Information Processing and Management 56 (2019) 102063

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

A multi-centrality index for graph-based keyword extraction

Didier A. Vega-Oliveros[a,b], Pedro Spoljaric Gomes[c], Evangelos E. Milios[d], Lilian Berton[*,c]

[a] School of Informatics, Computing and Engineering, Indiana University, Bloomington, IN, USA
[b] Department of Computing and Mathematics University of São Paulo, Ribeirão Preto, SP, Brazil
[c] Institute of Science and Technology Federal University of São Paulo, São José dos Campos, SP, Brazil
[d] Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada

ARTICLE INFO

Keywords:
Automatic keyword extraction
Centrality measures
Complex networks
Network science
Text mining
Text networks
Clustering

ABSTRACT

Keyword extraction aims to capture the main topics of a document and is an important step in natural language processing (NLP) applications. The use of different graph centrality measures has been proposed to extract automatic keywords. However, there is no consensus yet on how these measures compare in this task. Here, we present the multi-centrality index (MCI) approach, which aims to find the optimal combination of word rankings according to the selection of centrality measures. We analyze nine centrality measures (Betweenness, Clustering Coefficient, Closeness, Degree, Eccentricity, Eigenvector, K-Core, PageRank, Structural Holes) for identifying keywords in co-occurrence word-graphs representation of documents. We perform experiments on three datasets of documents and demonstrate that all individual centrality methods achieve similar statistical results, while the proposed MCI approach significantly outperforms the individual centralities, three clustering algorithms, and previously reported results in the literature.

1. Introduction

A huge number of text documents are available nowadays on the Internet, for example, e-newspaper, digital library, encyclopedia, blogs, etc. Automated systems to retrieve, summarize and index these documents are very important. Automatic keyword extraction is a process of selecting words that best represent the text content (Hasan & Ng, 2014). It is very useful in organizing the documents without the need of human annotators since assigning keywords manually can be costly and time-consuming. An automatic analysis that extracts information over a huge amount of data has been proposed in different domains (Hyung, Park, & Lee, 2017; Morillo & Álvarez-Bornstein, 2018; Nasar, Jaffry, & Malik, 2018; Raamkumar, Foo, & Pang, 2017; Xu et al., 2018).

Several approaches have been reported addressing the automatic keyword extraction problem (Bharti & Babu, 2017; Hasan & Ng, 2014). The simplest ones compute statistics from the text like term frequency (TF), term frequency-inverse document frequency (TF-IDF) or word co-occurrences. Since these measures are simple, they do not always produce good results (Batziou, Gialampoukidis, Vrochidis, Antoniou, & Kompatsiaris, 2017; Mihalcea & Tarau, 2004). Other approaches employ machine learning that trains a classifier to find keywords. A drawback of machine learning methods is the need for training data and the bias towards the domain on which they are trained. Linguistic approaches that employ lexical analysis (n-Grams, part-of-speech (POS) pattern, WordNet) and syntactic analysis (parsing, noun phrase) are also used.

On the other hand, graph-based methods appear as a different class of keyword extraction approaches. They relax the limiting

* Corresponding author.
E-mail address: lberton@unifesp.br (L. Berton).

https://doi.org/10.1016/j.ipm.2019.102063
Received 1 April 2019; Received in revised form 24 May 2019; Accepted 17 June 2019
Available online 25 June 2019
0306-4573/

term-independence assumption of the traditional vector space models, by constructing a graph-of-words from which the most central vertices indicate keywords. TextRank (Mihalcea & Tarau, 2004), which is based on PageRank (Brin & Page, 1998), is among the best-known methods in this class. Other centrality measures have been applied to identify these central vertices: Degree centrality for summarization (Litvak, Last, Aizenman, Gobits, & Kandel, 2011); Degree, Closeness, Betweenness and Eigenvector (Boudin, 2013); K-Core (Rousseau, Kiagias, & Vazirgiannis, 2015; Tixier, Malliaros, & Vazirgiannis, 2016). However, the question of which centrality measure is the most appropriate and robust for keyword extraction is still open.

In this paper, we propose a new method (multi-centrality index – MCI) for obtaining the best keyword ranking based on network topology information (in a unsupervised way). We also explore different approaches, like feature selection and clustering, for discovering the best keyword set based on the network structure. We first show that the centrality measures are strongly correlated and produce statistically similar results in the co-occurrence graph-of-words. We compare nine centrality measures for keyword extraction employing local, intermediate and global scale measurements (Betweenness, Clustering Coefficient, Closeness, Degree, Eccentricity, Eigenvector, K-Core, PageRank and Structural Holes). The words associated with the top values of these measures are considered keywords. We discuss their representativeness as keywords and the precision and recall achieved on three datasets of short and medium size (Hulth2003, Marujo2012 and Semeval2010).

Our contributions are fivefold: (1) Analysis of nine centrality measures for keyword extraction, Structural Holes is employed for the first time. We compared the centralities' time complexity and properties, considering measures on different scale: local, intermediate and global. (2) Statistical analysis using Pearson's and Spearman's coefficients, which indicates that the evaluated centrality measures are correlated in the graph-of-words; (3) We show that there exists an optimal combination of measures and we propose an approach to select the most representative measures. To the best of our knowledge, this is the first approach that combines multiple centrality measures into a single multi-centrality index (MCI); (4) Comparison of the nine centrality measures and the multi-centrality approach on keyword extraction that indicates we obtain a high precision, recall, and F1-score, with statistically significant difference in the Nemenyi–Friedman test (Demšar, 2006). We employed three datasets with different size and characteristics. (5) We also used three clustering algorithms, K-means, DBSCAN and Expectation Maximization (EM), to identify the keyword group for each entire dataset using the centrality measures as features for the algorithms. The results show that clustering of the keywords is a hard task, with results in some datasets worse than the individual centralities and the proposed MCI approach.

The rest of the paper is organized as follows: Section 2 presents related work on keyword extraction based on graph-of-word and centrality measures. Section 3 presents the research question of this work and the main objectives. Section 4 presents the materials and methods employed in this work, such as the centrality measures explored in the literature and in this paper, the multi-centrality approach, where we show the existence of an optimal combination among measures and to select the most relevant measures we employ different feature selection and dimension reduction methods. Finally, the clustering algorithms employed as other alternative to combine the measures. Section 5 describes the evaluation performed on three datasets (Hulth2003, Marujo2012, Semeval2010) usually employed in the literature, including the analysis of the centrality measures encompassing precision and recall, Pearson's and Spearman's correlations among measures. Finally, Section 6 presents the final remarks.

2. Related work

One of the most popular approaches to the task of unsupervised keyword extraction is TextRank (Mihalcea & Tarau, 2004). In TextRank the vertices are ranked based on the PageRank (Brin & Page, 1998) (PR) algorithm taking edge weights into account. The top best vertices are kept as keywords. PageRank, which is based on the concept of random walks, tends to favor vertices with many important connections. Given the similarity between TextRank and PageRank (both support weighted graphs), in this paper, we opt to focus on PageRank.

PageRank variations have been used for keyword extraction, for example, Weighted PageRank (Tsatsaronis, Varlamis, & Nørvåg, 2010), Biased-PageRank considering prior knowledge (Liu & Sun, 2012), PositionRank (Florescu & Caragea, 2017) that takes into account the positional information of terms in the document to assign weights to the candidate keywords. SingleRank (Wan & Xiao, 2008) is a simple modification of TextRank that considers the edge weights with the number of co-occurrences and no longer extracts keyphrases by assembling ranked words. The method also borrows co-occurrence information from multiple documents. TopicRank (Bougouin, Boudin, & Daille, 2013) represents a document as a complete graph, where vertices represent topics and each topic is a cluster of similar single and multiword expressions. BazzI, Mammass, Zaki, and Ennaji (2017) employed a textRank variation to extract Arabic keyphrases.

Both unweighted and weighted K-Core decomposition of graphs-of-words was also applied retaining the members of the main cores as keywords (Rousseau et al., 2015). Best results were obtained in the weighted case, with small main cores yielding good precision but low recall, and outperforming TextRank.

Taking cohesiveness into account with the core and truss decomposition of a graph-of-words has been hypothesized to improve keyword extraction performance (Tixier et al., 2016). That way, by analogy with the notion of influential spreaders in social networks, they assume that influential words in graphs-of-words will act as representative keywords and propose the use of K-Core and K-Truss.

Some centrality measures (Degree, Closeness, Betweenness and Eigenvector) for graph-based keyphrase extraction have been compared (Boudin, 2013). Through experiments carried out on three standard datasets of different languages and domains, it was demonstrated that simple Degree centrality achieves results comparable to the widely used TextRank algorithm. Recent approaches that combines more parameters to find keywords are proposed by Biswas, Bordoloi, and Shreya (2018) that is based on Node Edge rank centrality with node weight depending on different parameters.

Table 1
Centrality measures employed in other works for keyword extraction: PageRank (PR), K-Core (KC), K-Truss (KT), Degree (DE), Closeness (CL), Betweenness (BE), Eigenvector (EV), Clustering Coefficient (CC), Eccentricity (EC).

	PR	KC	KT	DE	CL	BE	EV	CC	EC
Mihalcea and Tarau (2004)	*V*								
Tsatsaronis et al. (2010)	*V*								
Liu and Sun (2012)	*V*								
Boudin (2013)				*V*	*V*	*V*	*V*		
Beliga et al. (2014)				*V*	*V*	*V*			
Lahiri et al. (2014)	*V*	*V*		*V*	*V*	*V*	*V*	*V*	
Rousseau et al. (2015)		*V*							
Tixier et al. (2016)		*V*	*V*						
Batziou et al. (2017)	*V*	*V*		*V*	*V*	*V*	*V*	*V*	*V*
Florescu and Caragea (2017)	*V*								

Beliga, Mestrovic, and Martincic-Ipsic (2014) proposed the node selectivity (originally proposed by Masucci & Rodgers (2006)) defined as the average weight distribution on the links of the single node. They applied in Croatian texts and compare to In-Degree, Out-Degree, Closeness and Betweenness measures. Since they applied only in Croatian texts it was not possible compare the effectiveness in English benchmarks.

Batziou et al. (2017) review seven graph-based models and compare in two public annotated collections with small size 779 and 183 documents, respectively. The centrality measures in graph of words considered for keyword extraction were Betweenness, Closeness, Degree, Eigenvector, PageRank, Eccentricity, Coreness, Clustering Coefficient and Term-Frequency (TF) scores. Moreover, they proposed a centrality measure, motivated by Mapping Entropy, which considered the largest community of the graph of words to provide a group of words as the most representative ones in the text document. They observed Closeness centrality and Infomap communities perform better than the other measures. However, the authors did not report the recall, F1-score, nor statistical test for a complete analysis.

