ensures a model with better generalization ability. Built-in Word2Vec computation in Gensim treate each text unit as a vertex of graph, and the similarities distribution among text units are calculated as edge between vertexs. Finally, in the integration part of TextRank and Word2Vec, the egde value between words are set as semantic similarity. In detail, to replace formula (2) in TextRank, weight value w_{ji} is reset by using semantic similarities counted by Word2Vec distance.

3. Result and Conclusion

3.1. Results of Comparison

The result of comparison among four selected topics are shown in Table 2.

Table 2. Comparison of performance for testing dataset with selected topics

Topics for testing text	TextRank	Our algorithm
Topic 1: Biomedical natural language processing	1. system-oriented evaluation 2. structured information 3. extraction performance 4. underlying biomedical 5. hypothesis generation 6. classification 7. 5-10 8. Enormous	1. biomedical text mining (√) 2. next 5-10 3. undertake user-oriented 4. great promise 5. PubMed search (√) 6. published biomedical 7. past year 8. text mining (√)
Topic 2: Intrusion detection system	1. detection evaluation 2. taxonomy 3. accurate 4. particular 5. knowledge encoding 6. statistical 7. general-purpose intrusion-detection (√) 8. extensive	1. DARPA Intrusion Detection Evaluation (√) 2. Mining Audit Data 3. real-time intrusion-detection expert (√) 4. general-purpose intrusion-detection expert (√) 5. Current IDSs pose 6. collaborative-based wireless IDPS 7. novel R2L 8. Intrusion Detection (√)
Topic 3: Bioinformatics method for predicting thermophilic proteins	1. reliable classifier 2. important sequence 3. thermostability (√) 4. classification 5. sequence 6. Decision 7. predictive successful 8. Matthews correlation	1. pseudo amino acid composition (√) 2. classification rule generator 3. amino acid composition (√) 4. amino acid distribution (√) 5. % non-thermophilic (√) 6. 10-fold cross-validation

		7. 5-fold cross-validation
		8. Decision tree
Topic 4: Thue equation	1. Logarithms	1. f r = g |
	2. u201cThue equationu201d (✓)	2. deg g n · max|
	3. well-known corollary	3. u201cThue equationu201d f (✓)
	4. defined	4. deg f I
	5. practical general	5. + deg g
	6. Theorem	6. f I
	7. integral	7. x =
	8. u201cintegralu201d	8. n =

As can be seen from the table, our proposed algorithm achieved better performance in the simulation test. For the first topic, terms including “Biomedical text mining”, “PubMed Search”, and “text mining” are highlighted in the result. For the second topic, “DARPA intrusion detection evaluation”, “real-time intrusion-detection expert”, “general-purpose intrusion-detection expert”, “Intrusion Detection” were selected as topic keyword. The results are accurate. Comparatively, the original TextRank algorithm mainly picked “general-purpose intrusion detection” as the keyword. In the third simulation test, “PSEAAC”, “AAC”, “amino acid distribution” were correctly extracted from the texts, and these terms referred to exact methodology used in these articles. “Non-thermophilic” was also chosen, which was a exact term to address the research target. Finally, for the topic 4, “Thue equation” were extracted in both our algorithm and TextRank. In this case, the topic extraction is challenging, as the texts are consists of un-regular mathematic symbols that shuffle the order of the natural language representation in sentences.

3.2. The Conclusion of the Result

The result obtained from the simulation result showed that semantic similarity is reliable aid for enhancement of keyword extraction in terms with TextRanking. In light of this effect, further attempt of computational linguistic method will be worthy for our further research.

4. Acknowledgement

This research is funded by the National Natural Science Foundation of China (Grant no. 61202305) and the Fundamental Research Funds for the Central Universities (Project No. 2013PY120).

5. References

[1] Page L, Brin S, Motwani R et al. The PageRank citation ranking: Bringing order to the web[R]. Stanford: Stanford InfoLab, 1999.

[2] Wen C, Huang TZ, Shen ZL. A note on the two-step matrix splitting iteration for computing PageRank. Journal of Computational and Applied Mathematics. 2017 May 1; 315:87-97.

[3] Liu Q, Xiang B, Yuan NJ, Chen E, Xiong H, Zheng Y, Yang Y. An Influence Propagation View of PageRank. ACM Transactions on Knowledge Discovery from Data (TKDD). 2017 Mar 21; 11(3):30.

[4] Mihalcea R, Tarau P. TextRank: Bringing Order into Texts [J]. *UNT*, 2004, 90:404-411.

[5] Mikolov T, Chen K, Corrado G et al. Efficient Estimation of Word Representations in Vector Space [J]. Computer Science, 2013, 103:1301-1309.

[6] Kevin Cohen, Jingbo Xia, Christophe Roeder, Lawrence Hunter. Reproducibility in natural language processing: A case study of two R libraries for mining PubMed/MEDLINE. Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language, Collocated with LREC2016 – 10th Language Resources and Evaluation Conference. Portorož, Slovenia. 2016. Pp.6-12.

[7] Li Yinghui, Xia Jingbo, Zhang silan, Yan Jiakai, Ai Xiaochuan, Dai Kuobin. An Efficient Intrusion Detection System Based on Support Vector Machines and Gradually Feature Removal Method. Expert Systems with Applications. 2012, 39(1): 424-430.

[8] Wang De, Yang Liang, Fu Zhengqi, Xia Jingbo. Prediction of Thermophilic Protein with Pseudo Amino Acid Composition: An Approach from Combined Feature Selection and Reduction. Protein & Peptide letters, 2011, 18(7): 684-689.

[9] Xia Jingbo, Zhang Silan, Chen jianhua. Simplifying method on Algebraic approximation of certain algebraic numbers. Mathematical notes. 2012, 92(3):383-395. Published in Russian in Matematicheskie Zametki, 2012, Vol. 92, No. 3, pp. 426–439.