2019 IEEE 6th International Conference on Industrial Engineering and Applications

Automatic Keyword Extraction Using TextRank

Papis Wongchaisuwat
Department of Industrial Engineering
Kasetsart University
Bangkok, Thailand
e-mail: fengppwo@ku.ac.th

Abstract—**Summarizing and extracting keywords from textual documents is a fundamental task involving in many applications in natural language processing and related fields. This work presents an automatic keyword extraction algorithm based primarily on a weighted TextRank model. In this model, word embedding vectors are used to compute a similarity measure as an edge weight. Incorporating sentence importance scores derived from the TextRank model at a sentence level enhances an overall performance. The proposed algorithm is experimented and compared with the traditional TextRank algorithm as well as the weighted TextRank algorithm with word embedding-based weights.**

Keywords-keyword extraction; TextRank; word embeddings

I. Introduction

In a digital era, data is considered as pervasive and plentiful resource. Significant amount of insightful information can be extracted and learned from the data. Structured data such as a relational database with clearly defined data patterns is traditionally easier to handle. However, a vast majority of available data nowadays is in an unstructured format. The unstructured data includes a digital image, a music file, video data, and textual data. Natural Language Processing or NLP mainly aims to process human language in an automated fashion. Understanding and extracting meaningful information from text is one of essential tasks in NLP. To be specific, manually summarizing long textual data into words or short phrases is very time-consuming. An automatic keyword extraction system is introduced in order to address this difficulty. Given a textual document, a keyword extraction task automatically identifies terms which describe its main subject.

An automatic algorithm to extract keywords from a textual document is proposed in this work. Specifically, important words representing a key idea are summarized from a whole document. The proposed algorithm relies primarily on a TextRank algorithm which is a graph-based ranking model built upon a well-known PageRank algorithm. The PageRank algorithm determines an importance of a webpage based on its reference information. Intuitively, a webpage frequently cited by many other webpages is relatively significant and therefore should be given a high importance score. In essence, the PageRank algorithm is a graph-based ranking algorithm which computes an importance score of each vertex based on information from an entire graph. In a view of the PageRank model, the TextRank algorithm builds a graph from textual documents where each vertex represents a text unit and each edge is a relationship between adjacent text units. The TextRank algorithm can be applied at various level such as words, phrases and sentences depending on choices of text units. For instance, applying the TextRank algorithm at a sentence level yields an importance score for each sentence. In order to extract keywords from textual data, this work applies the TextRank algorithm at a word level.

The traditional TextRank which incorporates a strength of a connection between edges is commonly known as a weighted TextRank algorithm. Variation of weight values are used in this model. A similarity measure between words based on embedding vectors can be used as an initial weight in the weighted TextRank algorithm. In this work, a combination of a TextRank-based heuristic approach and an embedded word representation is introduced in order to improve an accuracy of the traditional TextRank algorithm. In addition, this work takes into account not only an importance of words but also an importance of sentence where keywords are extracted from. The sentence importance score is initially computed based on the TextRank algorithm at a sentence level. An initial edge weight in the word-level weighted TextRank model is further adjusted with its corresponding sentence score.

The proposed algorithm is experimented and compared with the traditional TextRank algorithm as well as the weighted TextRank algorithm with word embedding-based weights. In a case study experimented in this work, data is collected from Industrial Engineering senior projects over the past 3 years. Abstracts from these projects are used as original input data where keywords are extracted from. Keywords stated in the project reports along with abstracts are employed as benchmark. The algorithm proposed in this work provides relatively better performance than both the traditional TextRank and the word-embedding weighted TextRank models.

The proposed approach distinguishes this work from others. The main contribution in this work is a novel algorithm based on an adjusted weighted TextRank algorithm. A combination of sentence and word importance scores is introduced in this work. It aims to take into account a surrounding context of extracted keywords presented in the corresponding sentence. To the best of my knowledge, no prior work exists incorporating an importance of sentence in order to assess an importance of word in a view of the TextRank model.