Lahiri, Choudhury, and Caragea (2014) employed eleven measures (Degree, Strength, Neighbourhood Size, Coreness, Clustering Coefficient, Structural Diversity Index, PageRank, HITS hub and Authority score, Betweenness, Closeness and Eigenvector) and Term-Frequency-Inverse-Document-Frequency (TFIDF) scores for keyword extraction from directed/undirected and weighted word and noun phrase collocation networks and analyze their performance on four benchmark datasets. They presented as the best centralities in keyword extraction the Degree, Strength, PageRank, and Neighborhood size.

Works that employed different centrality measures for keyword extraction are summarized in Table 1. No previous studies provided a systematically analysis comparing the centrality measures, nor a formal comparison using statistical approaches.

3. Research question

A plethora of online information is available for readers on the internet. Methods to summarize such information are each more relevant, such as automatic keyword extraction. Indeed, centrality measures are useful for keyword extraction in an unsupervised strategy. However, no study indicates which is more suitable for this task. Hence, does exist a centrality index approach that significantly outperforms the classification results for automatic keyword extraction? Our objective encompasses mainly three research topics.

- Analyze the centrality measures employed in the area for keyword extraction in terms of efficiency (time-complexity) and performance (precision, recall, and F1-score). We considered nine measures, Structural Holes is first time applied in the area.
- Verify if there exist some similarities or correlations among the centralities, by addressing statistical analysis. We apply Pearson's and Spearman's coefficients among all measures.
- Propose a new method that considers the combination of centralities. We introduce a multi-centrality index (MCI) which uses different properties from feature selection to extract the most relevant measures. As an alternative, we also employed some clustering algorithms to find the keyword group, and prove that this is not a trivial task.

4. Materials and methods

This section describes the adopted centrality measures, the workflow for unsupervised graph-based keyword extraction and the devised method called the multi-centrality index (MCI), which builds a single index by combining the results from multiple centrality indices. Finally, we also present some clustering approaches as alternatives to combine the measures.

4.1. Centrality measures

A graph (or network) G is defined by a set of vertices V and a set of edges E, $G = (V, E)$, where $n = |V|$ and $m = |E|$. The adjacency matrix A is the mathematical representation of the connections in the graph, i.e., $a_{ij} = 1$ if there is a connection between vertices i and j, and 0 otherwise. In the complex network's area, it has been shown that the more important or influential vertices are not

Table 2
Time complexity of the computation of the adopted centrality measures, where n is the number of vertices, m is the number of edges and $\langle g \rangle$ is the average degree of the graph.

Centrality algorithm	Graph scale	Time complexity
Degree (DE)	local	$O(n)$
Clustering Coef. (CC)	intermediate	$O(n^*\langle g \rangle^2)$
Structural Holes (SH)	intermediate	$O(n + n^*\langle g \rangle^2)$
Eigenvector (EV)	global	$O(n^2)$
PageRank (PR)	global	$O(n + m)$
K-Core (KC)	global	$O(n + m)$
Eccentricity (EC)	global	$O(n^*m)$
Closeness (CL)	global	$O(n^*m)$
Betweenness (BE)	global	$O(n^*m)$

necessarily the highest connected individuals (the hubs), but rather those strategically located on the network (i.e., forming dense and cohesive subgraphs with other central vertices) (Vega-Oliveros, Costa, & Rodrigues, 2017b).

Particularly, centrality measures attempt to rank the importance of the vertices according to their capacity of influence, i.e., those individuals that most influence or propagate a signal, information or disease to a large portion of the network in minimal time (Vega-Oliveros, Berton, Vazquez, & Rodrigues, 2017a; Vega-Oliveros, Costa, & Rodrigues, 2017b). In this way, the definitions of centrality focus on different network properties and applications (Das, Samanta, & Pal, 2018; Pastor-Satorras et al., 2015; Vega-Oliveros, Berton, Lopes, & Rodrigues, 2015). For instance, while the degree is based on the number of connections, related to local information flow, the betweenness centrality considers the load over the whole network, and the clustering coefficient captures a regional or intermediate scale connection.

The nine centrality measures adopted in this work, described in Appendix A, are the following: Degree (DE) or vertex connectivity, Betweenness (BE), Eigenvector (EV), PageRank (PR), Closeness (CL), K-Core (KC), Clustering Coefficient (CC), Eccentricity (EC), and Structural Holes (SH). We emphasize that the centrality measure Structural Holes is analyzed here as a novelty for keyword extraction, and it was disregarded in previous works.

We group the measures into three categories: local, intermediate and global, based on the information from the network needed to calculate them. Degree only needs the adjacent neighbors; Clustering Coefficient and Structural Holes need the adjacent neighbors and their neighbors; the remaining measures need information from the whole network to be calculated, such as minimum shortest path or eigenvectors of the adjacency matrix.

We summarize and compare the measures' time complexity in Table 2. The complexities are sorted in decreasing computation time. Some measures are more computationally expensive in comparison with others. The measures related to the average path calculation over the entire graph have the highest computational cost, with the worst case close to $O(n^3)$. The Clustering Coefficient and Structural Holes measures can be calculated considering only a specific vertex and part of the graph. Thus, intermediate scale measures are more suitable than global scale ones for real-world problems, where the size of the graphs is huge and therefore it can be intractable to process the entire graph. Besides, when calculating centrality measures for only a few vertices, the time complexity drops to the size of the neighborhood of the selected vertices, which can be a significant drop. On the other hand, the local and direct measure is the Degree centrality, which also presents the lowest time complexity, with no need to load and make the calculation for the entire graph. Therefore, in terms of performance and results, Degree is the most efficiently computable measure for different scenarios.

4.2. Graph-based keyword extraction

The workflow employed for graph-based keyword extraction is shown in Fig. 1. Given a set of documents, in the pre-processing

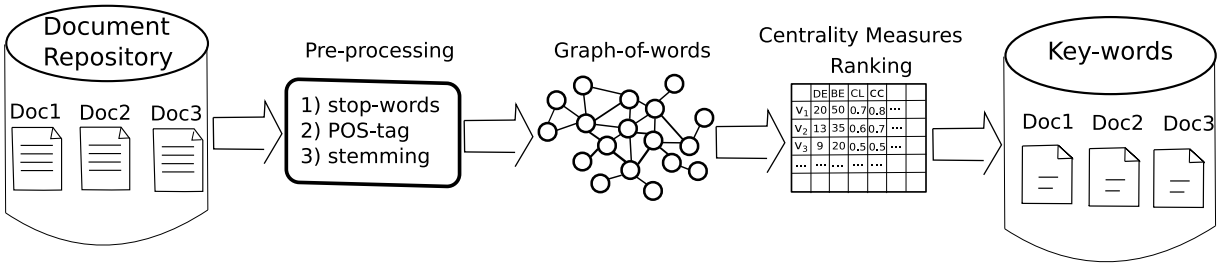


Fig. 1. Workflow employed for graph-based keyword extraction given a document repository or dataset of documents. The first step is to pre-process the text removing stop-words and stemming the remaining POS (verbs, substantives, and adjectives). Next, a co-occurrence graph-of-words is constructed and nine centrality measures are calculated. Finally, for each document, we have nine sets of top-ranked vertices according to the centrality measures, and each set is selected as keywords candidates.

steps we: (1) remove the stop words and punctuation marks; (2) apply POS tagging; and (3) stem using Natural Language Toolkit (NLTK) for Python. We considered words with size length bigger than two characters.

Then, we construct an undirected and unweighted graph-of-words for each document, where vertices are words. Co-occurrence graphs are the most commonly used relation (Batziou et al., 2017; Blanco & Lioma, 2012) where the words selected are lexical units of a POS (nouns, substantives, and adjectives). There is an edge between two vertices if the corresponding terms co-occur within a window of a predetermined size that is slid over the entire document from start to end. Here, we employed a size window equal to 3, since in the work of Mihalcea and Tarau (2004), they tested $w = 2, 3, 5, 10$ and concluded that the larger the window, the lower the precision. This result is explained by the fact that the relation between words that are further apart is not strong enough to the final connection in the text. Moreover, Rousseau et al. (2015) use window of size 4.

The nine centrality measures are calculated from these graphs using the Igraph library. These measures aim to identify the most important vertices in the graph, considering different approaches as explained in Section 4.1. Finally, the words are ranked by each centrality measure and the top words are included in a list of keywords for each document. These keywords are compared to those indicated by humans. Precision, Recall and F1-measure calculations are performed to evaluate the methods.

4.3. Problem definition

Many centrality measures assign higher values to the vertices that have more connections (Degree, PageRank), whereas others consider the distance among vertices (Betweenness, Closeness, Eccentricity), or vertex structures (Clustering Coefficient, K-Core). Furthermore, some measures consider local information (Degree), others intermediate scale information (Clustering Coefficient, Structural Holes) or global information (PageRank, etc).

Notwithstanding previous approaches claiming that a specific centrality measure is more suitable for keyword extraction, most of these network measures seem to be correlated (Schoch, Valente, & Brandes, 2017). Here, we show that the F1-scores of ranked lists obtained by different centrality measures have no statistically significant difference and confirm the high correlation among groups of measures. However, the combination of information from multiple centrality measures into a single index can generate a better descriptor for keyword extraction, as described below, and demonstrated in Section 5.4.

Let $\mathbb{G}_c^n$ be the set of n ranked words for a specific centrality measure c, where $\mathbb{G}_c^n = \{(w_i, v(w_i, c)): v(w_i, c) \in \Re \ \wedge \ i \in [1, n]\}$ is the set of pairs of words w_i and the respective centrality value $v(w_i, c)$. Words w_i are sorted by centrality, i.e., w_1 is the most central word, w_2 is second most central word, and so on. For example, in the case of the Closeness centrality (CL) from the motivational example (Table 4), we have $\mathbb{G}_{CL}^{11}$= { (system, 1.0), (adaptive, 0.78), (decentralized, 0.69), ..., (unknown, 0.36) }, where $c = CL$ and $n = 11$.

Given a set $\mathbb{G}_c^n$, let consider z_c as the number of correct identified keywords, in which $0 \leq z_c \leq n$. For example, in the set $\mathbb{G}_{CL}^{11}$ of the motivational example, we have $z_{CL} = 6$ true-positive keywords (italic words in Table 4). Now, we can define the group of combination between two sets of centrality measures $c1$ and $c2$ in the form $\mathbb{G}_{c1,c2}^n = \mathbb{G}_{c1}^n \cup \mathbb{G}_{c2}^n$. Then, $z_{c1,c2}$ is the total number of correct keywords in the new union set, and the value of $z_{c1,c2}$ is in the interval $[\min(z_{c1}, z_{c2}), \ \dot{n}]$, where $\dot{n}$ is the number of keywords assigned by human annotators and $\min(z_{c1}, z_{c2})$ is the lower value between z_{c1} and z_{c2}.

