Unsupervised Keyphrase Extraction from Scientific Publications

Eirini Papagiannopoulou and Grigorios Tsoumakas

Aristotle University of Thessaloniki
University Campus, 54124, Thessaloniki, Greece
{epapagia,greg}@csd.auth.gr

Abstract. We propose a novel unsupervised keyphrase extraction approach that filters candidate keywords using outlier detection. It 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 expressed by the dimensions of the learned vector representation. Candidate keyphrases only consist of words that are detected as outliers of this dominant distribution. Empirical results show that our approach outperforms state-of-the-art and recent unsupervised keyphrase extraction methods.

Keywords: Unsupervised keyphrase extraction, Outlier detection, MCD estimator

1 Introduction

Keyphrase extraction aims at finding a small number of phrases that express the main topics of a document. Automated keyphrase extraction 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 [11].

We propose a novel approach for unsupervised keyphrase extraction from scientific publications based on outlier detection. Our approach starts by learning vector representations of the words in a document via GloVe [29] trained solely on this document [26]. 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 [31] to model their dominant distribution and consider its outliers as candidate keyphrases.

Figure 1 shows the distribution of the Euclidean distances among vectors of non-keyphrase words, between vectors of non-keyphrase and keyphrase words,

and among vectors of keyphrase words for a subset of 50 scientific publications from the Nguyen collection [25]. We notice that non-keyphrase vectors are closer together (1st boxplot) as well as the keyphrase vectors between each other (3rd boxplot). However, the interesting part of the figure is the 2nd boxplot where the non-keyphrase vectors appear to be more distant from the keyphrase vectors, which is in line with our intuition.

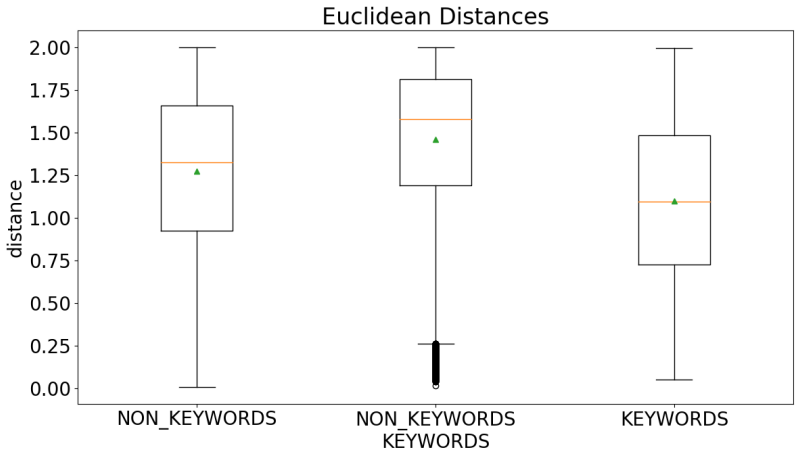


Fig. 1: Distribution of Euclidean distances among non-keywords (1st boxplot), between non-keywords and keywords (2nd boxplot), and among keywords (3rd boxplot).

Figure 2 plots 5d GloVe representations of the words in a computer science article from the Krapivin collection [18] on the first two principal components. The article is entitled "*Parallelizing algorithms for symbolic computation using MAPLE*" and is accompanied by the following two golden keyphrases: logic programming, computer algebra systems. We notice that keyphrase words are on the far left of the horizontal dimension, while the bulk of the words are on the far right. Similar plots, supportive of our key intuition, are obtained from other documents.

The rest of the paper is organized as follows. Section 2 gives a review of the related work in the field of keyphrase extraction as well as a brief overview of multivariate outlier detection methods. Section 3 presents the proposed keyphrase extraction approach. Section 4 describes experimental results highlighting different aspects of our method. We also compare our approach with other state-of-the-art unsupervised keyphrase extraction methods. Finally, Section 5 presents the conclusions and future directions of this work.

