Automatic Key Phrase Extraction using Phrase Embeddings - Key2Vec Approach

Anjana Anand
Department of Computer Science
University of Illinois at Chicago
aanand31@uic.edu

Eunice Daphne John Kanagaraj
Department of Computer Science
University of Illinois at Chicago
ejohnk2@uic.edu

ABSTRACT

Extraction of important topical words and phrases from documents, commonly known as terminology extraction or automatic key phrase extraction is a hot topic in the research field. It comes under one of the crucial tasks for the purposes of automatically extracting structured information from unstructured datasets. Key phrases provide a concise description of a document's content. They are useful for document search, clustering, categorization, and summarization; help in building a content-based recommendation system as you can quantify semantic similarity with other documents. In this paper we are implementing a key2vec unsupervised approach which uses phrase embeddings for extracting and ranking key phrases from scholarly documents. In this project, the documents are processed to get unigram tokens and multi-word phrase embeddings. These multi-word phrase embeddings are used for thematic representation of scientific articles and ranking of key phrases extracted from them using theme-weighted PageRank. The results of the evaluation are compared with the results of the traditional approaches using www data set.

INTRODUCTION AND BACKGROUND

Automatic keyphrase extraction concerns "the automatic selection of important and topical phrases from the body of a document" (Turney, 2000). In other words, its goal is to extract a set of phrases that are related to the main topics discussed in a given document. Keyphrases can help users get a feel for the content of a collection, provide sensible entry points into it, show how queries can be extended, facilitate document skimming by visually emphasizing important phrases; and offer a powerful means of measuring document similarity. Due to its widespread use, keyphrase extraction has received significant attention from researchers belonging to fields like Machine Learning, Deep Learning, Neural Networks, Natural Language Processing, Web Scraping, etc.

(i) Existing Challenges :

Some of the major challenges that a key phrase extraction IR system faces are mentioned here. The number of scholarly documents available in the web is growing rapidly now a days. Processing the documents, handling its structural inconsistency and developing various strategies that can extract key phrases with good accuracy in different domain has been a challenging task. Overgeneration errors occur when a system correctly predicts a candidate as a keyphrase because it contains a word that appears frequently in the associated document, but at the same time erroneously outputs other candidates as keyphrases because they contain the same word. Infrequency errors occur when a system fails to identify a keyphrase owing to its infrequent presence in the associated document. Redundancy errors occur when a system correctly identifies a candidate as a keyphrase, but at the same time outputs a semantically equivalent candidate as a keyphrase. An evaluation error occurs when a system outputs a candidate that is semantically equivalent to a gold standard keyphrase but is considered erroneous by a scoring program because of its failure to recognize that the predicted phrase and the corresponding gold keyphrase are semantically equivalent. All these errors sum up in reducing the accuracy of the keyphrases generated. Every IR system built to extract keyphrases has to overcome these challenges to some extent in order to obtain good results.

(ii)Popular Methodologies and its Limitations:

The supervised methods employ choosing the best keywords from a prepared set of keywords, which is likely to contain topics from all genres and fields of interest. This method requires labeled documents with their tags or keywords. A model is developed to learn the ways in which the tags and keywords can be associated with a text and how they can be generated from a text. While this can produce interpretable rules as to what characterizes a key-phrase, but the greatest challenge in this method is the availability of training data, tags and keywords for large number of texts whose topics encompass all genres of interest like scientific journals, news, business, education, entertainment, sports etc.

The unsupervised methods eliminate the need for training data. Unlike supervised methods, this uses the structure of the text itself and generates keywords and phrases from the text itself using its properties. This approach holds undeniable importance as this can be applied across all languages and domains.

Vector space model is well-known and the most used model for text representation in text mining approaches. Specifically, the documents represented in the form of feature vectors are located in multidimensional Euclidean space. This model is suitable for capturing simple word frequency, however structural and semantic information are usually disregarded.

Graph-based text representation is known as one of the best solutions which efficiently address these problems. Graph is a mathematical model, which enables exploration of the relationships and structural information very effectively. The document is models as graph where terms are represented by vertices and relations between terms is represented by edges.

