Unsupervised Keyphrase Extraction Based on Outlier Detection

Eirini Papagiannopoulou
School of Informatics
Aristotle University of Thessaloniki
epapagia@csd.auth.gr

Grigorios Tsoumakas
School of Informatics
Aristotle University of Thessaloniki
greg@csd.auth.gr

Abstract

We propose a novel unsupervised keyphrase extraction approach based on outlier detection. Our approach starts by training word embeddings on the target document to capture semantic regularities among the words. It then uses the minimum covariance determinant estimator to model the distribution of non-keyphrase word vectors, under the assumption that these vectors come from the same distribution, indicative of their irrelevance to the semantics expresses by the dimensions of the learned vector representation. Candidate keyphrases are based on words that are outliers of this dominant distribution. Empirical results show that our approach outperforms state-of-the-art unsupervised keyphrase extraction methods.

1 Introduction

Keyphrase extraction (KE) aims at finding a small number of phrases that express the main topics of a document. Automated KE is an important task for managing digital corpora, as keyphrases are useful for summarizing and indexing documents, in support of downstream tasks, such as search, categorization and clustering (Hasan and Ng, 2014).

We propose a novel unsupervised KE approach based on outlier detection. Our approach starts by learning vector representations of the words in a document via GloVe (Pennington et al., 2014) trained solely on this document (Papagiannopoulou and Tsoumakas, 2017). The obtained vector representations encode semantic relationships among words and their dimensions correspond typically to topics discussed in the document. The key novel intuition in this work is that we expect non-keyphrase word vectors to come from the same multivariate distribution indicative of their irrelevance to these topics. As the bulk of the words in a document are non-keyphrase we propose using the Minimum Covariance Determinant (MCD) estimator (Rousseeuw, 1984) to model their dominant distribution and consider its outliers as candidate keyphrases.

Figure 1 shows the distribution of the Euclidean distances among non-keyphrase word vectors and between non-keyphrase and keyphrase word vectors for the Nguyen collection of scientific publications (Nguyen and Kan, 2007). We notice that non-keyphrase vectors are closer together than they are with keyphrase vectors, which is in line with our intuition.

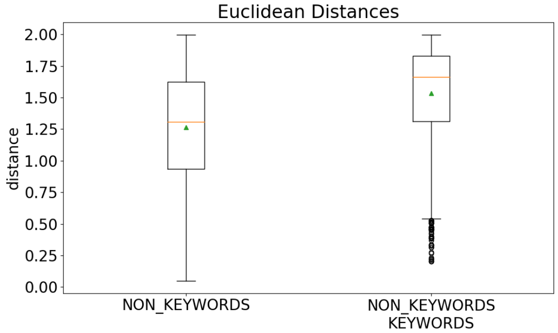

Figure 1: Euclidean distances between non-keywords (1st boxplot) and between non-keywords and keywords (2nd boxplot).

Figure 2 plots 5d GloVe representations of the words in a computer science article from the Krapivin collection (Krapivin et al., 2008) on the first two principal components. The article is entitled "*Excluding any graph as a minor allows a low tree-width 2-coloring*" and is accompanied by the following four golden keyphrases: tree-width, edge partitions, small components, vertex partitions. We notice that keyphrase words are on the far right of the horizontal dimension, while the bulk of the words are on the far left. Similar plots, supportive of our key intuition, are obtained from other documents.

