Automatic Keyphrase Extraction using Graph-based Methods

Josiane Mothe
IRIT, UMR5505, CNRS
Univ. Toulouse, EPSE, France
Toulouse, France
josiane.mothe@irit.fr

Faneva Ramiandrisoa
IRIT, UMR5505, CNRS
Univ. Toulouse
Toulouse, France
r.faneva.mahery@gmail.com

Michael Rasolomanana
Univ. Antananarivo
Antananarivo, Madagascar
tanjonaamikael@yahoo.ca

ABSTRACT

This paper analyses various unsupervised automatic keyphrase extraction methods based on graphs as well as the impact of word embedding. Evaluation is made on three datasets. We show that there is no differences when using word embedding and when not using it.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; *Document representation*; • **Computing methodologies** → Information extraction;

KEYWORDS

Information systems; Information retrieval; Automatic keyphrase extraction; Graph-based methods; Word-embedding; Word2Vec

ACM Reference Format:
Josiane Mothe, Faneva Ramiandrisoa, and Michael Rasolomanana. 2018. Automatic Keyphrase Extraction using Graph-based Methods. In *SAC 2018: SAC 2018: Symposium on Applied Computing , April 9–13, 2018, Pau, France.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3167132.3167392

1 INTRODUCTION

Keyphrase extraction is about "the automatic selection of important and topical phrases from the body of a document" [17]. Keywords or Keyphrases are usually associated to scientific publications in order to help the user having a quick overview of what the paper is about ; they can also be used as a search entry, in information retrieval, natural language processing, and text mining. Keyphrase assignment can be either manual (chosen by authors or librarians) or automatic (like in many search engines).

Automatic keyphrase or keyword extraction aims at extracting automatically a limited number of keyphrases from texts, without using any ontological resource. The result is more to be compared with keywords provided by the authors; for this reason when evaluating automatic extraction methods, authors' keywords are usually considered as ground truth [15].

In the literature, several methods have been proposed in order to extract automatically keyphrases, either supervised or unsupervised.

SAC 2018, April 9–13, 2018, Pau, France

ACM ISBN 978-1-4503-5191-1/18/04.
https://doi.org/10.1145/3167132.3167392

Unsupervised methods for keyword extraction are formulated as a ranking problem. Various unsupervised methods have been proposed in the literature and can be grouped into: *statistical approaches, topic-based clustering groups and graphed-based methods*. The advantage of unsupervised methods over supervised methods is that they do not need a training set; as a consequence they are less sensitive to topic changes and thus more adaptable.

In this paper, we focus on unsupervised methods, more precisely on graph-based methods because they are the most common and they are diverse enough. The principle of this type of method is to construct a graph of words and/or phrases (nodes are the candidate keyphrases and an edge connects two nodes if the candidate keyphrases are related). The edges and their weight can be computed using co-occurrence counts [18],[12], [13] or semantic relatedness [6], etc. Nodes are then ranked using graph properties such as centrality [11].

This paper aims at investigating the use of word embedding representation in keyphrase extraction and its impacts. More precisely, this paper tackles the following research questions: can word embedding be integrated into state of the art graph-based keyphrase extraction models and does word embedding representation improve the results?

We first show how word embedding can be integrated into keyphrase extraction models ; we detail this integration considering graph-based method.

We then evaluate word embedding keyphrase extraction considering several collections and study different parameters.

The paper is organized as follows: Section 2 presents the integration of word embedding in keyphrase extraction. Section 3 presents the evaluation framework, reports the results and discuss them. Finally Section 4 concludes this paper.

2 WORD EMBEDDING IN KEYPHRASE EXTRACTION METHODS

2.1 Word embedding

Word embedding represents words as vectors. It is based on the "Distributional Hypothesis" where words that are used and occur in the same contexts tend to purport similar meanings. Word embedding follows the idea that contextual information constitutes a viable representation of linguistic items. Word embedding methods are generally supervised and use machine learning algorithms to build word representation. They can be categorized in two main types : count-based and Neural Network (NN)-based [1].

These methods differ both in the way they construct the word vectors and the context they consider when building them. Count-based methods use documents as context and capture semantic similarity between documents, and they tend to be used for topic

modelling. On the other hand NN-based approaches use neighbour words as context to detect word-to-word similarity and are effective at catching the similarity between words.

In this work, we investigate the use of NN-based approaches that capture better word-to-word relationships. More precisely we considered Word2Vec[14] which is an implementation of word embedding. In our experiments, we used the pre-trained model on 100 billion words from Google News[1].

2.2 Word embedding in weighted graph-based keyphrase extraction methods

In this section, we explain how to integrate word embedding in graph-based keyphrase extraction methods.More precisely, word embedding is used when weighting graph edges and then graph-based ranking algorithm is applied for node (word) ranking. Graph-based keyphrase methods consist mainly in the following steps:

Preprocessing phase: We used Stanford POS Tagger [2] to compute the part of speech tags; adjectives and nouns only are kept to build graph of words.