(iii)Motivation and Current Work :

With recent advancements in deep learning techniques applied to natural language processing, the trend is to represent words as dense real-valued vectors, popularly known as word embeddings. The embedding vectors are supposed to preserve the semantic and syntactic similarities between words. They have been shown to be useful for several NLP tasks, like part-of-speech tagging, chunking, named entity recognition, semantic role labeling, syntactic parsing, and speech processing, among others(Collobertetal.,2011). Word embeddings have already shown promising results in the process of keyphrase extraction from scientific articles. We have experimented with domain-specific embeddings on scientific articles.

In this project, Domain specific word embedding is employed to extract the key phrases from the scholarly documents. The obtained key phrases are ranked using a theme-weighted PageRank algorithm. The thematic weight of the multi-word phrase embeddings shows how similar the candidates are to the theme of the scholarly document. The process of extraction is mentioned below in the following content.

PROPOSED APPROACH

The keyphrase extraction on scientific documents is carried out by first representing each candidate keyphrase through its phrase or word embedding and then ranking those candidate keyphrases by a theme-weighted PageRank algorithm (Langville and Meyer, 2004). The approach can be split into three steps: text pre-processing, embedding model, candidate selection and scoring, candidate ranking. The details of the implementation are explained below.

Text Pre-processing

For text pre-processing, we use Spacy as the NLTK toolkit. Spacy comes with pre-trained statistical models and word vectors. It features state-of-the-art speed, convolutional neural network models for tagging, parsing and named entity recognition. Using Spacy, we split each text document into sentences and tokenize these sentences on whitespace to get the unigram tokens. We also obtain the noun phrases and named entities from each document keeping track of the word offset to remove word or phrase overlap between noun phrase and named entity. Each of the unigrams and the multi-word tokens (noun phrase and named entity) are pre-processed further by filtering out the following: removal of stopwords; removal of punctuations except '-'; noun phrase and named entity that are entirely numeric; removal of dangling characters, punctuations and whitespace. The named entities belonging to the following types are removed as they don't contribute to a candidate phrase: DATE, TIME, PERCENT, MONEY, QUANTITY, ORDINAL, CARDINAL. Similarly, the leading and ending tokens of multi-word tokens belonging to the following types are removed: Interjection, Auxiliary, Coordinating Conjunction, Adposition, Interjection, Numeral, Particle, Pronoun, Subordinating Conjunction, Punctuation, Symbol, Other. Also removed are the determiners, if they are the first token; common adjectives and reporting words, if they are the first or last token. The remaining unigrams and multi-word tokens are merged together in the order in which they appear in the original sentence.

Embedding Model

The embedding model plays a major role in the implementation of the Key2Vec. We experimented with two types of embedding models. They are,

(i)Phrase Embedding using Fasttext:

A multi-word domain-specific phrase embedding model was trained using Fasttext, which captures both the semantic and morphological similarities between words whereas other embedding techniques like Glove and Word2Vec consider only the semantic similarity between the words. The Fasttext model was trained on scientific document abstracts collected from arxiv.org. These document abstracts were obtained using the API provided by arxiv.org which allows bulk access to the papers uploaded on their site. A total of 50,945 documents belonging to the following fields were collected: Astro Physics, Nuclear Experiment, Mathematics, Computer Science, Condensed Matter, Mathematical Physics, Quantum Physics, Physics, General Relativity and Quantum Cosmology, Statistics, Quantitative Biology, Nonlinear Sciences, Nuclear Theory, Quantitative Finance, High Energy Physics. These document abstracts were pre-processed by the methods mentioned above and trained using Fasttext-skipgram model with the following parameters: window size – 5, negative sampling, dimension – 100, number of epochs – 10.

(ii)Word embedding from WIKI:

Fasttext provides pre-trained word vectors for English, trained on Wikipedia. These vectors in dimension 300 were obtained using the skip-gram model with default parameters. This model was downloaded and used for word embedding. The candidate keyphrases which are obtained after pre-processing contain both unigrams and multi-word tokens therefore these word vectors are modified to represent the phrases. Each candidate keyphrase is tokenized on whitespace and the word vectors of each of these tokens are added together and divided by the total number of tokens in that particular candidate keyphrase to get the phrase vector for that candidate. When using this model, the vector of the keyphrases is obtained as an average of each individual word vector.