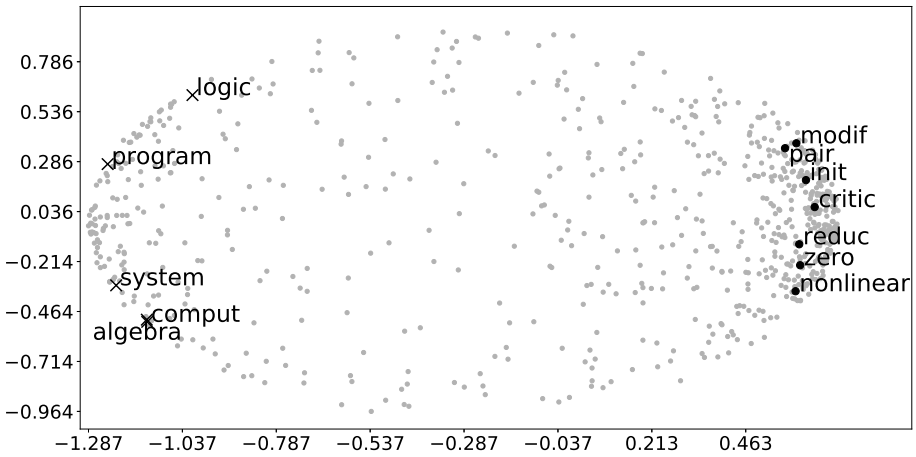


Fig. 2: PCA 2d projection of the 5d GloVe vectors in a document. Keyphrase words are the "x" in black color, while the rest of the words are the gray circle points. Indicatively, we depict a few non-keywords with black circle points.

2 Related Work

In this section, we present the basic unsupervised methodologies (Section 2.1). Then, we briefly review basic multivariate outlier detection methods (Section 2.2).

2.1 Keyphrase Extraction

Most keyphrase extraction 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 [11]. 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 this sub-section, we briefly review the literature on unsupervised keyphrase extraction methods.

TextRank [23] 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 [4] runs until it converges and sorts the nodes by decreasing order. Finally, the top-ranked nodes form the final keyphrases. Extensions to TextRank are *SingleRank* [36] which adds a weight to every edge equal to the number of co-occurrences of the corresponding words, and *ExpandRank* [36] 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* [8] that uses a biased PageRank that considers word's

positions in the text, and *CiteTextRank* [10] that builds a weighted graph considering information from citation contexts. Moreover, in [37,38] two similar graph-based ranking models are proposed that take into account information from *pre-trained* word embeddings. In addition, local word embeddings/semantics, i.e., embeddings trained from the single document under consideration are used by the Reference Vector Algorithm (RVA) [26]. RVA computes the mean vector of the words in the document's title and abstract and, then, candidate keyphrases are extracted from the title and abstract, ranked in terms of their cosine similarity with the mean vector, assuming that the closer to the mean vector is a word vector, the more representative is the corresponding word for the publication.

Topic-based clustering methods such as *KeyCluster* [21], *Topical PageRank (TPR)* [20], and *TopicRank* [3] 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 and runs PageRank for every topic changing the PageRank function so as to take 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, called MultipartiteRank, has been recently proposed in [2]. 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* [16] 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.

2.2 Multivariate Outlier Detection Methods

Outlier detection methods are categorized into three different groups based on the availability of labels in the dataset [9]: *Supervised methods* assume that the dataset is labeled and train a classifier, such as a support vector machine (SVM) [39] or a neural network [5]. However, having a labeled dataset of outliers is rare in practice and such datasets are extremely imbalanced causing difficulties to machine learning algorithms. *One-class classification* [24] assumes that training data consist only of data coming from one class without any outliers. In this case, a model is trained on these data that infers the properties of normal examples. This model can predict which examples are abnormal based on these properties. State-of-the-art algorithms of this category are One-class SVMs [34] and autoencoders [12]. The one-class SVM model calculates the support of a distribution by finding areas in the input space where most of the cases lie. In particular, the data are nonlinearly projected into a feature space and are then separated from the origin by the largest possible margin [6,35]. The main objective is to find a function that is positive (negative) for regions with high (low) density of points. Finally, *unsupervised methods*, which are the most popular ones, score the data

based only on their innate properties. Densities and/or distances are utilized to characterize normal or abnormal cases.