Definition 1. (*Optimal Subgroup*) An optimal subgroup (H^n) is the set of n words that contains the largest number of true-positive keywords, which were extracted from a group of centrality ranked words. In other words, given the group of combination of centrality ranked words $\mathbb{G}_{c1,c2, \ ..., \ cx}^n = \bigcup \mathbb{G}_{\vec{c}}^n$, $H^n = \{w: w \in \bigcup \mathbb{G}_{\vec{c}}^n\}$, where H^n has the largest number of keywords.

Note that any w word $\in H^n$ must belong to any of the sets of centrality measures. For example in the motivational case study (Table 4), we can find optimal subgroups (H^n) from the combination of Closeness with Degree (DE) or Clustering coefficient (CC), i.e., $\mathbb{G}_{DE,CL}^n$ and $\mathbb{G}_{CC,CL}^n$ respectively. Both groups produce the same optimal subgroup showed in Table 4, with $z = 7$ true-positive keywords.

We also notice that to fulfill the z condition of the largest number of keywords, it is as simple as combining all the centrality sets. However, the real constraint is in obtaining the subset with n words. This particular situation involves a combinatorial problem of selecting the optimal subgroup that reaches the best z result from all the possible sets of words among the centralities. This work presents an endeavor in this direction, where we aim to find a function f_h: $\bigcup \mathbb{G}_{\vec{c}}^n \mapsto H^n$ that selects n words and approximate to the best result of true-positive identified keywords.

4.4. The MCI approach

We seek a significant performance improvement in the keyword extraction task by finding an optimal subgroup among the centrality measures. Two main approaches can be followed: (*1*) the selection of a subset of the most relevant centralities; and (*2*) the dimension reduction to a new set of features from among the centrality measures. These approaches maximize some criterion of the selection, like variance information, correlation, or Laplacian regularization of the features. Nonetheless, several alternatives have been reported in the literature related to Feature Selection (FS), such as subset selection, filters, and wrappers; and linear transformation of the features like PCA, SVD, ICA, and LDA methods (Chandrashekar & Sahin, 2014). Here, the MCI approach employs an unsupervised FS method for finding a f_h function that reveals the optimal H^n group of words.

We explore the following unsupervised methods for FS and dimension reduction:

- The Multi-Cluster Feature Selection (MCFS) (Cai, Zhang, & He, 2010), which is a filter method that essentially reduced the selection to a search problem. It selects those features such that the multi-cluster structure of the data, i.e., the manifold

```
Require: set-Centrality = {DE, BE, CL, PR, EV, KC, EC, SH, CT}
1:  function getMatrixFeatures(repository)
2:      mtxFeatures ← ∅
3:      for all doc ∈ repository do
4:          G-Words ← constructGraph(doc)
5:          mtxDoc ← ∅
6:          for all cnt ∈ set-Centrality do
7:              calcCentrality(G-Words, cnt)
8:              G-Words[cnt]·normalize( )
9:              rank ← G-Words[cnt]
10:             mtxDoc·addColumn(rank)
11:         end for
12:         mtxFeatures·append(mtxDoc)
13:     end for
14:     return mtxFeatures
15: end function
```

Algorithm 1. Matrix of centrality features.

regularization, can be best preserved, based on spectral analysis and L1-regularized least squares regression problem.

- The Principal Feature Analysis (PFA) (Lu, Cohen, Zhou, & Tian, 2007), which is a wrapper method that wraps the search for the best feature subset around a clustering algorithm. The method computes the eigenvalues and empirical orthogonal functions of the covariance matrix of the original features, construct a new subspace dimension of reduced dimension, and cluster using K-Means. Finally, the corresponding original features are the closest to the mean of each cluster.
- The Principal Component Analysis (PCA) is a data reduction method that employs the correlation matrix of the features. It calculates the eigenvalues and principal components (PC) producing a new set of features. Each PC is the linear combination of the original features. We adopted the first principal component (PC1), i.e., the new feature with the highest variance of information, such that the representation is as faithful as possible to the original data.

The input of the FS or dimension reduction methods is the matrix of centrality measures, which includes the features (centrality measures) of each word in each document in the repository. We construct the matrix of features following the steps described in Algorithm 1, given the set of centrality measures (set-Centrality). For each document from the repository or dataset, we construct its corresponding graph-of-words (Algorithm 1, line 4) and calculate all the centrality measures, where each centrality is normalized and added as a column in the document matrix (Algorithm 1, lines 6–11). After that, the document matrices are joined into one single matrix of features (Algorithm 1, line 12) and returned.

The previous part is intended for finding, in an unsupervised form, the best subset of centralities or the more significant projection/transformation from the multi-dimensional space into one component. Algorithm 2 details the MCI approach for the case of combining the best subset of centralities. First, from the subset of selected features (selFeat), we calculate and normalize the centrality measures from the graph-of-word (G-Words), and then, create the $\mathbb{G}_c^n$ groups (Algorithm 2, lines 5–8). In general, the centrality measures rank the same words as centrals, depending on their perspective (local, intermediate, global) of the network, and the nature of the structural measurement (paths, triangles, connections, and others). Moreover, some of the top-ranked words are the same keywords identified by the individual centralities, e.g., Table 4. The previous assumptions are discussed in more detail and demonstrated in Sections 5.3 and 5.4 respectively. Because of that, in lines 11–17 of Algorithm 2, we construct intersection sets between pairs of features in order to prune the words on which there is no consensus and, after that, we expand the set of consensus words by joining the intersection groups, assuming the information interdependency among the features. The algorithm ends returning the selected keywords H^n group (line 19)

In terms of dimension reduction or component transformation methods, we propose the approach described in Algorithm 3. The inputs are the graph-of-words (G-Words), the most relevant component projection (PC1), and the required number of keywords (N). The PC1 comes from employing a linear transformation or projection method over the matrix of centrality measures. The MCI is then the linear combination between the PC1 weights with the respective centrality values of each word (Algorithm 3, line 6). The weights in PC1 represent the relevance of each centrality measure in terms of the information variation concerning the other centralities, and regarding the employed component transformation method. Finally, we sort the words according to MCI importance and return the group of H^n keywords, in line 10. All the codes will be available on the authors' GitHub.

4.5. Clustering approach

In order to analyze other strategies to combine the centrality measures in an abstracting way to find keywords, we also employ unsupervised clustering methods to group the keywords in some entire document. We expect the clustering algorithms generate two groups: keywords and the remaining words respectively. Three clustering algorithms, K-means, DBSCAN and Expectation-

```
function MCI-FS(G-Words, selFeat, N)
    grpWords ← ∅
    keyWords ← ∅
    for all cnt ∈ selFeat do
        calcCentrality(G-Words, cnt)
        G-Words[cnt]·normalize( )
        grpWords[cnt][words] ← G-Words·idName
        grpWords[cnt][value] ← G-Words[cnt]
    end for
    keepTopN(grpWords, N)
    for all cntA ∈ selFeat do
        setA ← grpWords[cntA]
        cntB ← cntA.next
        for cntB ∈ selFeat do
            setB ← grpWords[cntB]
            keyWords ← keyWords ∪ (setA ∩ setB)
        end for
    end for
    return getTopN(keyWords,N)
end function
```

Algorithm 2. MCI approach for feature selection.

```
Require: set-Centrality = {DE, BE, CL, PR, EV, KC, EC, SH, CT}
function MCI-PC1(G-Words, PC1, N)
    MCI ← zeros(G-Words·size)
    for all cnt ∈ set-Centrality do
        calcCentrality(G-Words, cnt)
        G-Words[cnt]·normalize( )
        MCI ← MCI + G-Words[cnt] ∗ PC1[cnt]
    end for
    G-Words·addAtt(MCI)
    keyWords ← sortNodesAtt(G-Words, MCI)
    return getTopN(keyWords, N)
end function
```

Algorithm 3. MCI approach for dimensional reduction.

maximization (EM), were used in the datasets to find the keyword group and compare how to efficiently find the groups. We use classes to cluster evaluation, where if a word is a keyword, its class is 1 and 0 otherwise. For applying cluster in text data, we need to do some text-to-numeric or word-to-vector transformation of our text data. Therefore, we adopted the centrality measures for each word as features for the clustering methods, extracted following Algorithm 1. This work is the first employing this local-topology embedding approach.

K-means clustering (Xu & Tian, 2015) is a method of partitioning data into K subsets, where each data element is assigned to the closest cluster based on the distance of the data element from the center of the cluster. We consider $K = 2$ to separate the data into two subsets: keywords and no-keywords. *Density-based spatial clustering of applications with noise* (DBSCAN) (Xu & Tian, 2015) is a density-based clustering algorithm that groups together points that are close and have many nearby neighbors, and marks as outliers the points that lie alone in low-density regions. We also set the number of clusters equal to two. *Expectation-maximization* (EM) (Xu & Tian, 2015) algorithm alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. We specify to generate two clusters in this case too.

5. Results and discussion

This section presents the employed datasets, a motivational example of the centrality ranking correlation and the evaluation of centrality measures on keyword extraction from graph-of-words. From an undirected and unweighted co-occurrence graph, nine centrality measures rank keywords. Moreover, we presented the two proposed approaches, which selects and combines the three

Table 3
Average of the structural properties among the co-occurrence graphs-of-words: (n) number of vertices, ($\langle g \rangle$) average degree, (Hub) maximum degree of the graph, (Diameter) the largest shortest path, (AvgPath) shortest paths average, (Assortativity) Pearson degree correlation, (Avg. Clust. Coef.) and the average of the clustering coefficient of vertices.

	n	$\langle g \rangle$	Hub	Diameter	AvgPath	Assortativity	Avg. Clust. Coef.
Hulth2003	49.33 ± 20.71	7.21 ± 0.81	21.29 ± 8.39	4.90 ± 1.38	2.40 ± 0.44	−0.08 ± 0.09	0.61 ± 0.04
Marujo2012	156.35 ± 106.19	7.82 ± 0.93	42.04 ± 26.77	6.23 ± 1.68	2.93 ± 0.51	−0.05 ± 0.06	0.57 ± 0.03
Semeval2010	962.12 ± 167.13	16.93 ± 1.63	322.20 ± 67.31	5.68 ± 1.23	2.66 ± 0.11	−0.12 ± 0.03	0.54 ± 0.02

most relevant measures (MCI-FS) and project the features into the first principal component (MCI-PC1). The effectiveness is shown by the precision, recall and statistical analysis of the results.