Candidate Selection and Scoring

After pre-processing the dataset, we obtain a list of keyphrases for each of the document. These keyphrases are called the candidate keyphrases which are used later for scoring and ranking. A simple representation of these candidate keyphrase will be ($C_{di} = \{c_1, c_2, ..., c_n\}_{di}$). The C_{di} represents the candidate keyphrase list for document d_i.

A theme vector (τˆd_i) is assigned to each of the documents. We consider the first line of every document to be the theme of that document. The extracted theme is pre-processed by the steps mentioned above and the thematic phrases are obtained. To get the vector representation of the thematic phrases we add the phrase vectors of each of the thematic phrases from the embedding model. We also get the vector representation for the candidate keyphrases from the model represented by (ˆc_k; k ϵ {1...n}).

We now calculate the cosine distance between the theme vector (τˆd_i) and the vector for each candidate keyphrase (ˆc_k), that will assign a score between 0 and 1 indicating the semantic closeness of the keyphrase to the document theme.

Candidate Ranking

A theme-weighted PageRank algorithm was implemented to perform the candidate ranking. A bidirected graph with the candidate keyphrases as vertices and an edge if two candidate keyphrase co-occur within a window size of 5, is constructed. The weights $sr(c_j^{di}, c_k^{di})$ of these edges are calculated using the following formulas from .

$$semantic(c_j^{d_i}, c_k^{d_i}) = \frac{1}{1 - cosine(c_j^{d_i}, c_k^{d_i})} \tag{1}$$

$$cooccur(c_j^{d_i}, c_k^{d_i}) = PMI(c_j^{d_i}, c_k^{d_i}) \tag{2}$$

$$sr(c_j^{d_i}, c_k^{d_i}) = semantic(c_j^{d_i}, c_k^{d_i}) \times cooccur(c_j^{d_i}, c_k^{d_i}) \tag{3}$$

The edge weight is calculated as a product of the semantic similarity and the frequency of co-occurrence. The semantic similarity is calculated from equation (1). The frequency of co-occurrence is obtained from Point-wise Mutual Information (PMI) calculated from equation (2). Now, the final PageRank score $R(c_j^{di})$ of a candidate keyphrase is calculated using the equation given below. We set the damping factor to be 0.85 and the $out(c_k^{di})$ is the out-degree of that vertex.

$$R(c_j^{d_i}) = (1-d)w_{c_j}^{d_i} + d \times \sum_{c_k^{d_i} \in \varepsilon(c_j^{d_i})} \left(\frac{sr(c_j^{d_i}, c_k^{d_i})}{|out(c_k^{d_i})|}\right) R(c_k^{d_i})$$

After the PageRank scores of all the candidate keyphrases from each of the document are calculated we evaluate its performance.

DATASET

The World Wide Web (WWW) dataset containing research papers from the WWW conference is used for evaluation. The dataset contains 1330 titles and abstracts of scientific articles and their corresponding gold standards. These documents are available with and without the POS tag. We use the data without POS tag. For evaluation we use 425 documents from the dataset, a list of these documents is provided as a separate “queries.overlap.list” file.

RELATED WORK :

Supervised Approaches:

Zhang C. et al. (2008) in [1] implement keyword extraction method from documents using Conditional Random Fields (CRF). CRF model is a state-of-the-art sequence labeling method, which can use the features of documents more sufficiently and efficiently and considers a keyword extraction as the string labeling task. CRF model outperforms other ML methods such as SVM, Multiple Linear Regression model, etc.

Wang (2006) in [8] follows these features in order to determine whether a phrase is a keyphrase: TF and IDF, appearing in the title or headings (subheadings) of the given document, and frequency appearing in the paragraphs of the given document in the combination with Neural Networks are proposed.

Krapivin et al. (2010) in [2] use NLP techniques to improve different machine learning approaches (SVM, Local SVM, Random Forests) to the problem of automatic keyphrases extraction from scientific papers. Evaluation shows promising results that outperform state-of-the-art Bayesian learning system KEA on the same dataset without the use of controlled vocabularies.