A popular unsupervised method is Elliptical Envelope [13,14,27], which attempts to find an ellipse that contains most of the data. Data outside of the ellipse are considered as outliers. The Elliptical Envelope method uses the Fast Minimum Covariance Determinant (MCD) estimator [32] to calculate the ellipse's size and shape. The MCD estimator is a highly robust estimator of multivariate location and scatter that can capture correlations between features. Particularly, 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:

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

Another popular unsupervised technique is Isolation Forest (IF) [19], which builds a set of decision trees and calculates the length of the path needed to isolate an instance in the tree. The key idea is that isolated instances (outliers) will have shorter paths than *normal* instances. Finally, the scores of the decision trees are averaged and the method returns which instances are inliers/outliers.

In this work, we are interested in detecting the outliers that do not fit the model well (built by the majority of the non-keyphrase words in a text document) or do not belong to the dominant distribution of those words. We expect that the keyphrases and a minority of words that are related to keyphrase words would be the outliers with respect to the dominant non-keyphrase words' distribution or the corresponding model built based on them.

3 Our Approach

Our approach, called *Outlying Vectors Rank* (OVR), comprises four steps that are detailed in the following subsections.

3.1 Learning Vector Representations

Inspired by the graph-based approaches where the vertices added to the graph are restricted with syntactic filters (e.g., selection only nouns and adjectives in order to focus on relations between words of such part-of-speech tags), we remove from the given document all punctuation marks, stopwords and tokens consisting only of digits. In this way, GloVe does not take common/unimportant words into account that are unlikely to be keywords during the model training. Then, we apply stemming to reduce the inflected word forms into root forms. We use stemming instead of lemmatization as there are stemmers for various languages. However, we should investigate the possibility of using lemmas, in the future.

Subsequently, we train the GloVe algorithm solely on the resulting document. As training takes place on a single document, we recommend learning a small

number of dimensions to avoid overfitting. It has been shown in [26] that such local vectors perform better in keyphrase extraction tasks than global vectors from larger collections. 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 [29]. 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 keyphrase extraction 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. Particularly, in previous unsupervised graph-based keyphrase extraction approaches, the limited number of keyphrases (minority) is often assumed to be densely connected to other words in the document and often lies in the center of the word graph. In this vein, the local word vectors capture in a more expressive and alternative way (vector representation) this type of special behavior that the most important words of a document (including the keywords) often present.

3.2 Filtering Non-Keyphrase Words

The obtained vector representation encodes semantic regularities among the document's words. Its dimensions are expected to correspond loosely to the main 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 main topics.

We employ the fast algorithm of [32] for the MCD estimator [31] in order to model the dominant distribution of non-keyphrase words. 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. Then, we apply a second filtering mechanism to the words whose vectors are outliers of the distribution of non-keyphrase words that was modeled with the MCD estimator. Specifically, we remove any words with length less than 3. We then rank them by increasing position of the 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 [7].

Notice that OVR does not have to consider further term frequency thresholds or syntactic information, e.g. part-of-speech filters/patterns, for the candidate keyphrases identification. The properties of the resulting local word vectors capture the essential information based on the flow of speech and presentation of the key-concepts in the article.

3.3 Generating Candidate Keyphrases

We adopt the paradigm of other keyphrase extraction approaches that extract phrases up to 3 words [15,22] from the original text, as these are indeed the most frequent lengths of keyphrases that characterize documents. As valid punctuation

mark for a candidate phrase we consider the hyphen ("-"). Candidate bigrams and trigrams are constructed by considering the candidate unigrams (i.e. the top 100 outliers mentioned earlier) 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 as keyphrases more frequently than unigrams [30].

4 Empirical Study

We first present the setup of our empirical study, including details on the corpora, algorithm implementations, and evaluation frameworks that were used (Section 4.1). Then, we study the performance of our approach based on the proportion of the outlier vectors that is considered (Section 4.2), and we compare the performance of the MCD estimator with other outlier detection methods (Section 4.3). In Section 4.4, we compare OVR with other keyphrase extraction methods and we discuss the results. Finally, we give a qualitative (Section 4.5) evaluation of the proposed approach.