5.1. Datasets

We used three standards, publicly available datasets featuring documents of various types and sizes. The Hulth2003 (Hulth, 2003) dataset contains abstracts drawn from the Inspec Database of Physics and Engineering papers. We used the 500 documents in the validation set and the uncontrolled keywords assigned by human annotators as the gold standard. The mean document size is 130 words. The training set of Marujo2012 (Marujo, Gershman, Carbonell, Frederking, & Neto, 2012), contains 450 web news stories of about 437 words on average, covering 10 different topics such as culture, business, sport, and technology. For each story, the keyphrases assigned by at least 9 out of 10 Amazon Mechanical Turkers are provided as the gold standard. The Semeval2010 dataset (Kim, Medelyan, Kan, & Baldwin, 2010) offers parsed scientific papers collected from the ACM Digital Library. More precisely, we used the 100 articles in the test set and the corresponding author-and-reader-assigned keyphrases. Each document is approximately 8040 words in length. For all documents, we split the keyphrases assigned by humans into unigrams.

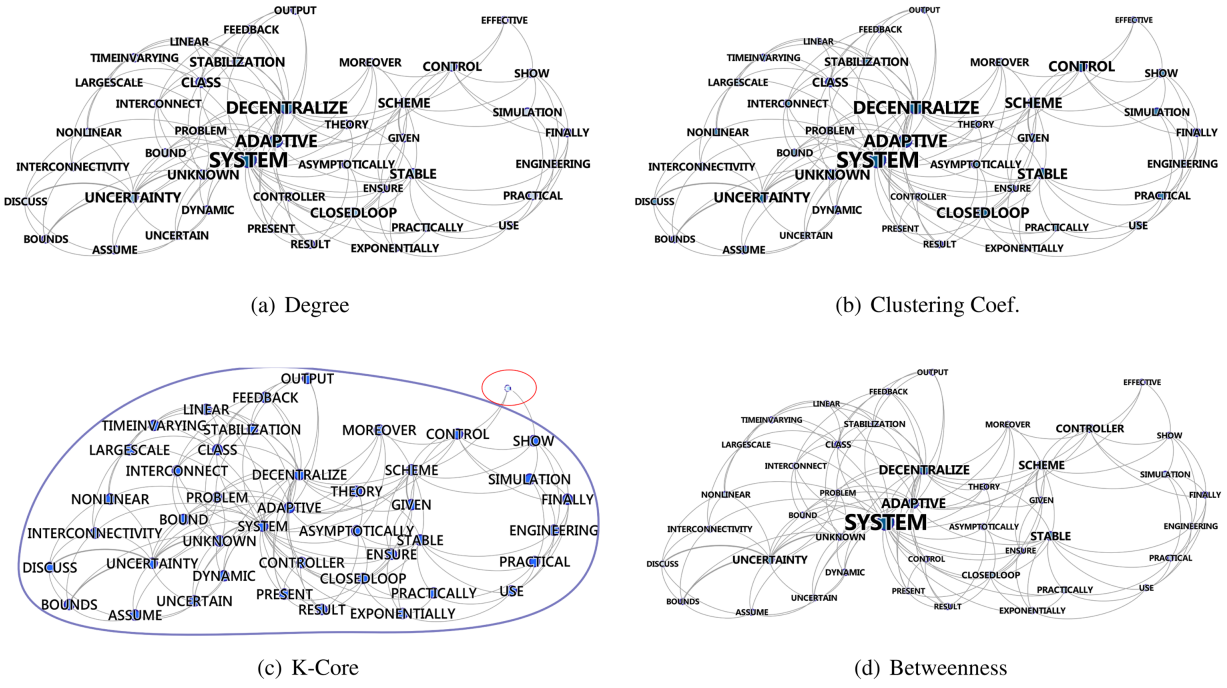
The description of the average properties of the co-occurrence graph-of-words regarding the dataset is shown in Table 3. The following are observations on the three datasets: (1) Hulth2003 generates the smallest graphs, followed by Marujo2012 and Semeval2010; (2) Semeval2010 has a bigger average degree, since it has more words the occurrence of the same words is high; (3) All graphs present hubs, which means some words appear very frequently in the text; (4) Both the diameter and average path of all graphs are similar and achieve small values, while the Clustering Coefficient is high, which characterize them as Small-World graphs (Pastor-Satorras et al., 2015); (5) The co-occurrence graphs-of-words tend to be disassortative, i.e. vertices with small degree connect to vertices with high degree, confirming that some keywords appear in various sentences of the text.

5.2. A motivational case study

As a motivational example, we present a case study analyzing the centrality measures on document 19 from the Hulth2003 dataset (Hulth, 2003). We show that the word rankings of different centrality measures are very similar, but, when combining the centrality measures, we can better identify the keywords. Although the combination of information among centrality measures lead

Table 4
Keywords assigned by human annotators (italic) and the top ranked words by some centrality measures for document 19 from the Hulth2003 dataset. In parentheses are the value of each measure.

	Keywords/Centrality ranking			
Human				
	adaptive	*decentralize*	*stabilization*	*closed-loop*
	system	*uncertain*	*dynamic*	*robust*
	control	*large*	*scale*	
Degree				
	system (26)	*adaptive* (18)	*decentralize* (18)	uncertainty (12)
	class (11)	stable (11)	*closedloop* (11)	*stabilization* (10)
	unknown (10)	scheme (10)	*control* (9)	
Clustering				
	system (0.21)	*decentralize* (0.27)	*adaptive* (0.28)	scheme (0.37)
	closedloop (0.38)	stable (0.38)	*control* (0.388)	uncertainty (0.39)
	unknown (0.4)	class (0.49)	*dynamic* (0.55)	
Closeness				
	system (1.0)	*adaptive* (0.78)	*decentralize* (0.69)	stable (0.51)
	scheme (0.49)	uncertainty (0.47)	*closedloop* (0.45)	class (0.43)
	stabilization (0.38)	*dynamic* (0.37)	unknown (0.36)	
Optimal				
	system	*adaptive*	*decentralize*	*closedloop*
	stabilization	*control*	*dynamic*	class
	uncertainty	scheme	unknown	stable



(a) Degree

(b) Clustering Coef.

(c) K-Core

(d) Betweenness

Fig. 2. Co-occurrence graph-of-words for document 19 of the Hulth2003 dataset (Hulth, 2003). The font size of vertices is scaled by the value of the following centrality measures: (a) vertex Degree, which captures local information from the network. Notice that hubs work well as keywords; (b) inverse Clustering Coefficient which captures intermediate scale information. It considers the information of its neighbors counting the number of the triangles for each vertex normalized by the degree. Lower values of Clustering Coefficient work well as keywords, because of the normalization effect; (c) K-Core which captures global information from the network. In this case, the network forms two cores, one with a value equal to 5 including almost all words and another core with a value equal to 3 including only one word. (d) Betweenness which captures global information but is related to the counting of shortest paths (distance). Vertices with high Betweenness values also work well as keywords.

to believe the achievement of better results, this is not a trivial task due to the correlation and similar accuracy among the measures (as discussed in Sections 5.3 and 5.4 respectively).

The unigram keywords assigned by human annotators in document 19 and the top 11 words with high centrality values are shown in Table 4. We observe in this case study that different measures capture relevant words that can be viewed as keywords. Words similar to those assigned by human annotators are in italics, which in most of the centralities are about six words. The Degree centrality captures local information from the network, i.e., the connections of each vertex. We can observe that Degree works well in identifying six of the golden keywords. The Clustering Coefficient centrality, which captures intermediate scale information, considers the neighbors' information by counting the number of the triangles for each vertex normalized by its degree (for more details of the formulation, please see Appendix A). Due to the degree normalization, words with lower values of Clustering Coefficient are more suitable as keywords, as shown for instance in Table 4. The Closeness centrality, which captures global information but is related to shortest path distances, selects the closest vertices to the other vertices of the network and it also can work for identifying keywords.

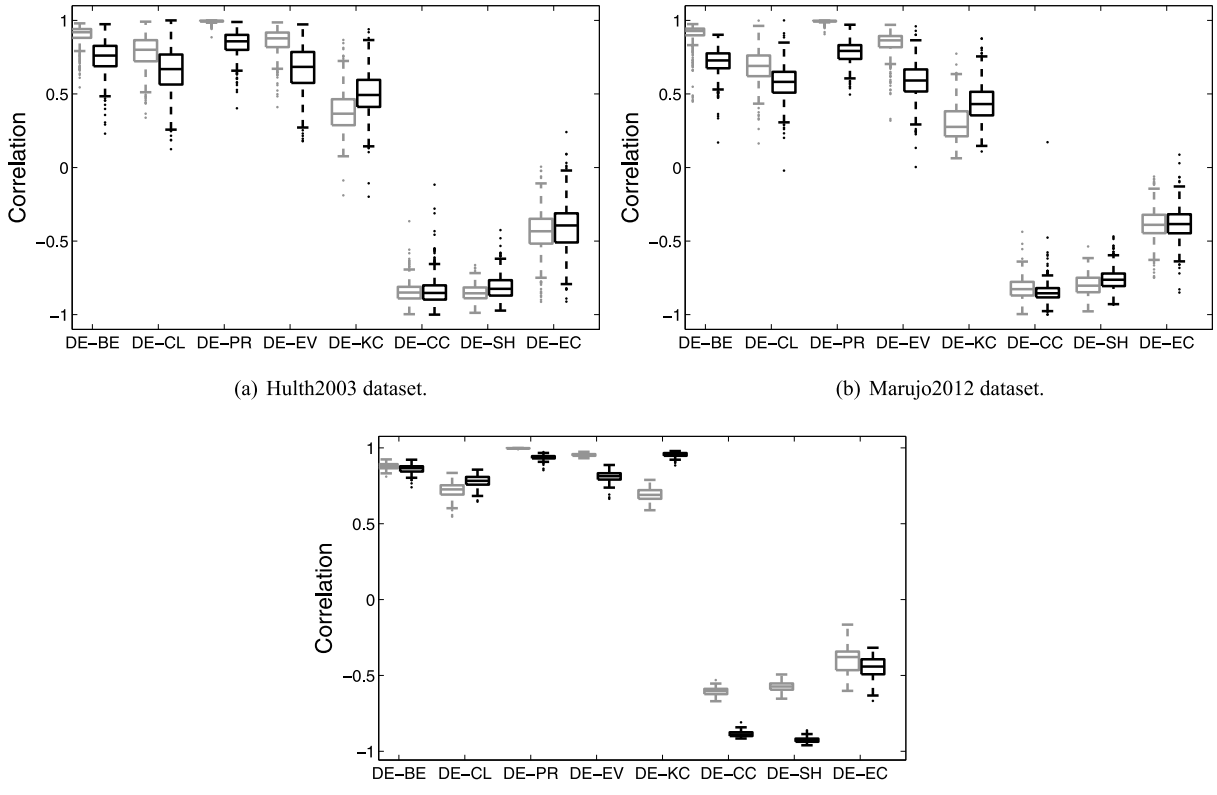
The co-occurrence graphs-of-words for document 19 from the Hulth2003 dataset are depicted in Fig. 2, where four versions of the same graph are shown, with the font size scaled by the centrality measure of a given vertex. In general, all measures can identify highly ranked words as keywords. The K-Core includes a lot of words in its biggest core, this way, it has more chances to match the words of human annotators. However, it is unknown which words are more important inside the same core. Some strategies to address this question were proposed in (Tixier et al., 2016).