978-1-7281-0851-3/19/$31.00 ©2019 IEEE

In Section 2, a literature review is provided. A detail discussion of the proposed TextRank-based algorithm is described in Section 3. The results of the proposed algorithm based on the case study data are provided in Section 4. Further discussions are reported in Section 5. Lastly, section 6 states conclusions and future work.

II. Relevant Work

Extracting keywords from a given textual document aims to analyze and summarize long text into a few compact words or phrases. Extracting keywords is a challenging research problem with various practical applications. Still, substantial research exists for developing novel systems to automatically extract keyword from documents. Kaur and Gupta [7] provided a review study on multiple keyword extraction techniques. These existing algorithms are built on various idea including co-occurrence statistical information [11, 15], an iterative reinforcement approach [18], conditional random fields [20], domain-specific information [1]. Some of these work rely on a supervised approach which requires additional annotated information in a training data set [3, 5, 6, 9]. In contrast, others [4, 17, 10] are purely unsupervised like this work.

This work is essentially built upon an adjusted TextRank algorithm. Mihalcea and Tarau [12] firstly introduced a graph-based ranking model named TextRank for extracting keywords from an original textual document. It is in a fully unsupervised manner which does not require any training corpus or an expensive annotation process. According to the TextRank algorithm, an original text is split into basic text units which are further mapped to vertices in a graph. In this graph, an edge represents the co-occurrence relationship within a pre-defined window size between adjacent text units. To address a possible difference of strength between various pairs of text units, each edge can be assigned with different weight value. The ranking algorithm is iterated until it converges. A final score for each vertex in the graph is eventually obtained. Li and Wang [8] introduced the Document Frequency Accessor Variety to select keyword candidates prior to applying the TextRank algorithm. This approach was shown to improve a quality of the keyword extraction system.

Another enhancement of the original model is using different similarity measures in the weighted TextRank algorithm. Federico et. al. [2] proposed a variation of the graph construction process compared to the TextRank algorithm. Various similarity measures as edge weights were also introduced. These measures improved a performance of the algorithm significantly. Additionally, a state-of-the-art Word2Vec word embedding method which map words into vectors is taken into account when computing word similarity as an edge weight in a graph [19, 21]. Word2Vec takes a large text corpus as an input in order to construct word vectors positioned in a vector space. These word vectors are employed to compute word similarity. This work enhances existing Word2Vec weighted TextRank models by incorporating sentence importance scores. These sentence scores are calculated from a sentence-level TextRank model.

III. Methodology

The proposed algorithm is built as a pipeline based on a variation of TextRank model. The TextRank model applies a well-known PageRank algorithm with Textual data. The graph-based PageRank algorithm is used to measure the relative importance of website pages within a hyperlink set [14]. After assigning each website page as a node, the algorithm computes an importance score for each node within the graph. Connecting one node to another implies a vote or a recommendation casting between these 2 connected nodes. The higher the number of votes, the more significant the corresponding node is. The importance score for each node is computed based on a probability of randomly going from an interested node to other nodes in the graph. Given a graph $G(V, E)$, the importance score of node V_i denoted as $S(V_i)$ is formulated as follows [12]:

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

where $In(V_i)$ and $Out(V_i)$ are a set of predecessors and a set of successors of V_i

d is a damping factor ranging between 0 and 1

A connection between nodes in the graph can be handled differently by incorporating connection's strength into the model. Specifically, an edge weight w_{ij} corresponding to node V_i and node V_j is considered when computing an importance score. A formula for a weighted score $WS(V_i)$ is defined as follows [12]:

$$WS(V_i) = (1-d) + d \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

In this graph, the proposed algorithm consists of: A) *Sentence score computation*; B) *Keyword score computation*. The first phase applies the TextRank algorithm at a sentence level resulting in an importance score for each sentence. In the second phase, a variation of the TextRank algorithm at a word level is implemented and combined with embedded word representations. The sentence scores from the first phrase are taken into account when computing word scores in the second phrase. Finally, words associated with top scores are retrieved as extracted keywords. All implementations are in python with Word2Vec implementation from genism [16]. Pre-trained Word2Vec models used in this work is trained on part of Google News dataset.