4.1 Experimental Setup

Our empirical study uses 3 popular collections of scientific publications: a) Krapivin [18], b) Semeval [17] and c) Nguyen [25], 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 [29]. We produce 5-dimensional vectors with 100 iterations. Vectors of higher dimensionality led to worse results. We used the NLTK suite[2] for preprocessing. Moreover, we used the EllipticEnvelope, OneClassSVM, and IsolationForest classes from the scikit-learn library[3] [28] for the MCD estimator, One-Class SVM (OC-SVM), and Isolation Forest (IF), respectively, with their default parameters. We utilize the PKE toolkit [1] for the implementations of the other unsupervised keyphrase extraction methods as well as our method. The code for the OVR method will be uploaded to our GitHub repository, in case the paper gets accepted.

We adopt two different evaluation approaches. The first one is the strict *exact match* approach, which computes the F_1-score between golden keyphrases and candidate keyphrases, after stemming and removal of punctuation marks,

[1] `https://github.com/stanfordnlp/GloVe`
[2] `https://www.nltk.org/`
[3] `https://http://scikit-learn.org`

such as dashes and hyphens. However, we also adopt the more loose *word match* approach [30], which calculates the F_1-score between the set of words found in all golden keyphrases and the set of words found in all extracted keyphrases after stemming and removal of punctuation marks. We compute F_1@10 and F_1@20, as the top of the ranking is more important in most applications.

4.2 Evaluation Based on the Proportion of Outlier Vectors

In Tables 1 and 2, we give the F_1@10 and F_1@20 of the OVR method using different proportion of outlier vectors, from 10% up to 49%, on the three data sets according to the exact match as well as the word match evaluation framework, respectively. Generally, we notice that the higher the outliers' percentage the better is the performance of OVR method. Particularly, in almost all cases (except for the F_1@10 of MR based on the word match evaluation), our approach with outlier percentages equal or higher than 30% outperforms the other competitive keyphrase extraction approaches that their performance is presented in Section 4.4 (Tables 5 and 6). We set the proportion of outliers for the rest of our experimental study to 0.49 for the Elliptical Envelope method as well as the other outlier detection methods, whose results are given below (Section 4.3), as with this proportion we achieve the highest F_1-scores.

Table 1: F_1@10 and F_1@20 of the OVR method using different proportion of outlier vectors on the three datasets according to exact match evaluation framework.

	Semeval		Nguyen		Krapivin	
% Outliers	F_1@10	F_1@20	F_1@10	F_1@20	F_1@10	F_1@20
10	0.130	0.124	0.164	0.139	0.113	0.086
20	0.172	0.179	0.204	0.192	0.149	0.122
30	0.184	0.194	0.225	0.209	0.164	0.137
40	0.190	**0.200**	0.230	0.212	0.169	0.143
49	**0.194**	**0.200**	**0.237**	**0.214**	**0.174**	**0.145**

The loose value for the proportion of the outliers helps us in order to apply an effective filtering approach on the candidate keywords that form the keyphrases. We consider that the weak majority of the vocabulary (51% of inliers) represent a common vocabulary that is used by the author during writing the article, while the strong minority (49% of outliers) represents the keywords and an accompanying vocabulary that goes hand in hand with the discussion and the description of the keywords (the core concepts of the document). Such information is captured through the co-occurrence statistics, which are utilized by GloVe.

4.3 Evaluation Based on the Type of Outlier Detection Method

We have designed 2 additional different versions of the proposed OVR approach using 2 alternative outlier detection techniques, which are described previously

Table 2: F_1@10 and F_1@20 of the OVR method using different proportion of outlier vectors on the three datasets according to word match evaluation framework.

	Semeval		Nguyen		Krapivin	
% Outliers	F_1@10	F_1@20	F_1@10	F_1@20	F_1@10	F_1@20
10	0.262	0.283	0.315	0.308	0.286	0.256
20	0.330	0.379	0.383	0.393	0.342	0.326
30	0.349	0.408	0.414	0.426	0.369	0.353
40	0.358	0.417	0.425	0.435	0.384	0.346
49	**0.364**	**0.424**	**0.433**	**0.438**	**0.390**	**0.375**

in Section 2.2, One-class SVM (OC-SVM) and Isolation Forest (IF). In Tables 3 and 4, we provide the F_1@10 and F_1@20 of the different variants of OVR method according to the exact match as well as the word match evaluation framework. Once more, the results confirm that the MCD estimator successfully captures correlations between the vectors' dimensions.

Table 3: F_1@10 and F_1@20 of the OVR method using various outlier detection techniques on the three data sets according to exact match evaluation framework.