Turney (2003) in [9] implements enhancements to the Kea keyphrase extraction algorithm by using statistical associations between keyphrases and enhances the coherence of the extracted keywords.

Graph based Approach :

Lahiri et al. (2014) in [3] extract keywords and keyphrases form co-occurrence networks of words and from noun-phrases collocations networks. Eleven measures (degree, strength, neighborhood size, coreness, clustering coefficient, structural diversity index, page rank, HITS hub and authority score, betweenness, closeness and eigenvector centrality) are used for keyword extraction from directed/undirected and weighted networks.

Litvak and Last (2008) in [5] compare supervised and unsupervised approaches for keywords identification in the task of extractive summarization. The approaches are based on the graph-based syntactic representation of text and web documents.

Mihalcea (2004) in [6] presents an extension to earlier work [34], where the TextRank algorithm is applied for the text summarization task powered by sentence extraction. On this task TextRank performed on a par with the supervised and unsupervised summarization methods, which motivated the new branch of research based on the graph-based extracting and ranking algorithms

Huang et al. [4] propose an automatic keyphrase extraction algorithm using an unsupervised method based on connectedness and betweeness centrality.

Unsupervised Approach :

HaCohen-Kerner (2003) in [10] presents a simple model that extracts keywords from abstracts and titles. Model uses unigrams, 2-grams and 3-grams, and stopwords list. The highest weighted group of words (merged and sorted n-grams) is proposed as keywords.

Pudota et al. (2010) in [11] design domain independent keyphrase extraction system that can extract potential phrases from a single document in an unsupervised, domain-independent way. They engaged n-grams, but they also incorporate linguistic knowledge (POS tags) and statistics (frequency, position, lifespan) of each n-gram in defining candidate phrases and their respective feature sets.

Yang et al. (2013) in [7] focused on keyword extraction based on entropy difference between the intrinsic and extrinsic modes, which refers to the fact that relevant words significantly reflect the author's writing intention. Their method uses the Shannon's entropy difference between the intrinsic and extrinsic mode, which refers that words occurrences are modulated by the author's purpose, while the irrelevant words are distributed randomly in the text.

ALGORITHM FOR COMPARING EXISTING RESULTS

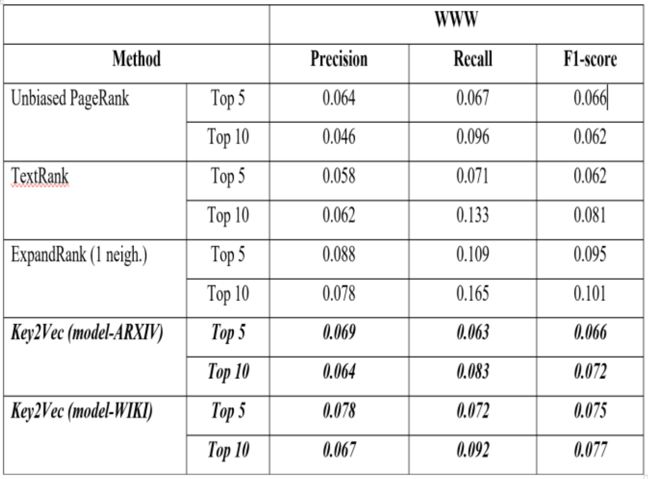
Text Rank : This algorithm is essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of "voting" or "recommendation". When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex.

Expand Rank : ExpandRank is a TextRank extension that exploits nearest neighbor documents to provide more knowledge to improve keyphrase extraction. Each document is represented by a term vector where each vector dimension corresponds to a word type present in the document and its weight is computed by Tf ldf.

Page Rank : The PageRank algorithm gives each page a rating of its importance, which is a recursively defined measure whereby a page becomes important if important pages link to it. This definition is recursive because the importance of a page refers back to the importance of other pages that link to it.