However, by combining the ranked lists of multiple measures, we aim to obtain an optimal set with most of the keywords. In this case study, for example, an optimal group would be the union of the ranked lists by all centrality measures, which lead to improving the results by identifying seven golden keywords.

5.3. Correlation analysis among centrality measures

Correlation measures the strength of the statistical relationship between two random variables. Some methods have been proposed and applied in the literature to assess the correlation between two random variables, such as Pearson's, Spearman's and Kendall's. The main differences among these methods are that Pearson's measures the linear correlation between two variables while Spearman's and Kendall's are rank-based methods (Hauke & Kossowski, 2011).

We compute Pearson's and Spearman's correlation coefficients to confirm the correlation among the centrality measures. The full pairwise correlation results are presented in Appendix B. We notice that almost all measures have a high correlation (positive or



(a) Hulth2003 dataset.

(b) Marujo2012 dataset.

(c) Semeval2010 dataset.

Fig. 3. Boxplot distribution, among documents, of pairwise correlation between Degree and the other centrality measures (Betweenness (BE), Closeness (CL), PageRank (PR), Eigenvector (EV), K-Core (KC), Clustering Coefficient (CC), Structural Holes (SH) and Eccentricity (EC)). Pearson's correlation is in gray and Spearman's correlation is in black. The Hulth2003 and Marujo2012 datasets present more variance because they are small graphs.

negative), especially with Degree and the exceptions are K-Core and Eccentricity. In the case of K-Core, the reason is that this measure has many vertices with the same core and does not produce a large ranking. Eccentricity is limited to the diameter of the graph, which is small and presents low variability on the datasets. Eccentricity only has a strong correlation with Closeness centrality, because both measures consider the distance among vertices.

A comparison of Pearson's and Spearman's coefficient for the pairwise correlation of Degree to other measures is shown in Fig. 3. The main observations are: (1) Datasets Hulth2003 and Marujo2012 present more variance than Semeval2010, because they are small graphs, which produce more perturbations in the measurement (Pastor-Satorras et al., 2015), and the sizes of the graphs are highly varied (notice the standard deviations of n in Table 3). Also, these co-occurrence graphs-of-words present more outliers. (2) In general, Spearman's yields lower values than Pearson, except for Degree to K-Core correlation; (3) Eccentricity presents less correlation to Degree (values close to zero); (4) K-Core presents more correlation to Degree in Semeval2010 dataset since the number of core is higher and it produces a more granular ranking.

In a text, the small-world effect means that words share word-neighbors since they are related to the ideas and subjects addressed in the document. This phenomenon is simpler to explain in social networks: if A is a friend of B, it is very likely they have C as a common friend. Moreover, starting from A, few steps are needed to reach any other person. The small-world characteristic can be also related to the neighborhood-inclusion preorder. It means the neighborhood of vertex j includes that of vertex i.

More formally, given the neighborhood N_i of a vertex $i \in V$ is defined as the set of vertices that connect to i, i.e., $N_i = \{j\colon (i, j) \in E\}$, while the closed neighborhood is defined as $N_{[i]} = N_i \cup \{i\}$. If the neighborhood N_i is contained in the closed neighborhood $N_{[j]}$, then the centrality score of node i will always be less or equal to the score of j. Neighborhood-inclusion therefore induces a preorder on the set of vertices which is respected by most centrality-based node rankings (Schoch et al., 2017).

Here, we observe that the role of the underlying network structure is very influential on the similarity of centrality measures. The structural properties are not always apparent in typical network statistics such as density or degree distribution but related to the completeness of the neighborhood-inclusion preorder (Schoch et al., 2017). This indicates the co-occurrence graph-of-word structure impacts on the centrality measures results and the correlation among them.

Table 5
Top five more relevant centrality measures according to the selection methods: Principal Feature Analysis (PFA), Multi-Cluster Feature Selection (MCFS), and Information Gain (IG). The centralities in bold are the common selected features across the methods.

	PFA	MCFS	IG
Hulth2013			
	Degree	**Clustering**	**Degree** (0.065)
	Betweenness	**Structural Holes**	**Clustering** (0.063)
	Eigenvector	Closeness	PageRank (0.063)
	Structural Holes	**Eigenvector**	**Eigenvector** (0.060)
	Clustering	**Degree**	**Structural Holes** (0.058)
Marujo2012			
	Eigenvector	**Eigenvector**	Degree (0.065)
	Degree	**Structural Holes**	Clustering (0.060)
	Closeness	Clustering	PageRank (0.053)
	Eccentricity	Betweenness	**Eigenvector** (0.042)
	Structural Holes	Eccentricity	**Structural Holes** (0.014)
Semeval2010			
	PageRank	**Eigenvector**	**Degree** (0.041)
	Eigenvector	Eccentricity	PageRank (0.040)
	Degree	**Degree**	**Structural Holes** (0.039)
	Structural Holes	Closeness	Clustering (0.038)
	Betweenness	**Structural Holes**	**Eigenvector** (0.038)

5.4. Performance in the keyword extraction task

For the MCI related to the best subset of centrality measures, we calculated MCFS and PFA methods finding the five most significant features in each dataset, as shown in Table 5. Besides, we compared the unsupervised results with the five most relevant features according to Information Gain (IG) method (Quinlan, 1986), which quantifies the relative entropy or high mutual information between the features and the keyword class.

Eigenvector and Structural Holes are the common centralities identified in the five most relevant features by the FS methods (Table 5). Since some centralities have higher computational cost than others, we select Degree, Eigenvector and Structural Holes centrality as the subset of features. The before is because these centralities appear in the top five of the FS methods. In the case of Degree, it has a high correlation to other identified measures like PageRank, Betweenness, and Clustering. Another important point is that Degree has the lowest computational cost among all the measures. Moreover, Degree, Structural Holes, and Eigenvector are local, intermediate and global graph measures, respectively. In this way, we have a comprehensive perspective of the graph-of-word.

After the identification of the best group of features, we use this subset as input to the MCI-FS approach (Algorithm 2), combining the intersection sets of keywords from each pairwise centrality, i.e., Degree, Eigenvector and Structural Holes, into a single set. The time complexity of MCI-FS approach is the cost to calculate these three measures, as shown in Table 2, plus the cost to combine the measures, which is linear.

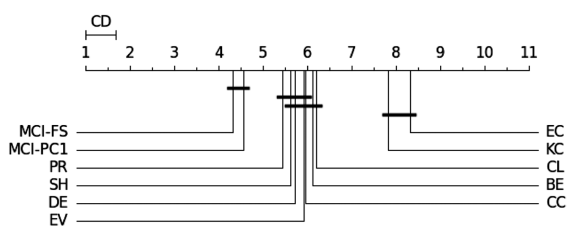
In the case of the dimension reduction approach, we computed the matrix of features of each dataset (Algorithm 1) and used the PCA method obtaining the first principal component (PC1) of the multi-dimensional centrality space. After getting the PC1, we can generalize the MCI-PC1 approach finding the set of potential keywords, for previous or new documents, according to Algorithm 3.

The performance of each centrality measure is evaluated with precision, recall and F1-score for each document and averaged at the dataset level. High precision means the method returned more relevant results than irrelevant ones, while high recall means the method returned most of the relevant results. Candidate and reference keywords are stemmed to reduce the number of mismatches.

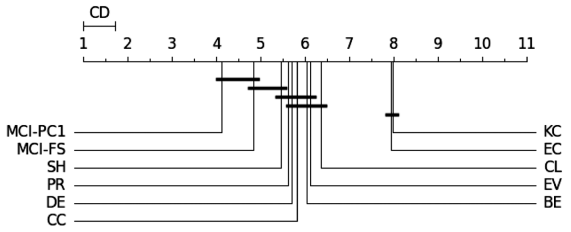
For the experiments, we retained the same number of keywords as human annotators for all datasets. This is a fair comparison

Table 6
Results of average Precision, Recall and F1-score over the datasets for the MCI proposals and the centrality measures.

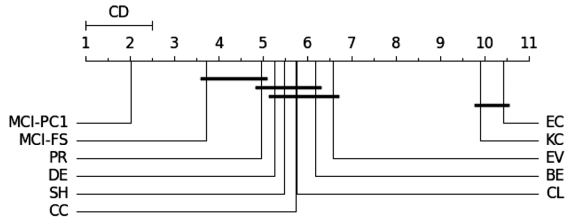
	Hulth2003			Marujo2012			Semeval2010		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Eccentricity	0.371	0.370	0.370	0.467	0.463	0.465	0.270	0.270	0.270
Eigenvector	0.463	0.461	0.462	0.501	0.496	0.498	0.324	0.324	0.324
Clustering Coef.	0.462	0.461	0.461	0.505	0.500	0.502	0.333	0.333	0.333
Betweenness	0.455	0.454	0.454	0.502	0.498	0.500	0.329	0.329	0.329
Degree	0.468	0.467	0.468	0.505	0.496	0.498	0.344	0.344	0.344
Closeness	0.456	0.455	0.456	0.498	0.493	0.495	0.338	0.338	0.338
PageRank	*0.474*	*0.473*	*0.473*	0.507	0.502	0.507	*0.347*	*0.347*	*0.347*
Structural Holes	0.472	0.471	0.471	*0.508*	*0.504*	*0.506*	0.341	0.341	0.341
K-Core	0.379	0.377	0.378	0.463	0.458	0.460	0.291	0.291	0.291
MCI-FS	**0.498**	**0.485**	**0.488**	0.516	0.509	0.511	0.357	0.351	0.354
MCI-PC1	0.493	0.484	0.488	**0.526**	**0.512**	**0.518**	**0.389**	**0.372**	**0.380**



(a) Hulth2003



(b) Marujo2012



(c) Semeval2010

Fig. 4. The critical difference (CD) of the Nemenyi statistical test for comparing the mean-ranking of the centrality methods at 95 percentile, according to the F1-scores of the keyword extraction results: (a) Hulth2003 dataset (b) Marujo2012 dataset and (c) Semeval2010 dataset. Mean-ranking with significance differences are unconnected in the diagrams. Methods in the lowest (best) positions are on the left side.

since we are comparing exactly the same number of words. In Table 6 we present the average performance results of precision, recall, and F1-score for the three datasets, with the centrality measures and the two proposed approaches.