	Semeval		Nguyen		Krapivin	
Method	F_1@10	F_1@20	F_1@10	F_1@20	F_1@10	F_1@20
OC-SVM	0.127	0.127	0.141	0.117	0.109	0.086
IF	0.167	0.171	0.192	0.167	0.153	0.126
MCD	**0.194**	**0.200**	**0.237**	**0.214**	**0.174**	**0.145**

Table 4: F_1@10 and F_1@20 of the OVR method using various outlier detection techniques on the three data sets according to word match evaluation framework.

	Semeval		Nguyen		Krapivin	
Method	F_1@10	F_1@20	F_1@10	F_1@20	F_1@10	F_1@20
OC-SVM	0.279	0.309	0.286	0.275	0.268	0.247
IF	0.337	0.380	0.369	0.366	0.351	0.335
MCD	**0.364**	**0.424**	**0.433**	**0.438**	**0.390**	**0.375**

This happens as the classical methods such as OC-SVM can be affected by outliers so strongly that the resulting model cannot finally detect the outlying observations (masking effect) [33]. Moreover, some normal data points may appear as outlying observations. On the other hand, robust statistics, such as the ones that the MCD estimator uses, aim at finding the outliers searching for the

model fitted by the majority of the word vectors. Then, the identification of the outliers is defined with respect to their deviation from that robust fit.

4.4 Comparison with Other Approaches

We compare OVR to the baseline TfIdf method, four state-of-the-art graph-based approaches SingleRank (SR), TopicRank (TR), PositionRank (PR), and MultipartiteRank (MR) with their default parameters, as finally set in the corresponding papers. We also compare our approach to the RVA method which also uses local word embeddings. All methods extract keyphrases from the full-text articles except for RVA which uses the full-text to create the vector representation of the words, but returns keyphrases only from the abstract.

Table 5 shows that OVR outperforms the other methods in all datasets by a large margin, followed by TfIdf (2nd) and MR (3rd), based on the exact match evaluation framework. TR and PR follow in positions 4 and 5, alternately for the two smaller datasets, but without large differences between them in Krapivin. RVA achieves generally lower scores according to the exact match evaluation framework, as it extracts keyprases only from the titles/abstracts, which approximately contain half of the gold keyphrases on average [26]. SR is the worst-performing method in all datasets.

Table 5: F_1@10 and F_1@20 of all competing methods on the three data sets according to exact match evaluation framework.

	Semeval		Nguyen		Krapivin	
Method	F_1@10	F_1@20	F_1@10	F_1@20	F_1@10	F_1@20
SR	0.036	0.053	0.043	0.063	0.026	0.036
TR	0.135	0.143	0.126	0.118	0.099	0.086
PR	0.132	0.127	0.146	0.128	0.102	0.085
MR	0.147	0.161	0.147	0.149	0.116	0.100
TfIdf	0.153	0.175	0.199	0.204	0.126	0.113
RVA	0.094	0.124	0.097	0.114	0.093	0.099
OVR	**0.194**	**0.200**	**0.237**	**0.214**	**0.174**	**0.145**

Moreover, Table 6 confirms the superiority of the proposed method based on the word match evaluation framework. Once more, OVR outperforms the other methods in all datasets by a large margin except for Semeval where MR slightly outperforms OVR.

Based on statistical tests, OVR is significantly better than the rest of the methods in all datasets (besides the MR in Semeval with respect to word match evaluation approach) at the 0.05 significance level. As far as the statistical significance tests concerned, we performed two-sided paired t-test or two-sided Wilcoxon test based on the results of the normality test on the differences of the F_1-scores across the three datasets' articles.

Table 6: F_1@10 and F_1@20 of all competing methods on the three data sets according to word match evaluation framework.