A. Sentence Scores Computation

In order to compute an importance score for each sentence, original documents are initially split into multiple sentences. A fully-connected graph is constructed where a node and an edge represent a sentence and a similarity score between 2 adjacent nodes, respectively. Formally, a sentence graph $G_s = (V, E)$ is an undirected graph with a set of sentences V and a set of edges E. Each edge is weighted accordingly with sentence similarity scores. The similarity score between 2 sentences is based on their common words and sentences' length as defined in [2]. The TextRank algorithm is then implemented on the graph G_s until convergence. The importance score for a sentence V_i defined as $WS_s(V_i)$ is retrieved from the algorithm.

B. Keyword Scores Computation

The weighted TextRank algorithm is applied at a word level for extracting words or phrases. After tokenizing original documents, an undirected graph G_w is constructed while a token (a word) is considered as a node. A co-occurrence relationship among words are added to an edge connecting between adjacent nodes. A window size of co-occur words is taken into consideration in this relationship. Specifically, an edge between any 2 nodes are added if and only if a distance between 2 corresponding words are less than a pre-specified window size.

An edge weight w_{ij} is partially obtained from a similarity between words V_i and V_j. In order to enhance a performance of the typical TextRank algorithm, the semantic similarity between 2 words are considered. To be specific, a vector representation for each word is retrieved from the Word2Vec model [13]. A cosine similarity between word vectors are computed and incorporated into the weighted TextRank formula [21]. Out-Of-Vocabulary denoted as OOV words are unseen words observed only in a test set. This implies that word vectors for OOV words cannot be retrieved from the Word2Vec model. In order to address OOV words, a pre-defined value of similarity is used as a default.

This work is built on an assumption that an importance of each word is obtained from a word itself and a sentence where it is drawn from. Sentence scores also contain useful insight to help elevate a performance of the algorithm. According to the proposed algorithm, the sentence scores WS_s calculated from the TextRank algorithm in the previous step are normalized to a 0 and 1 range. The edge weight computed from the Word2Vec model is further adjusted with these sentence scores. Essentially, word vectors corresponding to these 2 adjacent nodes V_i and V_j are retrieved from the Word2Vec model. The cosine similarity is computed between these 2 word vectors. In addition, a set of sentence scores corresponding to all sentences where both V_i and V_j are drawn from are collected. The average sentence scores across this set is then calculated. A final edge weight w_{ij} is a multiplication of the average sentence scores and the word similarities. Finally, the weighted TextRank formula with the final edge weight is iterated until convergence. The final score WS_w for each word is retrieved.

After sorting the final word scores in a reversed order, words corresponding to top scores are collected as potential keywords. These potential keywords are post-processed in order to search for multi-word keywords. Particularly, adjacent potential keywords contained in the original documents are combined into a single phrase keyword.

IV. Results

To test the algorithm, undergraduate senior projects at the department of Industrial Engineering, Kasetsart University are collected over the past 3 years. After removing missing or incomplete data, 200 projects are retrieved. For each project, an abstract is used as an original text from which keywords are extracted. A list of keywords mentioned in the project report is considered as a gold standard. A sample input and output from our algorithm are shown in Table I.