Construction of the graph of words: For each document, a non-directed graph is built where nodes are words resulting from the first step; two nodes are connected if they co-occur in a given window of words in the document. We compared three following means of weighting the graph of words in order to measure the impact of word embedding:

(i) *Co-occurrence only:* as in Boudin's work [4], the weight is the number of co-occurrences.

(ii) *Co-occurrence in addition to weight with word embedding*: the weight is obtained by multiplying the number of co-occurrences by the cosine similarity between the word vector representations of the two nodes the edge connects obtained with word embedding. The intuition behind this is to strengthen the semantic link of two words with the number of times they co-occur.

(iii) *Weighting with word embedding*: The edge weight corresponds to the cosine similarity measure between the word vector representation of the two nodes the edge connects. The idea behind this is to fully rely on the word embedding representation to quantify semantic relationships between words.

For RAKE [16], the difference from above is the way co-occurrence is computed. While previously, two nodes are connected if they co-occur in a given window of words in the document, in RAKE, two nodes are connected if they co-occur within candidate keyphrases. This allows RAKE to make abstraction of an arbitrarily sized sliding window.

Node-ranking: once the graph of words is constructed, the words are ranked according to different graph-based ranking algorithm such as centrality-measure.

Candidate keyphrases construction and ranking: candidate keyphrases are sequences of adjacent words in documents restricted to nouns and adjectives only. The score of each candidate keyphrase is obtained by summing the scores of the words it is composed of and then normalize this sum by the number of this words.

Keyphrases selection: The keyphrases are selected as the top ranked candidates keyphrases.

In the evaluation section, we report the results when integrating word embedding in graph-based methods where weights are calculated in three different ways as follows: i) co-occurrence is used to weight edges, ii) when using both co-occurrence and word embedding, iii) when using word embedding only.

3 EVALUATION

3.1 Methods, data collections and measures

We evaluated node ranking as in the following methods: TextRank [13], Hits [10], Eigenvector [3], Closeness [2], Betweenness [5] and degree centrality measure [4].

We used three different data collections: (a) SEMEVAL[3] [9] : consists of academic papers from ACM digital library. We used the 100 papers (title, abstract, a,d full text) of the test data set (b) INSPEC[3] [7] : 2,000 journal paper abstracts (title and abstract) divided into training, test and trial. We used the 2,000 abstracts. and (c) BIOMED : 3,632 publications in Biomedecine and IR we extracted from WoS (title and abstract).

To evaluate keyphrase extraction, as in SemEval framework [8], we compared the keyphrases provided by authors to those extracted by the system using exact match with precision (P), recall (R) and F-measure (F).

3.2 Results using word embedding

Table 1 presents the main results we obtained using precision, recall and F-measure. In this table, we report the results when considering the ranked list of keyphrases and cutting it at 10 for INSPEC data set, at 15 for SEMEVAL data set and at 5 for BIOMED data set. These numbers (10, 15, and 5) correspond to the average number of keyphrases that are provided by authors in the corresponding collections. Average author keyphrases is 9.8 for INSPEC, 14.7 for SEMEVAL, and 5.1 for BIOMED.

For our evaluation we set the window size to 10 words. In fact several results showed that this is the best empirical configuration to get the best performances with the TextRank model. This was first introduced in SingleRank which is a simple variant of TextRank model by setting windows size to 10, while the original TextRank window configuration was set to 2 [19].

We calculated whether the difference using and not using word embedding was statistically significant but it was not.

From Table 1, we can observe that using word embedding (word2Vec pre-trained by Google) has no significant impact on the different keyphrase extraction methods that we evaluated. We consider Student t-test using p-value=0.05. We can see some variations when using word embedding (increases in INSPEC and decreases on SEMEVAL and BIOMED) but these differences are not statistically significant.

In the same table, we can see that RAKE performs better on long documents (SEMEVAL) while the other graph-based methods perform better on short documents (INSPEC and BIOMED).

[1] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/view dataset. The model contains 300-dimensional vectors for 3 million words and phrases.

[2] https://nlp.stanford.edu/software/tagger.shtml

[3] available at https://github.com/boudinfl/centrality_measures_ijcnlp13/tree/master/data

Table 1: Keyphrase extraction performance in terms of Precision (P), Recall (R) and F-measure (F) on the 3 data collections using the various models: Degree (DG), Eigen Vector (EV), Closeness (CL), hits, Betweenness (BE), TextRank (TR), and RAKE (RK) considering (i) co-occurrence only, (ii) co-occurrence + word-embedding, (iii) word-embedding only. Values in bold font are the best on the considered dataset. DG is not sensible to word embedding and reported just of (i).

