An Overview of Graph-Based Keyword Extraction Methods and Approaches

Article *in* Journal of Information and Organizational Sciences · July 2015

CITATIONS
89

READS
4,996

3 authors:

Slobodan Beliga
University of Rijeka
10 PUBLICATIONS **136** CITATIONS
SEE PROFILE

Ana Meštrović
University of Rijeka
53 PUBLICATIONS **392** CITATIONS
SEE PROFILE

Sanda Martincic-Ipsic
University of Rijeka
72 PUBLICATIONS **307** CITATIONS
SEE PROFILE

An Overview of Graph-Based Keyword Extraction Methods and Approaches

Slobodan Beliga *sbeliga@inf.uniri.hr*
Department of Informatics
University of Rijeka, Rijeka, Croatia

Ana Meštrović *amestrovic@inf.uniri.hr*
Department of Informatics
University of Rijeka, Rijeka, Croatia

Sanda Martinčić-Ipšić *smarti@inf.uniri.hr*
Department of Informatics
University of Rijeka, Rijeka, Croatia

Abstract

The paper surveys methods and approaches for the task of keyword extraction. The systematic review of methods was gathered which resulted in a comprehensive review of existing approaches. Work related to keyword extraction is elaborated for supervised and unsupervised methods, with a special emphasis on graph-based methods. Various graph-based methods are analyzed and compared. The paper provides guidelines for future research plans and encourages the development of new graph-based approaches for keyword extraction.
Keywords: keyword extraction, graph-based methods, selectivity-based keyword extraction

1. Introduction

Keyword extraction (KE) is tasked with the automatic identification of a set of the terms that best describe the subject of a document [1], [6], [8], [15], [24], [27], [36], [41], [54], [64]. Different terminology for defining the terms that represent the most relevant information contained in the document is used: key phrases, key segments, key terms or just keywords. All listed variants have the same function – to characterize the topics discussed in a document [41]. Extracting a small set of units, composed of one or more terms, from a single document is an important problem in Text Mining (TM), Information Retrieval (IR) and Natural Language Processing (NLP).

Keywords are widely used to enable queries within IR systems as they are easy to define, revise, remember, and share. Keywords are independent of any corpus and can be applied across multiple corpora and IR systems [6]. Keywords have also been applied to improve the functionality of IR systems [6], [12]. In other words, relevant extracted keywords can be used to build an automatic index for a document collection or alternatively they can be used for document representation in categorization or classification tasks [27], [41]. An extractive summary of the document is also the task of many IR and NLP applications and includes automatic indexing, automatic summarization, document management, high-level semantic description, text, document or website categorization or clustering, cross-category retrieval, constructing domain-specific dictionaries, name entity recognition, topic detection, tracking, etc. [6], [20], [33], [42], [57].

While assigning keywords to documents manually is a very costly, time consuming and tedious task, in addition to which, the number of digitally available documents is growing, automatic keyword extraction has attracted the interest of researchers over the last years.

Although the keyword extraction applications usually work on single documents, keyword extraction is also used for a more complex tasks (i.e. keyword extraction for the whole collection [58], the entire web site or for automatic web summarization [63]). With the appearance of big-data, constructing an effective model for text representation becomes even more urgent and demanding at the same time [29]. State-of-the-art techniques for KE encounter scalability and sparsity problems. In order to circumvent these limitations, new solutions are constantly being proposed.

This work presents a comprehensive overview of the common techniques and methods with the emphasis on new graph-based methods, especially regarding keyword extraction for the Croatian language. We systematize the existing state-of-the-art keyword extraction methods and approaches as well as new graph-based methods that are based on the foundations of graph theory. Additionally, the paper explores the advantages of graph-based methods over traditional supervised methods.