All measures find similar results, where, among the centrality measures, PageRank is the best ranked in Hulth2003 and Semeval2010 datasets, and Structural Holes is the best ranked in Marujo2012. The MCI approaches, which combine the centrality measures, always achieved the best results with statistical significance difference according to the Nemenyi–Friedman test (Demšar, 2006), in comparison to the individual centralities.

In some datasets, keywords were freely chosen by human coders in an abstractive way and as such, some of them are not present in the original text. This makes it impossible to obtain a high recall with extractive methods. Considering the same number of keywords as humans we seem to be able to identify with centrality measures around half of the human-defined keywords.

If we consider a higher number of keywords, as reported in other works, (Mihalcea & Tarau, 2004; Tixier et al., 2016), the recall will increase. For example, document 19 from Hulth2003 dataset generates a co-occurrence graph-of-words with 43 words and the human annotators indicated 11 keywords (Table 4). The top 11 words ranked by Degree contains 6 same/similar keywords to the human annotator. Precision and recall are equal to 0.6 in this case study. Let us suppose we retain the top 13˚ words, which contains 9 same/similar keywords to the human annotator. Precision and recall are now equal to 0.69 and 0.9, respectively.

We perform a statistical analysis of the keyword extraction results to understand the ranking and possible significant differences better. We execute the Nemenyi post-hoc test (Demšar, 2006) grouping by datasets. In Fig. 4 we have the results of the statistical analysis. On the top of the diagrams is the critical difference (CD) and in the axis are plotted the average ranks of the methods, where the lowest (best) positions are on the left side. When a set of methods have no significant difference, they are connected by a black line in the diagram. In all the statistical test, for Friedman and Nemenyi, we considered the statistics at 95 percentile. More details about the statistical test can be found in Appendix C.

According to the Friedman test, the null hypothesis that all methods have similar behavior should be rejected in the three datasets. For Hulth2003 in the Nemenyi diagram (Fig. 4(a)), significant mean-rankings are unconnected. Notice that both MCI approaches have statistical differences compared to other methods. In Marujo2012 for the Nemenyi statistical test (Fig. 4(b)), notice that the proposed MCI-PC1 approach is the best ranked and has statistical differences with all other methods, excepts for MCI-FS. In Semeval2010, concerning the Nemenyi statistics (Fig. 4(c)) the MCI-PC1 is the best ranked with a significant difference compared to all the methods. The MCI-FS approach is the second best ranked and has a statistical difference with many well-known centralities, except for PageRank. However, the mean-ranking difference with PageRank was very close to the critical difference, as shown in the diagram.

Therefore, the statistical results indicate that the MCI approaches are the best ranked compared to the individual centrality methods, and present significance difference with many of the well-known centralities. In particular, MCIs achieve remarkable results in Hulth2003 and Semeval2010 datasets, obtaining statistical difference with all the measures, even the Degree centrality.

5.5. Clustering test

We use three unsupervised algorithms to cluster the keywords of each entire dataset, considering as attributes all the centrality measures. The clustering results are presented in Table 7. The EM algorithm always achieved the highest F1-score. This approach

Table 7
Clustering results: average Precision, Recall and F1-score in the top of the Hulth2003, Marujo2012 and Semeval2010 dataset.

	Precision	Recall	F1-score
Hulth2003			
K-means	0.598	0.42	0.493
DBSCAN	0.588	0.396	0.473
EM	0.492	0.518	0.505
Marujo2012			
K-means	0.468	0.302	0.367
DBSCAN	0.469	0.317	0.378
EM	0.448	0.443	0.445
Semeval2010			
K-means	0.03	0.877	0.057
DBSCAN	0.03	0.86	0.058
EM	0.034	0.849	0.065

finds a bit better result than MCIs methods for the dataset Hulth2003, which is the smallest dataset, however, for Marujo2012 and Semeval2010 MCI approaches perform much better. Semeval2010 is a bigger dataset and the proportion of keyword is smaller compared to no-keyword. The clustering algorithms select a huge number of no-keyword as keywords achieving high false-negative rate and diminishing the precision.

In order to understand the structure of our underlying data and to examine if the centrality measures are good separators for clustering, we present the mean values found for each attribute, i.e., the centrality measures of the words in the full dataset and in the clustering algorithms, respectively, in Table 8. When used with text data, clustering can provide a different way to organize the thousands of words in a document.

In general, Degree, Betweenness and Clustering centralities are good separator of the clusters. Degree values in keyword group are high compared to the no-keyword group. No-keywords groups have lower than half the average Degree of the full dataset, and keywords groups are twice the average Degree of the full dataset for all clustering algorithms. Clustering centrality values are smaller in the keyword group and high in the no-keyword group. Thus, Clustering centrality has the opposite behavior from Degree, due to the observation that the lower the Clustering centrality, the better ranked is the word. Betweenness centrality achieves high values for keyword groups and very low values for no-keyword groups (lower than 10% the average of full dataset). For Closeness, PageRank, Eigenvector, and Structural Holes the difference of the values is small among the groups. Structural Holes also has the same inverse pattern presented by Clustering centrality. K-Core and Eccentricity do not present many differences among the groups.

We can observe that the three methods agree in the overall separations of the groups concerning the centrality measures, with some particular difference in terms of selecting the precise cut or threshold between the groups. The before indicates that it is not a

Table 8
Clustering results over the datasets. The centrality measures of each word were used as features and the average values are presented for each cluster in order to demonstrate if the measures could separate the clusters effectively.

Type		Degree	Betweenness	Closeness	PageRank	Eigenvector	K-Core	Clustering	Structural Holes	Eccentricity
Hulth2003										
Full dataset	(24665.0)	7.4102	44.2467	0.4179	0.0203	0.1187	5.0439	0.6014	0.3194	4.1066
K-means	C_0 (21%)	13.346	150.79	0.5089	0.0331	0.2202	5.5317	0.3643	0.1892	3.4887
	C_1 (79%)	5.8243	15.7836	0.3936	0.0168	0.0916	4.9136	0.6647	0.3541	4.2717
DBSCAN	C_0 (22%)	13.3465	150.7919	0.5089	0.0331	0.2202	5.5317	0.3643	0.1892	3.4887
	C_1 (78%)	5.8243	15.7836	0.3936	0.0168	0.0916	4.9136	0.6647	0.3541	4.2717
EM	C_0 (34%)	10.2765	103.0379	0.4642	0.0295	0.1766	4.9428	0.5136	0.279	3.9002
	C_1 (66%)	5.9071	13.4173	0.3936	0.0154	0.0884	5.0969	0.6474	0.3405	4.2149
Marujo2012										
Full dataset	(70358.0)	8.2413	236.8124	0.3406	0.0064	0.0567	5.506	0.5612	0.2859	5.0863
K-means	C_0 (22%)	16.4449	891.962	0.3973	0.0104	0.1128	5.9367	0.2952	0.1454	4.5253
	C_1 (78%)	5.9341	52.5583	0.3246	0.0053	0.0409	5.3849	0.636	0.3254	5.2441
DBSCAN	C_0 (23%)	16.4449	891.962	0.3973	0.0104	0.1128	5.9367	0.2952	0.1454	4.5253
	C_1 (77%)	5.9341	52.5583	0.3246	0.0053	0.0409	5.3849	0.636	0.3254	5.2441
EM	C_0 (34%)	12.6943	611.1843	0.3773	0.0103	0.0961	5.4901	0.4383	0.2221	4.7595
	C_1 (66%)	5.9957	48.0178	0.3221	0.0044	0.0368	5.514	0.6232	0.318	5.2511
Semeval2010										
Full dataset	(96212.0)	16.9818	827.0139	0.3814	0.001	0.0196	9.1762	0.5381	0.1719	4.4751
K-means	C_0 (39%)	32.9999	2031.1597	0.4195	0.0018	0.0374	13.8177	0.3166	0.0756	4.2223
	C_1 (61%)	6.6057	46.9971	0.3567	0.0005	0.008	6.1695	0.6817	0.2343	4.6388
DBSCAN	C_0 (36%)	32.9999	2031.1597	0.4195	0.0018	0.0374	13.8177	0.3166	0.0756	4.2223
	C_1 (64%)	6.6057	46.9971	0.3567	0.0005	0.008	6.1695	0.6817	0.2343	4.6388
EM	C_0 (34%)	35.9491	2333.8082	0.4239	0.002	0.0408	14.3817	0.3084	0.0721	4.2296
	C_1 (66%)	7.0265	36.145	0.359	0.0005	0.0085	6.444	0.6588	0.2242	4.6039

trivial task the separation of the keywords in the attribute space. However, the clustering algorithms achieved low improvements in the F1-score compared to individual centrality measures. Besides, we show that the combination of measures by clustering techniques, in general, does not perform as well as the proposed MCI approaches, except EM in Hulth2003 dataset.

6. Conclusions

In this paper, we have presented a comparison of nine centrality measures (Betweenness, Clustering Coefficient, Closeness, Degree, Eccentricity, Eigenvector, K-Core, PageRank and Structural Holes) for graph-based keyword extraction. In graph-based text processing, in general, each word corresponds to a vertex in the graph and edges connect words that co-occur in a window of size N. Using three standard datasets of different sizes and domains (Hulth2003, Marujo2012 and Semeval2010), we have demonstrated that the measures achieve similar results (shown by precision, recall and F1-score in Table 6). Moreover, we confirm there is a correlation among all measures by Pearson's and Spearman's coefficient (shown in Figure 3 and all the pairwise comparisons in the Tables of Appendix B). These correlations can be due to the structure of the co-occurrence graph-of-words. We calculate different network measures and confirm that these graphs have small-world characteristics: small average-path and high Clustering Coefficient (see Table 3).

We have demonstrated there exists an optimal subgroup of words, which best combines the ranked words of each centrality measure and produces the most complete set of keywords. This work presents an endeavor in this direction reporting the MCI approach, a combination of centrality measures to obtain better results by applying feature selection or dimension reduction. In the experiments, we employ Degree, Eigenvector and Structural Holes combination since these measures were highly relevant features according to different criteria (covariance, Laplacian regularization, and information gain) and have low correlation. Therefore, we obtain an increase in Precision, Recall and F1-score for all the datasets analyzed. We also proposed to employ the first principal component as a linear combination of the centrality features of the graph-of-words. As results, both combination approaches present a very high significant difference in comparison to individual centrality approaches.