	Semeval		Nguyen		Krapivin	
Method	F_1@10	F_1@20	F_1@10	F_1@20	F_1@10	F_1@20
SR	0.285	0.299	0.322	0.309	0.290	0.256
TR	0.347	0.380	0.376	0.351	0.312	0.277
PR	0.296	0.318	0.371	0.350	0.342	0.302
MR	**0.365**	0.403	0.407	0.383	0.342	0.303
TfIdf	0.308	0.368	0.370	0.394	0.309	0.305
RVA	0.333	0.366	0.374	0.379	0.348	0.337
OVR	0.364	**0.424**	**0.433**	**0.438**	**0.390**	**0.375**

4.5 Qualitative Results

In this section, we use OVR to extract the keyphrases of a publication. This scientific article belongs to the Nguyen data collection. We quote the publication's title and abstract below in order to get a sense of its content:

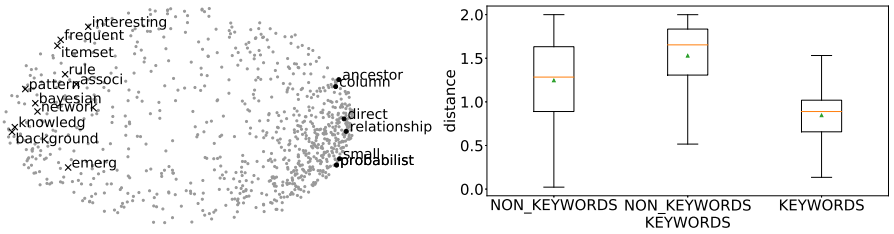
Title: Interestingness of Frequent Itemsets Using Bayesian Networks as Background Knowledge
Abstract: The paper presents a method for pruning frequent itemsets based on background knowledge represented by a Bayesian network. The interestingness of an itemset is defined as the absolute difference between its support estimated from data and from the Bayesian network. Efficient algorithms are presented for finding interestingness of a collection of frequent itemsets, and for finding all attribute sets with a given minimum interestingness. Practical usefulness of the algorithms and their efficiency have been verified experimentally.
Gold Keyphrases: ***association rule, frequent itemset, background knowledge, interestingness, Bayesian network, association rules, emerging pattern***

In Fig. 3, we give the PCA 2d projection of the 5d GloVe vectors of the document as well as the Euclidean distances distribution among non-keywords, between non-keywords and keywords, and among keywords. Moreover, for evaluation purposes, we transform the set of "gold" keyphrases into the following one (after stemming and removal of punctuation marks, such as dashes and hyphens):

{(associ, rule), (frequent, itemset), (background, knowledg), (interesting), (bayesian, network) (emerg, pattern)}

The OVR's result set is given in the first box below, by decreasing ranking score, followed by its stemmed version in the second box. The words that are both in the golden set and in the set of our candidates are highlighted with bold typeface:

{attribute sets, **bayesian network**s, **interestingness**, **itemset**s, **background knowledge**, **bayesian**, attribute, **frequent itemset**s, interesting attribute, interesting attribute sets, **interestingness** measure, interesting **patterns**, **association rules**, data mining, probability distributions, given minimum, minimum **interestingness**, given minimum **interestingness**, minimum support, **knowledge** represented}



(a) (b)

Fig. 3: Figure 3a gives the PCA 2d projection of the 5d GloVe vectors of the document, while Fig. 3b shows the Euclidean distances distribution among non-keywords (1st boxplot), between non-keywords and keywords (2nd boxplot), and among keywords (3rd boxplot)

{(attribut, set), (**bayesian**, **network**), (**interesting**), (**itemset**), (**background, knowledg**), (**bayesian**), (attribut), (**frequent**, **itemset**), (interest, attribut), (interest, attribut, set), (**interesting**, measur), (interest, **pattern**), (**associ**, **rule**), (data, mine), (probabl, distribut), (given, minimum), (minimum, **interesting**), (given, minimum, **interesting**), (minimum, support), (**knowledg**, repres)}

According to the exact match evaluation technique, the top-20 returned candidate keyphrases by OVR include 5 True Positives (TPs), 15 False Positives (FPs) and 1 False Negative (FNs). Hence, precision = 0.25, recall = 0.83 and $F_1 = 0.38$. However, according to the word match evaluation technique, the top-20 returned candidate keyphrases by OVR include 10 TPs, 12 FPs and 1 FNs. Hence, precision = 0.45, recall = 0.91, and $F_1 = 0.60$.

5 Conclusion and Future Work

We proposed a novel unsupervised method for keyphrase extraction from scientific publications, called Outlying Vectors Rank (OVR). 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 appear closer to the beginning of the text. Finally, we use TfIdf to rank the candidate keyphrases.