EXPERIMENTS AND RESULTS

The performance of this Key2Vec implementation is evaluated on the www dataset. The performance parameters are Precision, Recall and F1-score. We compare our results with some of the other methods of keyphrase extraction like the unbiased PageRank algorithm, TextRank and ExpandRank. We have used the results of TextRank and ExpandRank as reported by (Sujatha et al., 2014) using the www dataset. The Key2Vec was implemented using both the Fasttext model trained using ARXIV (Key2Vec – ARXIV) and the Fasttext model from WIKI (Key2Vec - WIKI) for the phrase and word vectors. The performance comparison of Key2Vec with other methods is given in the Table 1. A comparison of F1-score between all these methods on top 5 and top 10 keyphrases are visualized in Fig.1 and Fig.2.



		WWW		
Method		Precision	Recall	F1-score
Unbiased PageRank	Top 5	0.064	0.067	0.066
	Top 10	0.046	0.096	0.062
TextRank	Top 5	0.058	0.071	0.062
	Top 10	0.062	0.133	0.081
ExpandRank (1 neigh.)	Top 5	0.088	0.109	0.095
	Top 10	0.078	0.165	0.101
Key2Vec (model-ARXIV)	***Top 5***	***0.069***	***0.063***	***0.066***
	Top 10	***0.064***	***0.083***	***0.072***
Key2Vec (model-WIKI)	***Top 5***	***0.078***	***0.072***	***0.075***
	Top 10	***0.067***	***0.092***	***0.077***

Table 1

From the results, it can be interpreted that the number of documents trained for building the embedding model increases the overall performance of the keyphrase extraction. In the Fasttext model built with ARXIV the vocabulary size is 425,093 whereas in the model built from WIKI the vocabulary size is 999,994. Thus, Key2Vec (WIKI) has a better performance than the Key2Vec (ARXIV). Considering the evaluation at Top 5, Key2Vec (WIKI) has performed better than TextRank and unbiased PageRank. But, ExpandRank has better performance over Key2Vec (WIKI) at Top 5. Similarly, considering the evaluation at Top 10; Key2Vec (WIKI) has better performance over unbiased PageRank. But ExpandRank and TextRank have better performance over Key2Vec (WIKI) at Top 10.

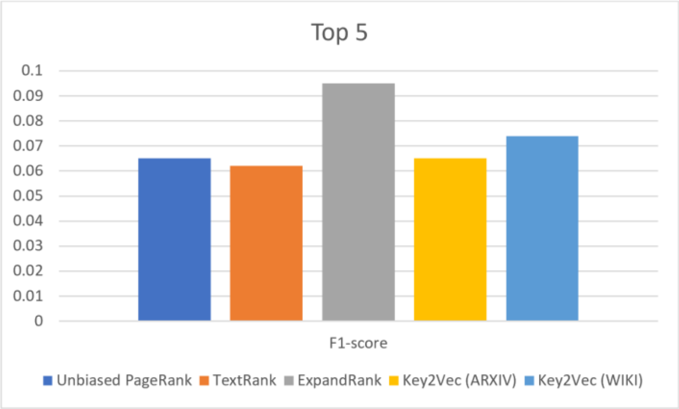


Fig. 1

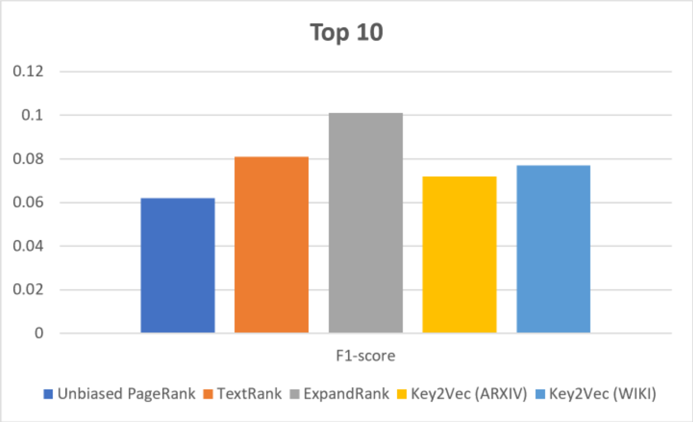


Fig. 2

CONCLUSION AND FUTURE WORK :

In this paper we performed automatic key phrase extraction and ranking of the key phrases from scientific articles by constructing thematic representation of the articles and assigning thematic weights to candidate keyphrases through word/phrase embeddings. The ranking of the keyphrases is done by theme-weighted PageRank. A comparative study was made with the existing models and the experimental evaluations of the key2vec model produced results equivalent to the certain benchmark results.