Besides, we used three clustering algorithms, K-means, DBSCAN, and EM, in the datasets to find the keyword group. The features for the clustering algorithms were the centrality measures calculated for each word. This way, we investigated if the measures are good descriptors for separating the clusters between keyword and no-keyword. We demonstrate that some measures are better separators than others, especially Degree, Betweenness and Clustering. However, the use of the measures by the clustering algorithms does not lead to better results compared to the individual measures and the MCI approaches, except EM in Hulth2003 dataset.

We performed a systematical analysis of centrality measures for keyword extraction and we open a new research path on multi-centrality approaches for combining methods to improve the results. As future work, we will investigate the role of network structure, focusing on the pre-processing and graph-construction steps. Moreover, new studies considering other attribute selection methods and approaches for combining the centrality measures can be proposed.

Acknowledgments

The authors' thanks to the Sao Paulo Research Foundation (FAPESP) Grant No.: 2018/01722-3 for the financial support. DAVO acknowledges FAPESP Grant No.: 2018/24260-5, 2016/23698-1, and 2015/50122-0. PSG thanks to FAPESP Grant No.: 2017/18126-1 for the financial support.

Appendix A. Centrality Measures

The centrality measures applied in this work are described as follow:

- **Degree (DE)** or connectivity g_i of vertex i, is related to the number of edges or connections that are going out of (g_i^{out}) or going into (g_i^{in}) vertex i. The *average degree* $\langle g \rangle$ for directed networks is the average of the input or output edges. When the network is undirected, the average degree is equal to $\langle g \rangle = 2m/n$, i.e., the sum of all the edges of the network over the number of vertices. The g_i values can be calculated as follows:

$$g_i = \sum_{i \in n} a_{ij}. \tag{A.1}$$

  Vertices with very high g_i values are called *hubs*, which represent strongly connected instances that impact on the dynamics of the network (Das et al., 2018; Pastor-Satorras et al., 2015; Vega-Oliveros & Berton, 2015).
- **Betweenness centrality (BE)** is related to the capacity of information transmission of vertices. The betweenness centrality of a vertex j is given by the number of shortest paths between all pairs of vertices (i, k) that contain j (Das et al., 2018), where i, j and k are different vertices. Mathematically,

$$B_j = \sum_{i,k \in V, i \neq k} \frac{\sigma_{ik}(j)}{\sigma_{ik}} \tag{A.2}$$

  where σ_{ik} is the total number of different shortest paths between i and k, and $\sigma_{ik}(j)$ is the number of times j appears in those paths.
- **Eigenvector centrality (EV)** considers that vertices with the same degree may have different levels of importance depending on

the importance of their neighbors. The eigenvector associated with the highest eigenvalue of the matrix A describes the importance of the vertices in relation to that of its connections (Das et al., 2018). Formally $A\vec{X} = \alpha\vec{X}$, where $\vec{X} = \{X_1, X_2, ..., X_N\}$ is the eigenvector centrality, α is the highest eigenvalue of A, and $\forall\, x, y \in V,\ A_{xy} = A_{yx} \geq 0$ by the Perron-Frobenius theorem. Each component X_i of X gives the relative centrality score of vertex i in the network. X_i is computed by the iterative procedure:

$$X_i^t = \sum_{y \in V} A_{ij} X_j^{t-1}, \tag{A.3}$$

where t represents the iteration step, and $\vec{X}^0 = \{1, ..., 1\}$ when $t = 0$.

- **PageRank (PR)** is defined through a random walk in the network. It expresses the importance of the vertices as the probability of arriving at certain vertex after a large number of steps. The idea is to simulate the behavior of an user that is surfing on the World Wide Web. The user can follow a link incident on the current page or jump to another page by typing a new URL in the browser with a defined probability. The centrality can be calculated by taking into account the long behavior of Markov chains, i.e. $\vec{\pi}^t = \vec{\pi}^{t-1}\mathbb{G}$, where the elements $\vec{\pi}^t$ are the PageRank values for each vertex in iteration step t of the random walk, and $\mathbb{G}$ is known as the Google matrix (Brin & Page, 1998). The procedure to obtain the measure, under the undirected assumption, is

$$\pi_i^t = \alpha \sum_j \frac{A_{ij}\pi_j^{t-1}}{\tilde{g}_j} + \frac{(1-\alpha)e_i}{n} \tag{A.4}$$

where $\tilde{g}_j$ is the degree of its neighbor j that attenuates the importance of i, and when $t = 0$ all vertices have the same probability, $\vec{\pi}^0 = [1/n, ..., 1/n]_{1\times n}$. We have that $\tilde{g}_j = g_j + \delta(g_j, 0)(1/N)$ is a stochastic adjustment in the case of j being unconnected or into a local cycle, and $\delta(a, b)$ is the Kronecker delta function. The jumps in the navigation are represented by the probability α. Note that $\sum_i \pi_i = 1$ for all steps t. The value of $\alpha = 0.85$ adopted here is the same as defined in the original version of the algorithm (Brin & Page, 1998).

- **Closeness centrality (CL)** is defined based on the lengths of the shortest paths from each vertex to the rest of the network (Das et al., 2018). Formally, the closeness centrality CL_i is the inverse of the average of the shortest paths from i to all the vertices, i.e.,

$$CL_i = \frac{n}{\sum_{j\neq i} \ell_{ij}}, \tag{A.5}$$

where ℓ_{ij} is the length of the shortest path from i to j and n the size of the network. In this way, vertices that are closer to many others should have a higher closeness centrality.

- **K-Core (KC)**: The graph can be decomposed in terms of shells or cores (Das et al., 2018). A core H_k ($H_k \in \mathbb{N}_{>0}$) represents the set of vertices with degree g_i that belongs to that H_k shell. The K-Core centrality is obtained by calculating $KC(i) = H_k$ for all vertices, i.e., the highest H_k core value that each vertex is part. The iterative procedure begins with $H_k = 1$ until all the vertices have been removed (Das et al., 2018):
  1. All vertices with degree lower or equal than H_k are removed.
  2. The remaining vertices are re-evaluated several times, in order to remove those with g_i lower or equal than H_k. Vertices from disconnected components are also removed.
  3. The set of removed vertices are part of the H_k core and thus, the K-Core centrality $KC(i) = H_k$ is assigned. After that, H_k is incremented and the process starts again with step (1).
  4. The process continues until all vertices have been removed from the graph.

  The final set of vertices is the main or most central core of the network, which has the largest K-Core centrality.
- **Clustering coefficient (CC)**, in topology terms, CC measures the presence of triangles (cycles of order three) in the network. The clustering coefficient (Pastor-Satorras et al., 2015) of a vertex i is defined as the number of triangles centered on i over its maximum number of possible connections, i.e.,

$$CC_i = \frac{2t_i}{g_i(g_i - 1)}. \tag{A.6}$$

In the case of $g_i \in \{0, 1\}$, CC is assumed to have a value of zero, and $CC_i = 1$ only if all the neighbors of i are interconnected.

- **Eccentricity (EC)**: The shortest path between two vertices is the shortest sequence of edges that connect them, and the distance is the number of edges contained in the path. In the case that i and j belong to different components, it is assumed that $\ell_{ij} = n$. In this way, the eccentricity value of a vertex i is the longest distance over all the shortest paths to the other vertices (Das et al., 2018), as follow:

$$EC_i = \max_{i\neq j}\{|\ell_{ij}|\}, \tag{A.7}$$

where $|\ell_{ij}|$ is the distance of the shortest path between vertices i and j. This measure evaluates how close is a vertex to its most distant vertex. Lower values of eccentricity indicate that the vertex is more central and close to the others. Therefore, vertices located at the network center have the lowest eccentricity values.

- **Structural Holes (SH)**: Some vertices in the network work such as a bridge of clusters for other vertices, and if they are removed a structural hole will occur. The structural hole vertices act as spanners among communities or groups of vertices without direct connections. These individuals are important to the connectivity of local regions. The algorithm considers all vertices as ego

networks, where connections not related to a specific vertex are not considered (Vega-Oliveros et al., 2015). The higher the fraction of structural holes associated with the vertex, the more central it is. Therefore, vertices with higher degree centrality tend to have low structural holes values, given that its ego networks are larger and more densely interconnected, and this diminishes the fraction of isolated holes.

Appendix B. Correlations among centrality measures

Tables A.9, A.10 and A.11 show the results of the pairwise Pearson's and Spearman's correlation coefficients between centrality measures on the Hulth2003, Marujo2012, and Semeval2010 datasets. The coefficients represent the average of the pairwise centrality correlations over the co-occurrence graphs-of-words of the documents of each dataset.

Table A.9
Pairwise Pearson's / Spearman's coefficients between centrality measures on the Hulth2003 dataset.

	1	2	3	4	5	6	7	8	9
1 Degree									
2 Betweenness	0.90 / 0.75								
3 Closeness	0.79 / 0.66	0.75 / 0.64							
4 Pagerank	0.99 / 0.84	0.91 / 0.74	0.74 / 0.37						
5 Eigenvector	0.86 / 0.67	0.71 / 0.42	0.88 / 0.84	0.80 / 0.29					
6 K-Core	0.37 / 0.48	0.19 / 0.24	0.34 / 0.33	0.32 / 0.29	0.42 / 0.43				
7 Clustering	−0.85 / −0.84	−0.74 / −0.83	−0.66 / −0.52	−0.85 / −0.83	−0.64 / −0.42	−0.39 / −0.34			
8 Structural Holes	−0.85 / −0.81	−0.69 / −0.67	−0.83 / −0.81	−0.82 / −0.50	−0.83 / −0.80	−0.52 / −0.45	0.88 / 0.72		
9 Eccentricity	−0.43 / −0.40	−0.47 / −0.49	−0.71 / −0.71	−0.41 / −0.23	−0.51 / −0.52	−0.16 / −0.15	0.39 / 0.33	0.52 / 0.52	

Table A.10
Pairwise Pearson / Spearman coefficients among the centrality measures for Marujo2012 dataset.