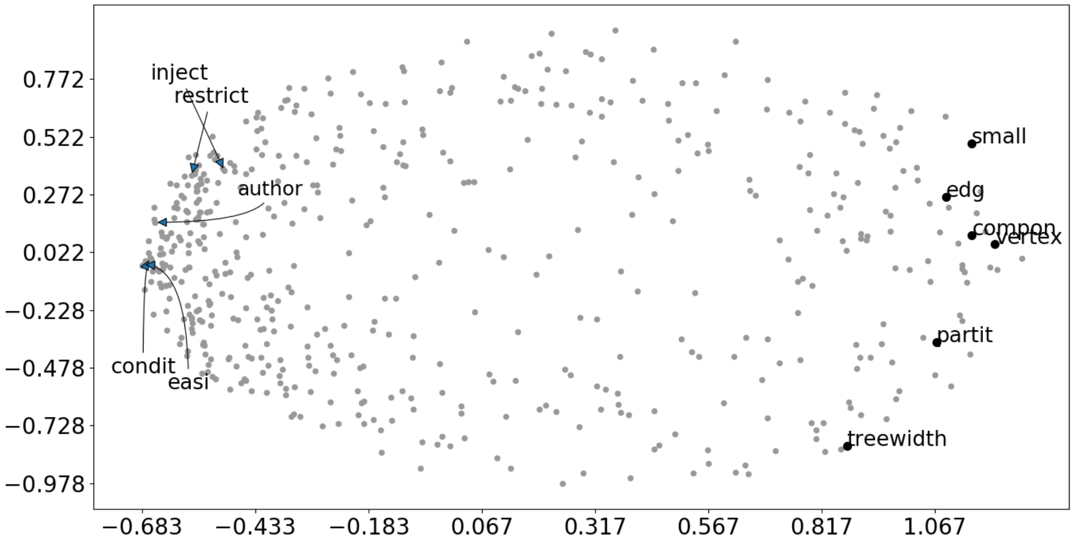

Figure 2: PCA 2d projection of the 5d GloVe vectors in a document. Keyphrases are in black color, while the rest of the words in gray.

2 Related Work

Most KE methods have two basic stages: a) the selection of candidate words or phrases, and b) the ranking of these candidates. As far as the first one is concerned, most techniques detect the candidate lexical units or phrases based on grammar rules and syntax patterns (Hasan and Ng, 2014). For the second stage, supervised and unsupervised learning algorithms are employed to rank the candidates. Supervised methods can perform better than unsupervised ones, but demand significant annotation effort. For this reason, unsupervised methods have received more attention from the community. In the rest of the section, we briefly refer to the basic families of unsupervised KE methods.

TextRank (Mihalcea and Tarau, 2004) builds an undirected and unweighted graph of the nouns or adjectives in a document and connects those that co-occur within a window of W words. Then, the PageRank algorithm (Brin and Page, 1998) runs until it converges and nodes are sorted by decreasing order. Finally, the top-ranked nodes form the final keyphrases. Extensions to TextRank are *SingleRank* (Wan and Xiao, 2008) which adds a weight to every edge equal to the number of co-occurrences of the corresponding words, and *ExpandRank* (Wan and Xiao, 2008) which adds as nodes to the graph the words of the k-nearest neighboring documents of the target document. Additional variations of TextRank are *PositionRank* (Florescu and Caragea, 2017b) that uses a biased PageRank that considers word's positions in the text, and *CiteTextRank* (Gollapalli and Caragea, 2014) that builds a weighted graph considering information from citation contexts. Moreover, Wang et al. (2014, 2015) propose similar graph-based ranking models that take into account information from *pretrained* word embeddings.

Topic-based clustering methods such as *KeyCluster* (Liu et al., 2009), *Topical PageRank (TPR)* (Liu et al., 2010), and *TopicRank* (Bougouin et al., 2013) aim at extracting keyphrases that cover all the main topics of a document utilizing only nouns and adjectives and forming noun phrases that follow specific patterns. KeyCluster groups candidate words using Wikipedia and text statistics, while TPR utilizes Latent Dirichlet Allocation (Blei et al., 2003) and runs a PageRank for every topic changing the PageRank function by taking into account the word topic distributions. Finally, TopicRank creates clusters of candidates using hierarchical agglomerative clustering. It then builds a graph of topics with weighted edges that consider phrases' offset positions in the text and runs PageRank. A quite similar approach to TopicRank has been recently proposed by Boudin (2018). Specifically, the incoming edge weights of the nodes are adjusted promoting candidates that appear at the beginning of the document.

Finally, we should mention the strong baseline approach of *TfIdf* (Jones, 1972) that scores the candidate n-grams of a document with respect to their frequency inside the document, multiplied by the inverse of their frequency in a corpus.

3 Our Approach

Our approach, called *deviant vectors* (DV), comprises four steps that are detailed in the following subsections.

3.1 Learning Vector Representations

We first remove from the given document all punctuation marks, stopwords and tokens consisting only of digits. Then we apply stemming. Subsequently we train the GloVe algorithm solely on the resulting document, following the paradigm of Papagiannopoulou and Tsoumakas (2017). As training takes place on a single document, we recommend learning a small number of dimensions to avoid overfitting.

The GloVe model learns vector representations of words such that the dot product of two vectors equals the logarithm of the probability of co-occurrence of the corresponding words (Pennington et al., 2014). At the same time, the statistics of word-word co-occurrence in a text is also the primary source of information for graph-based unsupervised KE methods. In this sense, the employed *local* training of GloVe on a single document and the graph-based family of methods can be considered as two alternative views of the same information source.

3.2 Filtering Non-Keyphrase Words

The obtained vector representations encode semantic regularities among the document's words. Their dimensions typically correspond to topics discussed in the document. We hypothesize that the vectors of non-keyphrase words can be modeled with a multivariate distribution indicative of their irrelevance to the document's topics.