In future, we plan to improvise the accuracy of the proposed model by the below mentioned points.

While much recent work has focused on algorithmic development, keyphrase extractors need to have a deeper understanding of a document in order to reach the next level of performance. Such an understanding can be facilitated by the incorporation of background knowledge.

Currently the model was trained by 50k arxiv documents belonging to multiple domains. If it was trained with much higher number of documents, the accuracy of the prediction might show significant improvement.

The implemented algorithm can be further extended with position biased page rank algorithm which provides leverage to the words, based on the word position. The evaluations of this can be compared with the proposed methodologies and can be checked for improved accuracy.

Our experiments are focused on WWW datasets of research papers. In the future, it would be interesting to explore the performance of the model on varieties of data set like SemEval, KDD and Inspec.

REFERENCES :

[1] C. Zahang, H. Wang, Y. Liu, D. Wu, Y. Liao, B. Wang, “Automatic Keyword Extraction from Documents Using Conditional Random Fields” in Journal of CIS 4:3(2008), pp. 1169-1180, 2008.

[2] M. Krapivin, A. Autayeu, M. Marchese, E. Blanzieri, N. Segata, “Keyphrases Extraction from Scientific Documents: Improving Machine Learning Approaches with Natural Language Processing” in Proc. of 12th Int. Conf. on Asia-Pacific Digital Libraries, ICADL 2010, Gold Coast, Australia, LNAI v.6102, pp. 102-111, 2010.

[3] S. Lahiri, S. R. Choudhury, C. Caragea, “Keyword and Keyphrase Extraction Using Centrality Measures on Collocation Networks”, arXiv preprint arXiv:1401.6571, 2014.

[4] C. Huang, Y. Tian, Z. Zhou, C.X. Ling, T. Huang “Keyphrase extraction using semantic networks structure analysis” in IEEE Int. Conf. on Data Mining, pp.275-284, 2006

[5] M. Litvak, M. Last, “Graph-based keyword extraction for singledocument summarization” in ACM Workshop on Multi-source Multilingual Information Extraction and Summarization, pp.17- 24, 2008.

[6] R. Mihalcea, “Graph-based ranking algorithms for sentence extraction, applied to text summarization” in Proc. of 42nd Annual Meeting of the Assoc. for Comput. Linguistics, ACL 2004, 2004.

[7] Z. Yang, J. Lei, K. Fan, Y. Lai, “Keyword extraction by entropy difference between the intrinsic and extrinsic mode” in Physica A: Statistical Mechanics and its Applications, V. 392, I. 19, pp. 4523-4531, 2013

[8] J. Wang, H. Peng, J.-S. Hu, “Automatic Keyphrases Extraction from Document Using Neural Network”, 4th Int. Conf. ICMLC 2005, Guangzhou, China, LNCS V.3930, pp. 633-641, 2006.

[9] P. D. Turney, “Coherent Keyphrase Extraction via Web Mining” in Proc. of IJCAI 2003, pp. 434-439, San Francisco, USA, 2003.

[10] Y. HaCohen-Kerner, “Automatic Extraction of Keywords from Abstracts” in Proc. of 7th Int. Conf. KES 2003 (LNCS v. 2773), pp, 843-849, 2003.

[11] N. Pudota, A. Dattolo, A. Baruzzo, C. Tasso, “A New Domain Independent Keyphrase Extraction System” in CCIS 2010, V.91, pp. 67-78, 2010.

[12] D. Mahata, J. Kuriakose, R.R. Shah, R.Zimmermann, “Key2Vec: Automatic Ranked Keyphrase Ectraction from Scientific Articles using Phrase Embeddings” Association for Computational Linguistics, 2018.

[13] Sujatha Das Gollapalli, Cornelia Caragea, “Extracting Keyphrases from Research Papers Using Citation Networks”, AAAI 2014.

[14] http://arxiv.org

[15] https://fasttext.cc/docs/en/pretrained-vectors.html