In the next steps of this work, we aim to delve deeper into the local vector representations obtained by our approach and their relationship with keyphrase and non-keyphrase words. 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.

References

1. Boudin, F.: 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. pp. 69–73. Osaka, Japan (December 11-16 2016), http://aclweb.org/anthology/C/C16/C16-2015.pdf
2. Boudin, F.: Unsupervised keyphrase extraction with multipartite graphs. 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 (June 1-6 2018)
3. Bougouin, A., Boudin, F., Daille, B.: TopicRank: Graph-based topic ranking for keyphrase extraction. In: Proceedings of the 6th International Joint Conference on Natural Language Processing, IJCNLP 2013. pp. 543–551. Nagoya, Japan (October 14-18 2013), http://aclweb.org/anthology/I/I13/I13-1062.pdf
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks **30**(1-7), 107–117 (1998). https://doi.org/10.1016/S0169-7552(98)00110-X, https://doi.org/10.1016/S0169-7552(98)00110-X
5. Das, S.: Elements of artificial neural networks [book reviews]. IEEE Trans. Neural Networks **9**(1), 234–235 (1998). https://doi.org/10.1109/TNN.1998.655048, https://doi.org/10.1109/TNN.1998.655048
6. Dreiseitl, S., Osl, M., Scheibböck, C., Binder, M.: Outlier detection with one-class svms: An application to melanoma prognosis. AMIA Annual Symposium proceedings. AMIA Symposium **2010**, 172–6 (2010), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041295/
7. Florescu, C., Caragea, C.: A position-biased pagerank algorithm for keyphrase extraction. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. pp. 4923–4924. San Francisco, California, USA. (February 4-9 2017), http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14377
8. Florescu, C., Caragea, C.: 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. pp. 1105–1115. Vancouver, Canada (July 30 - August 4 2017). https://doi.org/10.18653/v1/P17-1102, https://doi.org/10.18653/v1/P17-1102
9. Goldstein, M., Uchida, S.: A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one **11**(4), e0152173 (2016)
10. Gollapalli, S.D., Caragea, C.: Extracting keyphrases from research papers using citation networks. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence. pp. 1629–1635. Québec, Canada (July 27 -31 2014), http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8662
11. Hasan, K.S., Ng, V.: 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). pp. 1262–1273. Baltimore, MD, USA (June 22-27 2014), http://aclweb.org/anthology/P/P14/P14-1119.pdf
12. Hawkins, S., He, H., Williams, G.J., Baxter, R.A.: Outlier detection using replicator neural networks. In: Data Warehousing and Knowledge Discovery, 4th International Conference, DaWaK 2002, Aix-en-Provence, France, September 4-6, 2002, Proceedings. pp. 170–180 (2002). https://doi.org/10.1007/3-540-46145-0_17, https://doi.org/10.1007/3-540-46145-0_17