	1	2	3	4	5	6	7	8	9
1 Degree									
2 Betweenness	0.90 / 0.71								
3 Closeness	0.69 / 0.58	0.64 / 0.54							
4 Pagerank	0.99 / 0.78	0.92 / 0.68	0.63 / 0.16						
5 Eigenvector	0.84 / 0.59	0.72 / 0.36	0.85 / 0.89	0.79 / 0.10					
6 K-Core	0.30 / 0.43	0.17 / 0.23	0.30 / 0.27	0.26 / 0.22	0.33 / 0.33				
7 Clustering	−0.82 / −0.85	−0.68 / −0.77	−0.59 / −0.40	−0.82 / −0.79	−0.62 / −0.34	−0.38 / −0.34			
8 Structural Holes	−0.80 / −0.76	−0.63 / −0.57	−0.80 / −0.80	−0.76 / −0.29	−0.79 / −0.80	−0.47 / −0.40	0.87 / 0.62		
9 Eccentricity	−0.39 / −0.38	−0.40 / −0.44	−0.72 / −0.71	−0.35 / −0.12	−0.50 / −0.57	−0.15 / −0.15	0.36 / 0.28	0.52 / 0.53	

Table A.11
Pairwise Pearson / Spearman coefficients among the centrality measures for Semeval2010 dataset.

	1	2	3	4	5	6	7	8	9
1 Degree									
2 Betweenness	0.88 / 0.86								
3 Closeness	0.72 / 0.78	0.48 / 0.61							
4 Pagerank	1.00 / 0.94	0.90 / 0.93	0.69 / 0.63						
5 Eigenvector	0.95 / 0.81	0.74 / 0.59	0.84 / 0.97	0.93 / 0.63					
6 K-Core	0.69 / 0.96	0.37 / 0.77	0.83 / 0.80	0.66 / 0.86	0.81 / 0.84				
7 Clustering	−0.61 / −0.89	−0.36 / −0.91	−0.60 / −0.57	−0.61 / −0.92	−0.60 / −0.58	−0.79 / −0.81			
8 Structural Holes	−0.57 / −0.93	−0.30 / −0.71	−0.84 / −0.86	−0.55 / −0.77	−0.68 / −0.91	−0.88 / −0.92	0.79 / 0.77		
9 Eccentricity	−0.41 / −0.45	−0.28 / −0.39	−0.60 / −0.57	−0.39 / −0.36	−0.47 / −0.53	−0.46 / −0.44	0.36 / 0.35	0.50 / 0.50	

Appendix C. Additional information of keyword extraction statistical test

According to the Friedman test for Hulth2003 the critical value of the F-statistics with 9 and 4491 degrees of freedom and at 95 percentile is 1.88. In the Friedman test using the F-statistics, the null hypothesis that all algorithms have comparable behavior should be rejected. According to the Nemenyi statistics, the critical value for comparing the mean-ranking of two different algorithms at 95 percentile is 0.61. Mean-rankings differences above this value are significant.

In Marujo2012, the critical value of the F-statistics with 9 and 3924 degrees of freedom and at 95 percentile is 1.88. According to the Friedman test using the F-statistics, the null hypothesis that all algorithms behave similarly should be rejected. For the Nemenyi

statistical test, the critical value for comparing the mean-ranking of two different algorithms at 95 percentile is 0.64.

In Semeval2010, the critical value of the F-statistics with 9 and 891 degrees of freedom and at 95 percentile is 1.89. According to the Friedman test using the F-statistics, the null hypothesis that all algorithms have similar behavior should be rejected. Concerning the Nemenyi statistics, the critical value for comparing the mean-ranking of two different algorithms at 95 percentile is 1.35. The critical difference between the MCI method and the PageRank was very close to the critical difference.

References

Batziou, E., Gialampoukidis, I., Vrochidis, S., Antoniou, I., & Kompatsiaris, I. (2017). Unsupervised keyword extraction using the gow model and centrality scores. In I. Kompatsiaris, J. Cave, A. Satsiou, G. Carle, A. Passani, E. Kontopoulos, ... D. McMillan (Eds.). *Internet science* (pp. 344–351). Cham: Springer International Publishing.

BazzI, M. S. E., Mammass, D., Zaki, T., & Ennaji, A. (2017). A graph-based ranking model for automatic keyphrases extraction from arabic documents. In P. Perner (Ed.). *Advances in data mining. applications and theoretical aspects* (pp. 313–322). .

Beliga, S., Mestrovic, A., & Martincic-Ipsic, S. (2014). *Toward selectivity-based keyword extraction for croatian news. Proceedings of the workshop on surfacing the deep and the social web co-located with the 13th international semantic web conference (ISWC*.

Bharti, S. K., & Babu, K. S. (2017). Automatic keyword extraction for text summarization: A survey. *CoRR*. abs/1704.03242.

Biswas, S. K., Bordoloi, M., & Shreya, J. (2018). A graph based keyword extraction model using collective node weight. *Expert Systems with Applications, 97*, 51–59.

Blanco, R., & Lioma, C. (2012). Graph-based term weighting for information retrieval. *Information Retrieval, 15*, 54–92.

Boudin, F. (2013). *A comparison of centrality measures for graph-based keyphrase extraction. Proceedings of the sixth international joint conference on natural language processing - IJCNLP*834–838.

Bougouin, A., Boudin, F., & Daille, B. (2013). *Topicrank: Graph-based topic ranking for keyphrase extraction. Proceedings of the international joint conference on natural language processing - IJCNLP*543–551.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, V*, 107–117.

Cai, D., Zhang, C., & He, X. (2010). *Unsupervised feature selection for multi-cluster data. Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining - KDD*333–342.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering, 40*, 16–28.

Das, K., Samanta, S., & Pal, M. (2018). Study on centrality measures in social networks: A survey. *Social Network Analysis and Mining, 8*, 13.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *JMLR, 7*, 1–30.

Florescu, C., & Caragea, C. (2017). *A position-biased pagerank algorithm for keyphrase extraction. Association for the advancement of artificial intelligence – AAAI*4923–4924.

Hasan, K. S., & Ng, V. (2014). *Automatic keyphrase extraction: A survey of the state of the art. Proceedings of the 52nd annual meeting of the association for computational linguistics (Vol. 1: Long papers)*. Baltimore, Maryland: Association for Computational Linguistics1262–1273.

Hauke, J., & Kossowski, T. (2011). *Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data.* 87–93.

Hulth, A. (2003). *Improved automatic keyword extraction given more linguistic knowledge. Proceedings of the conference on empirical methods in natural language processing – EMNLP*216–223.

Hyung, Z., Park, J. S., & Lee, K. (2017). Utilizing context-relevant keywords extracted from a large collection of user-generated documents for music discovery. *Information Processing and Management, 53*, 1185–1200.

Kim, S. N., Medelyan, O., Kan, M. Y., & Baldwin, T. (2010). *Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. Proceedings of the 5th international workshop on semantic evaluation – SemEval*21–26.

Lahiri, S., Choudhury, S. R., & Caragea, C. (2014). Keyword and keyphrase extraction using centrality measures on collocation networks. *CoRR*. abs/1401.6571.

Litvak, M., Last, M., Aizenman, H., Gobits, I., & Kandel, A. (2011). Degext — a language-independent graph-based keyphrase extractor. In E. Mugellini, P. S. Szczepaniak, M. C. Pettenati, & M. Sokhn (Eds.). *Advances in intelligent web mastering–3* (pp. 121–130). Springer Berlin Heidelberg.

Liu, Z., & Sun, M. (2012). Can prior knowledge help graph-based methods for keyword extraction? *Frontiers of Electrical and Electronic Engineering, 7*, 242–253.

Lu, Y., Cohen, I., Zhou, X. S., & Tian, Q. (2007). *Feature selection using principal feature analysis. Proceedings of the 15th ACM international conference on multimedia - MM*301–304.

Marujo, L., Gershman, A., Carbonell, J. G., Frederking, R. E., & Neto, J. P. (2012). *Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. Proceedings of the eighth international conference on language resources and evaluation - LREC*399–403.

Masucci, A. P., & Rodgers, G. J. (2006). Network properties of written human language. *Physical Review E, 74*, 026102.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In D. Lin, & D. Wu (Eds.). *Proceedings of the conference on empirical methods in natural language processing - EMNLP* (pp. 404–411). Association for Computational Linguistics.

Morillo, F., & Álvarez-Bornstein, B. (2018). How to automatically identify major research sponsors selecting keywords from the wos funding agency field. *Scientometrics, 117*, 1755–1770.

Nasar, Z., Jaffry, S. W., & Malik, M. K. (2018). Information extraction from scientific articles: A survey. *Scientometrics, 117*, 1931–1990.

Pastor-Satorras, R., Castellano, C., Van Mieghem, P., & Vespignani, A. (2015). Epidemic processes in complex networks. *Reviews of Modern Physics, 87*, 925–979.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Raamkumar, A. S., Foo, S., & Pang, N. (2017). Using author-specified keywords in building an initial reading list of research papers in scientific paper retrieval and recommender systems. *Information Processing and Management, 53*, 577–594.

Rousseau, F., Kiagias, E., & Vazirgiannis, M. (2015). *Text categorization as a graph classification problem. Proceedings of the international joint conference on natural language processing - IJCNLP*1702–1712.

Schoch, D., Valente, T. W., & Brandes, U. (2017). Correlations among centrality indices and a class of uniquely ranked graphs. *Social Networks, 50*, 46–54.

Tixier, A. J. P., Malliaros, F. D., & Vazirgiannis, M. (2016). A graph degeneracy-based approach to keyword extraction. In J. Su, X. Carreras, & K. Duh (Eds.). *Proceedings of the conference on empirical methods in natural language processing - EMNLP* (pp. 1860–1870). .

Tsatsaronis, G., Varlamis, I., & Nørvåg, K. (2010). *Semanticrank: Ranking keywords and sentences using semantic graphs. Proceedings of the 23rd international conference on computational linguistics - COLING*1074–1082.

Vega-Oliveros, D., Berton, L., Lopes, A., & Rodrigues, F. (2015). *Influence maximization based on the least influential spreaders. Socinf 2015, co-located with international joint conferences on artificial intelligence - IJCAIvol. 1398. Socinf 2015, co-located with international joint conferences on artificial intelligence - IJCAI* 3–8.

Vega-Oliveros, D. A., Berton, L., Vazquez, F., & Rodrigues, F. A. (Berton, Vazquez, Rodrigues, 2017a). *The impact of social curiosity on information spreading on networks. Proceedings of the IEEE/ACM international conference on advances in social networks analysis and mining - ASONAM*459–466.

Vega-Oliveros, D. A., da F Costa, L., & Rodrigues, F. A. (Costa, and Rodrigues, 2017b). *Influence maximization on correlated networks through community identification* arXiv:1705.00630.

Wan, X., & Xiao, J. (2008). *Single document keyphrase extraction using neighborhood knowledge. Proceedings of the 23rd national conference on artificial intelligence - AAAI*855–860.

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science, 2*, 165–193.

Xu, J., Bu, Y., Ding, Y., Yang, S., Zhang, H., Yu, C., & Sun, L. (2018). Understanding the formation of interdisciplinary research from the perspective of keyword evolution: A case study on joint attention. *Scientometrics, 117*, 973–995.