The paper is organized as follows: firstly, we systematize keyword extraction methods; secondly, we present a brief overview of various measures for network (graph) analysis; thirdly, we describe related work for supervised and unsupervised methods, with special emphasis on graph-based keyword extraction; fourthly, we compare graph-based measures of experiments extracting keywords from Croatian News articles; and finally, we conclude with some remarks regarding network-enabled extraction and turn to brief guidelines for future research.

2. Systematization of Methods

Keyword assignment methods can be divided roughly into two categories: (1) keyword assignment and (2) keyword extraction [14], [32], [53], [57] as presented in Figure 1. Both revolve around the same problem – selecting the best keyword. In **keyword assignment**, keywords are chosen from a controlled vocabulary of terms or predefined taxonomy, and documents are categorized into classes according to their content. **Keyword extraction** enriches a document with keywords that are explicitly mentioned in text [37]. Words that occurred in the document are analyzed in order to identify the most representative ones, usually exploring the source properties (i.e. frequency, length) [61]. Commonly, keyword extraction does not use a predefined thesaurus to determine the keywords.

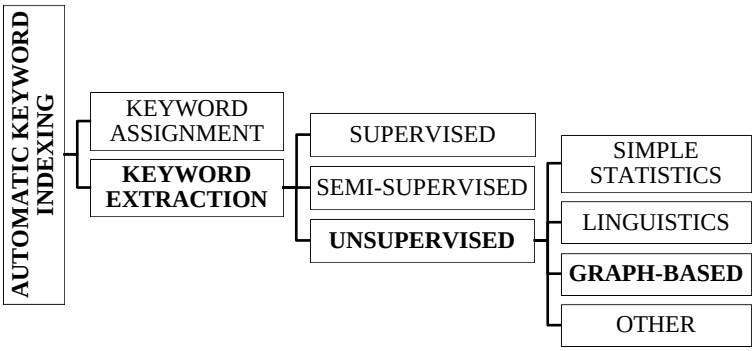

Figure 1. Classification of keyword extraction methods.

The scope of this work is calibrated only on keyword extraction methods. Existing methods for automatic keyword extraction can be according to Ping-I and Shi-Jen [10] divided roughly into:

- Statistical Approaches and
- Machine Learning Approaches,

or slightly more detailed in the four categories according to Zhang et al. [61]:

- Simple Statistical Approaches,

- Linguistic Approaches,
- Machine Learning Approaches and
- Other Approaches.

Simple Statistical Approaches comprise of simple methods which do not require the training data. In addition, these methods are language and domain-independent. Commonly, the statistics of the words from a document can be used to identify keywords: n-gram statistics, word frequency, TF-IDF (term frequency-inverse document frequency) model, word co-occurrences, PAT Tree (Patricia Tree; a suffix tree or position tree), etc. The disadvantage is that in some professional texts, such as from the health and medical domain, the most important keyword may appear only once in the article (e.g. diagnosis). The use of statistically empowered models may inadvertently filter out these words [10].

Linguistic Approaches use the linguistic properties of the words, sentences and documents. Lexical, syntactic, semantic and discourse analysis are some of the most commonly examined properties, although they are demanding and complex NLP problems.

Machine Learning Approaches consider supervised or unsupervised learning from the examples, but related work on keyword extraction prefers the supervised approach. Supervised machine learning approaches induce a model which is trained on a set of keywords. They require manual annotations of the learning dataset which is extremely tedious and inconsistent (sometimes requesting predefined taxonomy). Unfortunately, authors usually assign keywords to their documents only when they are compelled to do so. The model can be induced using one of the machine learning algorithms: Naïve Bayes, SVM (Support Vector Machines), C4.5, etc. Thus, supervised methods require training data, and are often dependent on the domain. A system needs to re-learn and establish the model every time when a domain changes [22], [46]. Model induction itself can also be demanding and time consuming on massive datasets.

Other Approaches for keyword extraction in general combine all the methods mentioned above. Additionally, sometimes for fusion they incorporate heuristic knowledge, such as the position, the length, the layout features of the terms, html and similar tags, the text formatting information etc.