TABLE I. A Sample of Input and Output from the Proposed Algorithm

Abstract (input)	The objective of this study is to investigate the investment possibility of purchasing printing machine to store in the factory instead of printing from suppliers. Project feasibility study has been applied to analyze in three steps. First, technical study is the analysis of demand by forecasting 3 methods, the moving average, the weighted moving average and exponential smoothing to find the method which provides the least Mean Absolute Percentage Error (MAPE). The study indicated that exponential smoothing had the least MAPE. From calculation, the appropriate speed of the machine is 33.21 meters per minute. ZRY-420 AUTOMATIV UV FLEXOGRAPHIC PRINTING MACHINE from China is the most suitable for this project. Second, financial study is the analysis of project financial and finally, evaluate project feasibility. The expected Minimum Attractive Rate of Return (MARR) is 9.96 percent. From the study's result indicated that Internal Rate of Return (IRR) was 48 percent higher than MARR approximately 38.04 percent. Consequently, this project is feasible and the payback period is 1 year 7 months.
Keywords (gold standard)	Feasibility study, printing
Extracted keywords (output)	Study, printing machine, project feasibility, rate

Precision, recall and F1-score are computed to evaluate a performance of the proposed algorithm. These evaluation metrics are further compared against those computed from the traditional TextRank model [12] and the Word2Vec weighted TextRank model [21]. After precision, recall and F1-score are assessed for each project (each instance), they are averaged across all collected instances. A comparison of all evaluation metrics computed from these 3 algorithms is provided in Table II.

TABLE II. Evaluation Metrics Compared across All 3 TextRank-based Algorithms

	Precision	Recall	F1-score
Traditional TextRank	0.19	0.63	0.26
Word2Vec-based weighted TextRank	0.24	0.38	0.27
Proposed TextRank-based model	**0.26**	0.4	**0.29**

V. Discussions

In this paper, the automate keyword extraction algorithm is developed based on the TextRank algorithm. Even though a specific case study of Industrial Engineering senior projects is used to assess the proposed algorithm, it can be adapted and applied to other data sets. Among different models experimented, the proposed algorithm gives better performances than the traditional TextRank and the Word2Vec weighted TextRank models. Overall, the proposed model achieves 0.26 averaged precision, 0.4 averaged recall and 0.28 averaged F1-score.

Compared to [12] and [21] which are the main motivations of this work, the proposed algorithm in this paper adjusts an edge weight added to the graph. Specifically, [12] introduced the traditional TextRank algorithm which took into account an co-occurrence of terms within a pre-specified window size in order to construct a graph. Zuo et. al. [21] further enhance the TextRank algorithm by using a Word2Vec similarity as an edge weight. Both work only consider an importance of words without assessing an importance of the sentence where words are drawn from. In constrast, information regarding a sentence is incorporated when evaluating an importance of words in this work. According to the proposed algorithm, the sentence scores are first computed from the sentence-level TextRank algorithm. These scores are then used to adjust an edge weight in the weighted TextRank at a word level.

The proposed algorithm is based on word embedding vectors. The idea of word embedding becomes more popular nowadays and it is considered as state-of-the-art in many NLP applications. The pre-trained word vectors obtained from the Word2Vec model are considered. This Word2Vec model was trained on a large corpus, i.e., Google News dataset. This model contains 300-dimensional vectors for 3 million words and phrases. This pre-trained word vectors is well-known and commonly used in many research work. Due to a large training corpus, almost all terms in the data set used in this paper are contained in the vocabulary set. For Out-Of-Vocabulary (OOV) words, the default value is assumed.

VI. Conclusion and Future Work

The proposed algorithm aims to process and automatically extract keywords from long textual data. It is based mainly on the TextRank algorithm with a further adjustment on an edge weight in the graph constructed from an input text. Specifically, each vertex represents a basic text units while each edge is weighted with Word2Vec similarity measures adjusted with a sentence importance score. Superior performance of the proposed algorithm can possibly be achieved if an edge weight better represents the true relationship among adjacent text units. The edge weight is based mainly on both sentence and word importance score. Hence, more accurate sentence similarity potentially enhances the overall performance of the algorithm. Taking additional domain-specific knowledge into a consideration is likely to enable the proposed algorithm to extract more insightful keywords. Another future work includes applying the proposed algorithm to multiple data sets in order to verify a generalizability of the algorithm in various fields.