We employ the fast algorithm of Rousseeuw and van Driessen (1999) for the MCD estimator (Rousseeuw, 1984) in order to model the dominant distribution of non-keyphrase words. Given a data set D, MCD estimates the center, $\bar{x}^*_J$, and the covariance, S^*_J, of a subsample $J \subset D$ of size h that minimizes the determinant of the covariance matrix associated to the subsample:

$$(\bar{x}^*_J, S^*_J) : \det S^*_J \leq \det S_K, \forall K \subset D, |K| = h$$

We expect that the bulk of the words will be non-keyphrase words. At the same time, we expect a number of words, besides the ones involved in keyphrases, to be related to keyphrases words, and therefore to be outliers with respect to the dominant non-keyphrase distribution. In addition, this step of our approach is used for filtering non-keyphrase words and we are therefore interested in achieving high, if not total, recall of keyphrase words. For the above reasons, we recommend using a quite high (loose) value for the proportion of outliers, which can still filter at least half of the non-keyphrases.

3.3 Generating Candidate Keyphrases

We start with the words whose vectors are outliers of the distribution of non-keyphrase words that was modeled with the MCD estimator. We remove any words with length less than 3. We then rank them by increasing position of first occurrence in the document and consider the top 100 as candidate unigrams, in line with the recent research finding that keyphrases tend to appear closer to the beginning of a document (Florescu and Caragea, 2017a).

We adopt the paradigm of other keyphrase extraction approaches that extract phrases up to 3 words (Hulth, 2003; Medelyan et al., 2009), as these are indeed the most frequent lengths of keyphrases that characterize documents. Candidate bigrams and trigrams are constructed by considering candidate unigrams that appear consecutively in the document.

3.4 Scoring Candidate Keyphrases

As a scoring function for candidate unigrams, bigrams, and trigrams we use the TfIdf score of the corresponding n-gram. However, we prioritize to bigrams and trigrams by doubling their TfIdf score, since such phrases are more descriptive and accompany documents more frequently than unigrams (Rousseau and Vazirgiannis, 2015).

4 Empirical Study

4.1 Data Sets and Experimental Setup

Our empirical study uses 3 popular collections of scientific publications: a) Krapivin (Krapivin et al., 2008), b) Semeval (Kim et al., 2010) and c) Nguyen (Nguyen and Kan, 2007), containing 2304, 244 and 211 articles respectively, along with author- and/or reader-assigned keyphrases.

We used the implementation of GloVe from Stanford's NLP group[1], initialized with default parameters (x_{max} = 100, $\alpha = \frac{3}{4}$, *window size* = 10), as set in the experiments of Pennington et al.

[1] https://github.com/stanfordnlp/GloVe

(2014). We produce 5-dimensional vectors with 100 iterations. Vectors of higher dimensionality led to worse results.

We used the NLTK[2] Python suite for preprocessing. Moreover, we used the EllipticEnvelope class from the scikit-learn[3] (Pedregosa et al., 2011) Python library for the MCD estimator, and the PKE toolkit (Boudin, 2016) for the implementations of other unsupervised KE methods.

We follow the strict *exact match* evaluation approach, which computes the F_1-measure between golden keyphrases and candidate keyphrases, after stemming and removal of punctuation marks, such as dashes and hyphens. In particular, we compute F_1@10 and F_1@20, as accuracy at the top of the ranking is more important in typical applications.

We compare DV to the baseline TfIdf method, and four state-of-the-art graph-based approaches: SingleRank (SR), TopicRank (TR), PositionRank (PR), and MultipartiteRank (MR) with their default parameters.

4.2 Results

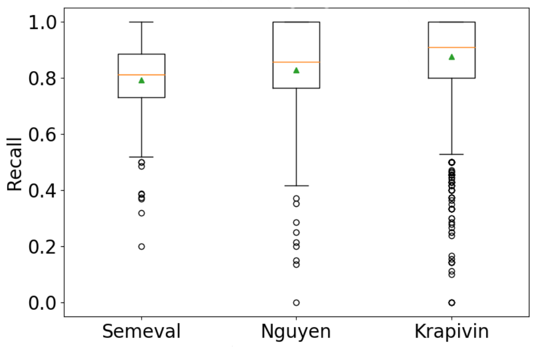

Figure 3: Recall on the 100 candidate unigrams for all documents of Semeval (left), Nguyen (middle), and Krapivin (right).

Figure 3 shows a boxplot of the distribution of the recall achieved by the top 100 candidate unigrams considered in our approach in each document of each collection. We can see that our approach successfully models and filters non-keyphrase words, as a very large percentage of keyphrase words are retained. In particular, half the articles of Semeval, Nguyen and Krapivin achieve recall scores greater than or equal to 0.81, 0.86 and 0.91, while their inter-quartile range lies between 0.73-0.88, 0.76-1.00, and 0.80-1.0 respectively.