Vector space model (VSM) is well-known and is the most used model for text representation in text mining approaches [5], [14], [18]. Specifically, the documents represented in the form of feature vectors are located in a multidimensional Euclidean space. This model is suitable for capturing simple word frequency, however structural and semantic information are usually disregarded. Due to its simplicity VSM has several disadvantages [49]:

- the meaning of a text and structure cannot be expressed explicitly,
- each word is independent from other, word appearance sequences or other relations are disregarded,
- if two documents have a similar meaning expressed with different words, similarity cannot be computed easily.

Graph-based text representation efficiently addresses these problems [49]. A graph is a mathematical model, which enables the exploration of the relationships and structural information very effectively. More about the graph representations of text is discussed in Section 3, and in [4], [35], [48], [49], [56]. For now, in short, document is modelled as graph where terms (words) are represented by vertices (nodes) and their relations are represented by edges (links). The taxonomy of the graph-enabled keyword extraction methods is presented in Figure 4.

The edge relation between words can be established on many principles exploiting different scopes of the text or relations among words for the graph's construction [35], [49]:

- co-occurrence relations – connecting neighboring words co-occurring within the window of a fixed size in text; or connecting all words co-occurring together in a sentence, paragraph, section or document (adding them to the graph as a clique[1]);

[1] Clique is a subgraph of a graph in which every two vertices are connected (a subgraph which is a complete graph).

- syntax relations – connecting words according to their relations in the syntax dependency graph;
- semantic relations – connecting words that have similar meanings, words spelled the same way but have different meanings, synonyms, antonyms, homonyms, etc;
- other possible relations – for example, intersecting words from a sentence, paragraph, section or document, etc.

There are various possibilities for the analysis of a network structure (topology) and we will focus on the most common – network structure of the linguistic elements themselves using various relations: semantic, pragmatic, syntax, morphology, phonetic and phonology. More precisely, in this work we narrow the scope of the study to (1) **co-occurrence** [7], (2) **syntactic** [28], (3) **semantic** [56] and (4) **similarity** networks [35].

2.1. Graph types

The formal definition of a graph according to graph theory is given in Section 3. Here we broadly discuss the classification of a graph-based method which can be established on the (1) **vertices** or (2) **edges** [35].

In vertex representation models, vertices represent advanced concepts which can be **atomic** (one component; also called homogenous) or **multiple** (more than two components; also called heterogeneous). The homogeneous graph model is usually used for the representation of grammatical associations between words or semantic similarities [9], [26]. Additionally, vertices can also be weighted or unweighted which conditions the representation model, which is respectively (1) weighted or (2) unweighted graph. Weighted vertices in this case commonly indicate the importance of the vertex in the graph, and different measures (explained in Section 3) are used to calculate the importance of the vertex. The measures and algorithms listed in Table 1, very often take into account the number of edges, the weight of vertices which are connected by the edge, etc.

Between two vertices, relationships can be established by edges. In edge representation models (graphs) graphs can be either (1) **directed** (called digraph, e.g. for word order in text) or (2) **undirected** (for connecting related words). Edges can also be (1) **weighted** or (2) **unweighted**, depending on relationships between vertices. In a language complex network, weight could be the distance of two words in paragraphs or text or the frequency of word pairs' co-occurrence. Beside weights, edge models can be (1) **labeled** or (2) **unlabeled**. It is almost conventional to explain the relationships or rules between related vertices by the edge label in many graph models in computer science (e.g. Entity-Relationship). In related work of graphs in the language's edge label can denote POS (part of speech), grammatical rule of word, etc.

There are also more complex models that are represented by combinations of the previously described models or parts of their structure. These are: (1) **multigraphs** – this model allows a connection with a plurality of different edges, and also a vertex connection with itself, (2) **hipergraph** – one connection can be established with any number of vertices; edges are not binary relations, (3) **multiplex** – a multilayer graph which shares the same vertices at all levels, and has edges between levels that are achieved by connecting only the same vertices.