	Method	INSPEC			SEMEVAL			BIOMED		
		P	R	F	P	R	F	P	R	F
(i)	DG	0.10	0.14	0.12	0.10	0.10	0.10	0.10	0.11	0.10
	EV	0.09	0.13	0.11	0.08	0.09	0.08	0.09	0.11	0.10
	CL	0.10	0.14	0.12	0.05	0.05	0.05	0.08	0.09	0.08
	hits	0.09	0.13	0.11	0.08	0.09	0.08	0.09	0.11	0.10
	BE	0.09	0.13	0.11	0.08	0.08	0.08	0.09	0.10	0.09
	TR	0.10	0.14	0.12	0.09	0.09	0.09	**0.10**	**0.11**	**0.10**
	RK	0.08	0.11	0.09	**0.18**	**0.12**	**0.14**	0.03	0.03	0.03
(ii)	EV	0.09	0.13	0.11	0.07	0.08	0.07	0.08	0.09	0.08
	CL	**0.11**	**0.14**	**0.12**	0.04	0.04	0.04	0.07	0.08	0.07
	hits	0.09	0.13	0.11	0.07	0.08	0.07	0.08	0.09	0.08
	BE	0.09	0.13	0.11	0.09	0.09	0.09	0.08	0.09	0.08
	TR	0.10	0.14	0.12	0.09	0.10	0.09	0.09	0.10	0.09
	RK	0.08	0.11	0.09	0.14	0.14	0.14	0.03	0.03	0.03
(iii)	EV	0.09	0.13	0.11	0.09	0.08	0.09	0.09	0.10	0.09
	CL	0.10	0.14	0.12	0.03	0.03	0.03	0.05	0.06	0.05
	hits	0.09	0.13	0.11	0.09	0.09	0.09	0.09	0.10	0.09
	BE	0.09	0.12	0.10	0.08	0.08	0.08	0.07	0.08	0.07
	TR	0.10	0.14	0.12	0.10	0.10	0.10	0.08	0.09	0.08
	RK	0.07	0.10	0.08	**0.11**	**0.11**	**0.11**	0.02	0.02	0.02

For INSPEC, we reported the results when using the entire data set while some previous studies used the 500 documents from test set only. When considering the 500 documents only, the results are similar to the ones reported in the literature (about 0.39 for Recall, 0.33 for Precision and 0.36 for F-measure for Closeness). It is also the case for other methods that we evaluated in this paper (RAKE, TextRank,...), i.e there is a better performance with 500 documents only.

We also made the cut-off varying, that is to say the value of k used to select the k-top keyphrases in the ranked list in order to evaluate high precision. Apart from TR_Semeval the curves have about the same behaviour whatever the dataset is. The best result is obtained when we consider 15 to 25 top keyphrases for all the cases; less than 10 keyphrases makes lower results (i.e. that the F-measure first increases and then tends to decrease).

4 CONCLUSIONS

In this paper we analysed the impact of word embedding representation, when integrated it into various automatic keyphrase extraction methods. The experiments we conduced show that word embedding representation does not improve significantly the results when compared to the same methods without word embedding, but that it neither decreases the results.

REFERENCES

[1] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors.. In Association for Computational Linguistics (ACL) (1). 238–247.

[2] Alex Bavelas. 1950. Communication patterns in task-oriented groups. Journal of the acoustical society of America (1950).

[3] Phillip Bonacich. 1987. Power and centrality: A family of measures. American journal of sociology (1987), 1170–1182.

[4] Florian Boudin. 2013. A comparison of centrality measures for graph-based keyphrase extraction. In International Joint Conference on Natural Language Processing (IJCNLP). 834–838.

[5] Linton C Freeman. 1977. A set of measures of centrality based on betweenness. Sociometry (1977), 35–41.

[6] Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In Proceedings of the 18th international conference on World wide web. Association for Computing Machinery (ACM), 661–670.

[7] Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 conference on Empirical methods in natural language processing. Association for Computational Linguistics, 216–223.

[8] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In Proceedings of the 5th International Workshop on Semantic Evaluation. Association for Computational Linguistics, 21–26.

[9] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2013. Automatic keyphrase extraction from scientific articles. Language resources and evaluation 47, 3 (2013), 723–742.

[10] Jon M Kleinberg. 1999. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM) 46, 5 (1999), 604–632.

[11] Shibamouli Lahiri, Sagnik Ray Choudhury, and Cornelia Caragea. 2014. Keyword and keyphrase extraction using centrality measures on collocation networks. Preprint arXiv:1401.6571 (2014).

[12] Yutaka Matsuo and Mitsuru Ishizuka. 2004. Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools 13, 01 (2004), 157–169.

[13] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. Association for Computational Linguistics.

[14] T Mikolov and J Dean. 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems (2013).

[15] Faneva Ramiandrisoa and Josiane Mothe. 2016. Extraction automatique de termes-clés : Comparaison de méthodes non supervisées (regular paper). In Rencontres Jeunes Chercheurs en Recherche d'Information (RJCRI). Association Francophone de Recherche d'Information et Applications (ARIA), http://www.irit.fr/ARIA, 315–323. http://www.irit.fr/publis/SIG/2016_RJCS_FM.pdf

[16] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. Text Mining: Applications and Theory (2010), 1–20.

[17] Peter D Turney. 2000. Learning algorithms for keyphrase extraction. Information retrieval 2, 4 (2000), 303–336.

[18] Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge.. In AAAI, Vol. 8. 855–860.

[19] Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In Association for Computational Linguistics (ACL), Vol. 7. 552–559.