13. Hubert, M., Debruyne, M.: Minimum covariance determinant. Wiley interdisciplinary reviews: Computational statistics **2**(1), 36–43 (2010)
14. Hubert, M., Debruyne, M., Rousseeuw, P.J.: Minimum covariance determinant and extensions. Wiley Interdisciplinary Reviews: Computational Statistics **10**(3), e1421 (2018)
15. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP 2003. pp. 216–223. Stroudsburg, PA, USA (2003). https://doi.org/10.3115/1119355.1119383
16. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation **28**(1), 11–21 (1972). https://doi.org/10.1108/00220410410560573, https://doi.org/10.1108/00220410410560573
17. Kim, S.N., Medelyan, O., Kan, M., Baldwin, T.: Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010. pp. 21–26. Uppsala, Sweden (July 15-16 2010), http://aclweb.org/anthology/S/S10/S10-1004.pdf
18. Krapivin, M., Autayeu, A., Marchese, M.: Large dataset for keyphrases extraction. In: Technical Report DISI-09-055. Trento, Italy (2008)
19. Liu, F.T., Ting, K.M., Zhou, Z.: Isolation forest. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy. pp. 413–422 (2008). https://doi.org/10.1109/ICDM.2008.17, https://doi.org/10.1109/ICDM.2008.17
20. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010. pp. 366–376. Massachussets, USA (October 9-11 2010), http://www.aclweb.org/anthology/D10-1036
21. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009. pp. 257–266. Singapore (August 6-7 2009), http://www.aclweb.org/anthology/D09-1027
22. Medelyan, O., Frank, E., Witten, I.H.: Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009. pp. 1318–1327. Singapore (August 6-7 2009), http://www.aclweb.org/anthology/D09-1137
23. Mihalcea, R., Tarau, P.: TextRank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004. pp. 404–411. Barcelona, Spain (July 25-26 2004), http://www.aclweb.org/anthology/W04-3252
24. Moya, M.M., Hush, D.R.: Network constraints and multi-objective optimization for one-class classification. Neural Networks **9**(3), 463–474 (1996). https://doi.org/10.1016/0893-6080(95)00120-4, https://doi.org/10.1016/0893-6080(95)00120-4
25. Nguyen, T.D., Kan, M.: Keyphrase extraction in scientific publications. In: Proceedings of the Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, 10th International Conference on Asian Digital Libraries, ICADL 2007, Hanoi, Vietnam, December 10-13, 2007. pp. 317–326 (2007)
26. Papagiannopoulou, E., Tsoumakas, G.: Local word vectors guiding keyphrase extraction. Information Processing & Management **54**(6), 888–902 (2018), https://doi.org/10.1016/j.ipm.2018.06.004

27. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
28. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**, 2825–2830 (2011), http://dl.acm.org/citation.cfm?id=2078195
29. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014. pp. 1532–1543. Doha, Qatar (October 25-29 2014), http://aclweb.org/anthology/D/D14/D14-1162.pdf
30. Rousseau, F., Vazirgiannis, M.: 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. pp. 382–393. Vienna, Austria (March 29 - April 2 2015). https://doi.org/10.1007/978-3-319-16354-3_42, https://doi.org/10.1007/978-3-319-16354-3_42
31. Rousseeuw, P.J.: Least median of squares regression. Journal of the American Statistical Association **79**(388), 871–880 (1984). https://doi.org/10.1080/01621459.1984.10477105
32. Rousseeuw, P.J., van Driessen, K.: A fast algorithm for the minimum covariance determinant estimator. Technometrics **41**(3), 212–223 (1999). https://doi.org/10.1080/00401706.1999.10485670, https://doi.org/10.1080/00401706.1999.10485670
33. Rousseeuw, P.J., Hubert, M.: Robust statistics for outlier detection. Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery **1**(1), 73–79 (2011). https://doi.org/10.1002/widm.2, https://doi.org/10.1002/widm.2
34. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation **13**(7), 1443–1471 (2001). https://doi.org/10.1162/089976601750264965, https://doi.org/10.1162/089976601750264965
35. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C.: Support vector method for novelty detection. In: Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]. pp. 582–588 (1999), http://papers.nips.cc/paper/1723-support-vector-method-for-novelty-detection
36. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence, AAAI 2008. pp. 855–860. Chicago, Illinois, USA (July 13-17 2008), http://www.aaai.org/Library/AAAI/2008/aaai08-136.php
37. Wang, R., Liu, W., McDonald, C.: Corpus-independent generic keyphrase extraction using word embedding vectors. In: Software Engineering Research Conference (2014)
38. Wang, R., Liu, W., McDonald, C.: Using word embeddings to enhance keyword identification for scientific publications. In: Proceedings of the Databases Theory and Applications - 26th Australasian Database Conference, ADC 2015. pp. 257–268. Melbourne, VIC, Australia (June 4-7 2015). https://doi.org/10.1007/978-3-319-19548-3_21, https://doi.org/10.1007/978-3-319-19548-3_21

39. Wille, L.T.: Review of "learning kernel classifiers: Theory and algorithms by ralf herbrich." MIT press, cambridge, mass., 2002. ISBN 026208306x, 384 pages; and review of "learning with kernels: Support vector machines, regularization optimization and beyond by bernhard scholkopf and alexander j. smola." IT press, cambridge, mass., 2002, ISBN 0262194759, 644 pages. SIGACT News **35**(3), 13–17 (2004). https://doi.org/10.1145/1027914.1027921, http://doi.acm.org/10.1145/1027914.1027921