An example of such a model is the multiplex of many realizations of the same text, always containing the same set of words interlinked with different edges: as direct neighbors, co-occurrence in the sentence, syntax dependencies, etc.

The classifications of graph types with all previous described features are shown in Figure 2 according to concepts, weight, direction or label for a vertex or edge representation model. The classifications of advanced graph models are shown in Figure 3.

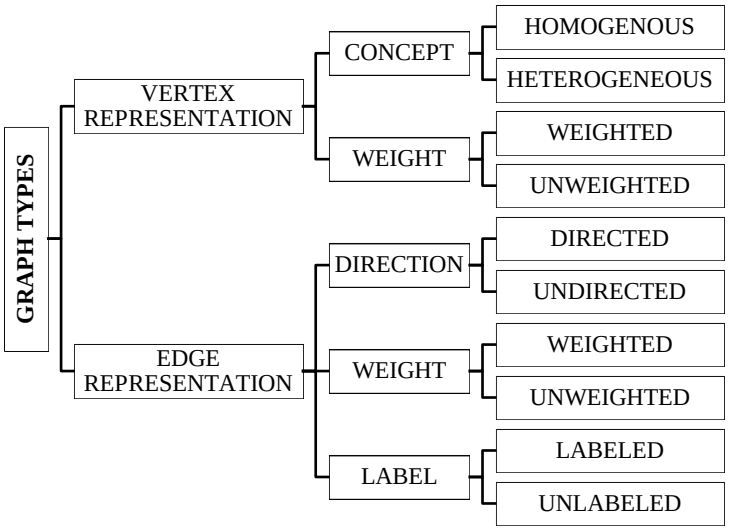

Figure 2. Classification of graph types.

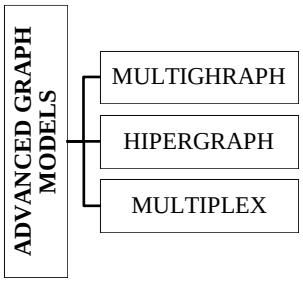

Figure 3. Classification of advanced graph models.

3. Graph-based centrality measures

This section defines some basic concepts from graph theory and the centrality measures necessary for understanding the graph-based approach. More details about graph measures can be found in [7], [11], [30], [38].

A graph is an ordered pair $G = (V, E)$ where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. A graph is directed if the edges have a direction associated with them. A graph is weighted if there is a weight function ω that assigns value (real number) to each edge. We use $N = |V|$ and $K = |E|$ as shorthand for the number of vertices and edges in a graph.

A path in a graph is a sequence of edges which connects a sequence of vertices which are all distinct from one another. A shortest path between two vertices u and v is a path with the shortest length and it is called the distance between u and v.

In the graph theory centrality measures refer to indicators which identify the most important vertices within a graph and that approach is used for the task of ranking the vertices. In the domain of keyword extraction various centrality measures are proposed and used for the task of ranking the words in a text.

Centrality measures are local graph measures, focused on a single vertex and its neighborhood. The neighborhood of a vertex v in graph G is defined as a set of neighbors of a vertex v and is denoted by $N(v)$. The neighborhood size is the number of immediate neighbors to a vertex. The number of edges between all neighbors of a vertex is denoted by $E(v)$. In the directed graph, the set of $N_{in}(v)$ is the set of vertices that point to a vertex v (predecessors) and set of $N_{out}(v)$ is the set of vertices that vertex v points to (successors).

The clustering coefficient of a vertex measures the density of edges among the immediate neighbors of a vertex. It determines the probability of the presence of an edge between any

two neighbors of a vertex. It is calculated as a ratio between the number of edges E_i that actually exist among these and the total possible number of edges among neighbors:

$$c(v) = \frac{2E(v)}{|N(v)|(|N(v)|-1)}. \quad (1)$$

The degree $d(v)$ of a vertex v is the number of edges at v; it is equal to the number of neighbors of v.