References

[1] M. A. Andrade and A. Valencia, “Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families,” Bioinformatics, vol. 14, 1998, pp. 600-607, doi: 10.1093/bioinformatics/14.7.600

[2] F. Barrios, F. Lopez, L. Argerich and R. Wachenchauzer, “Variations of the Similarity Function of TextRank for Automated Summarization,” CoRR, vol. abs/1602.03606, 2016.

[3] P. Bhaskar, K. Nongmeikapam, S. Bandyopadhyay, “Keyphrase extraction in scientific articles: A supervised approach,” Proc. of COLING, 2012, pp.17-24.

[4] D. B. Bracewell, F. Ren and S. Kuriowa, “Multilingual single document keyword extraction for information retrieval,” 2005 International Conf. on Natural Language Processing and Knowledge Engineering, 2005, pp. 517-522, doi:10.1109/NLPKE.2005.1598792.

[5] A. Hulth, “Improved automatic keyword extraction given more linguistic knowledge,” Proc. of the 2003 Conf on Empirical methods in Natural Language Processing, 2003, pp. 216-223, doi: 10.3115/1119355.1119383.

[6] K. Zhang, H. Xu, J. Tang and J. Li, “Keyword extraction using support vector machine,” Advanced in web-age information management, Springer, 2006.

[7] J. Kaur and V. Gupta, “Effective approaches for extraction of keywords.,” International journal of computer science issues, vol. 7, 2010, pp. 144-148.

[8] G. Li and H. Wang, “Improved automatic keyword extraction based on TextRank using domain knowledge,” Natural Language Processings and Chinese computing, Springer, 2014, pp. 403-413.

[9] M. Litvak and M. Last, “Graph-based keyword extraction for single-document summarization,” Proc. of the workshop on multi-source multilingual information extraction and summarization, ACL, 2008, pp. 17-24.

[10] F. Liu, D. Pennell, F. Liu and Y. Liu, “Unsupervised approaches for automatic keyword extraction using meeting transcripts,” Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, ACL, 2009, pp. 620-628.

[11] Y. Matsuo and M. Ishizuka, “Keyword extraction from a single document using word co-occurrence statistical information,” International journal on artificial intelligence tools, vol. 13, 2004, pp. 157-169.

[12] R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Text,” Conf. on Empirical Methods in Natural Language Processing, ACL, 2004, pp. 404-411.

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, “Distributed representations of words and phrases and their compositionality,” Proc. of the 26th International Conference on Neural Information Processing Systems, Curran Assoc. Inc., 2013, pp. 3111-3119.

[14] L. Page, S. Brin, R. Motwani and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” 1999.

[15] G.K. Palshikar, “Keyword extraction from a single document using centrality measures,” Proc. of the 2nd international conference on Pattern recognition and machine intelligence, Springer-Verlag, 2007, pp. 503-510.

[16] R. Rehurek and P. Sojka, “Software framework for topic modelling with large corpora,” 2010.

[17] S. Rose, D. Engel, N. Cramer and W. Cowley, “Automatic Keyword Extraction from Individual Documents,” Text Mining, 2010, doi:10.1002/9780470689646.ch1.

[18] X. Wan, J. Yang and J. Xiao, “Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction,” Proc. of the 45th annual meeting of the association of computational linguistics, 2007, pp. 552-559.

[19] H.K. Yaakov, “Automatic extraction of keywords from abstracts,” Knowledge-based intelligent information and engineering systems, Springer, vol. 2773, 2003.

[20] C. Zhang, “Automatic keyword extraction from documents using conditional random fields,” Journal of Computational Information Systems, vol. 4, 2008, pp. 1169-1180.

[21] X. Zuo, S. Zhang and J. Xia, “The enhancement of TextRank algorithm by using word2vec and its application on topic extraction,” Journal of Physics: conference series, vol. 887, doi: 10.1088/1742-6596/887/1/012028.