	F_1-measure					
	Top-10			Top-20		
Method	Sem.	Ng.	Krap.	Sem.	Ng.	Krap.
SR	0.036	0.043	0.026	0.053	0.063	0.036
TR	0.135	0.126	0.099	0.143	0.118	0.086
PR	0.132	0.146	0.102	0.127	0.128	0.085
MR	0.147	0.147	0.112	0.161	0.149	0.100
TfIdf	0.153	0.199	0.126	0.175	0.204	0.113
DV	**0.194**	**0.233**	**0.169**	**0.204**	**0.219**	**0.143**

Table 1: F_1@10 and F_1@20 of all competing methods on the three data sets.

Table 1 shows that DV outperforms the other methods in all datasets by a large margin, followed by TfIdf (2nd) and MR (3rd). TR and PR follow in positions 4 and 5, alternately for the two smaller datasets, but without large differences between them in Krapivin. SR is the worst-performing method in all datasets.

Based on statistical tests, DV is significantly better than the rest of the methods in all datasets at the 0.05 significance level. In particular, we used either the paired t-test or the Wilcoxon test based on the results of the normality test on the differences of the F_1 scores across the three datasets' articles.

5 Conclusion and Future Work

We proposed a novel unsupervised method for KE, called deviant vectors. Our method learns vector representations of the words in a target document by locally training GloVe on this document and then filters non-keyphrase words using the MCD estimator to model their distribution. The final candidate keyphrases consist of those lexical units whose vectors are outliers of the non-keyphrase distribution and who appear closer to the beginning of the text. Finally, we use TfIdf to rank the candidate keyphrases.

In the next steps of this work in progress, we aim to delve deeper into the local vector representations obtained by our approach and their relationship with keyphrase and non-keyphrase words. In particular, we plan to study issues such as the effect of the vector size and the number of iterations for the convergence of the GloVe model, as well as look into alternative vector representations. In addition, we aim to investigate the effectiveness of the Mahalanobis distance in the scoring/ranking process.

[2] https://www.nltk.org/

[3] https://http://scikit-learn.org

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Florian Boudin. 2016. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of the 26th International Conference on Computational Linguistics, COLING 2016, Proceedings of the Conference System Demonstrations*, pages 69–73, Osaka, Japan.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics Proceedings of NAACL, NAACL 2018*, New Orleans.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the 6th International Joint Conference on Natural Language Processing, IJCNLP 2013*, pages 543–551, Nagoya, Japan.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117.

Corina Florescu and Cornelia Caragea. 2017a. A position-biased pagerank algorithm for keyphrase extraction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4923–4924, San Francisco, California, USA.

Corina Florescu and Cornelia Caragea. 2017b. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1105–1115, Vancouver, Canada.

Sujatha Das Gollapalli and Cornelia Caragea. 2014. Extracting keyphrases from research papers using citation networks. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1629–1635, Québec, Canada.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, (Volume 1: Long Papers)*, pages 1262–1273, Baltimore, MD, USA.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP 2003*, pages 216–223, Stroudsburg, PA, USA.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

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, SemEval@ACL 2010*, pages 21–26, Uppsala, Sweden.

Mikalai Krapivin, Aliaksandr Autayeu, and Maurizio Marchese. 2008. Large dataset for keyphrases extraction. In *Technical Report DISI-09-055*. Trento, Italy.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010*, pages 366–376, Massachussets, USA.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, pages 257–266, Singapore.

Olena Medelyan, Eibe Frank, and Ian H. Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, pages 1318–1327, Singapore.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004*, pages 404–411, Barcelona, Spain.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *Proceedings of the 10th International Conference on Asian Digital Libraries, ICADL 2007*, pages 317–326, Hanoi, Vietnam.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2017. Local word vectors guiding keyphrase extraction. *arXiv preprint*, arXiv:1710.07503. Version 4.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1532–1543, Doha, Qatar.

François Rousseau and Michalis Vazirgiannis. 2015. Main core retention on graph-of-words for single-document keyword extraction. In *Proceedings of the Advances in Information Retrieval - 37th European Conference on IR Research, ECIR 2015*, pages 382–393, Vienna, Austria.

Peter J. Rousseeuw. 1984. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880.

Peter J. Rousseeuw and Katrien van Driessen. 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence, AAAI 2008*, pages 855–860, Chicago, Illinois, USA.

Rui Wang, Wei Liu, and Chris McDonald. 2014. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Software Engineering Research Conference*.

Rui Wang, Wei Liu, and Chris McDonald. 2015. Using word embeddings to enhance keyword identification for scientific publications. In *Proceedings of the Databases Theory and Applications - 26th Australasian Database Conference, ADC 2015*, pages 257–268, Melbourne, VIC, Australia.