In a directed graph, the in-degree of a vertex v, $d^{in}(v)$ is defined as the number of inward edges from a vertex v. Analogously, the out-degree of a vertex v, $d^{out}(v)$ is defined as the number of outward edges from a vertex v.

The degree centrality $C_d(v)$ of a vertex v is defined as the degree of the vertex. It can be normalized by dividing it by the maximum possible degree $N-1$:

$$C_d(v) = \frac{d(v)}{N-1}. \quad (2)$$

In the directed graph the in-degree centrality of the vertex v is defined as in-degree of the vertex (normalized by dividing it by the maximum possible degree $N-1$):

$$C_d{}^{in}(v) = \frac{d^{in}(v)}{N-1}. \quad (3)$$

The out-degree centrality $C_d{}^{out}(v)$ of a vertex v is defined analogously.

The strength of the vertex v is a sum of the weights of all the edges incident with the vertex v:

$$s(v) = \sum_u w_{vu} \quad (4)$$

In the directed network, the in-strength $s^{in}(v)$ of the vertex v is defined as the sum of all weights of inward edges from a vertex v:

$$s^{in}(v) = \sum\nolimits_u w_{uv}. \quad (5)$$

The out-strength $s^{out}(v)$ of a vertex v is defined analogously.

The selectivity measure is introduced in [30]. It is an average strength of a vertex. For the vertex v the selectivity is calculated as a fraction of the vertex strength and vertex degree:

$$e(v) = \frac{s(v)}{d(v)}. \quad (6)$$

In the directed network, the in-selectivity of the vertex v is defined as:

$$e^{in}(v) = \frac{s^{in}(v)}{d^{in}(v)}. \quad (7)$$

The out-selectivity $e^{out}(v)$ of a vertex v is defined analogously.

The closeness centrality $C_c(v)$ of a vertex v is defined as the inverse of farness, i.e. the sum of the shortest distances between a vertex and all the other vertices in a graph. Let d_{vu} be the shortest path between vertices u and v. The normalized closeness centrality of a vertex v is given by:

$$C_c(v) = \frac{N-1}{\sum_{v \neq u} d_{vu}}. \quad (8)$$

The betweenness centrality $C_b(v)$ of a vertex v quantifies the number of times a vertex acts as a bridge along the shortest path between two other vertices. Let σ_{ut} be the number of the shortest paths from vertex u to vertex t and let $\sigma_{ut}(v)$ be the number of those paths that pass through the vertex v. The normalized betweenness centrality of a vertex v should be divided by the number of all possible edges in the graph and is given by:

$$C_b(v) = \frac{2\sum_{v \neq u \neq t} \frac{\sigma_{ut}(v)}{\sigma_{ut}}}{(N-1)(N-2)}. \quad (9)$$

The eigenvector centrality $C_{EV}(v)$ measures the centrality of a vertex v as a function of the centralities of its neighbors. For the vertex v and constant λ it is defined:

$$C_{EV}(v) = \frac{1}{\lambda} \sum_{u \in N(v)} C_{EV}(u). \tag{10}$$

In the case of weighted networks, the equation can be generalized. Let w_{uv} be the weight of edge between vertices u and v and λ a constant. The eigenvector centrality of a vertex v is given by:

$$C_{EV}(v) = \frac{1}{\lambda} \sum_{u \in N(v)} w_{uv} \times C_E(u). \tag{11}$$

There are various centrality measures based on the idea of eigenvector centrality defined.

The HITS method defines authority $x(v)$ and a hub score $y(v)$ for vertex v. Let e_{vu} represent the directed edge from vertex v to vertex u. Given that each vertex has been assigned an initial authority score $x(v)^{(0)}$ and hub score $y(v)^{(0)}$ as described in [25], HITS iteratively refines these scores by computing:

$$x(v)^{(i)} = \sum_{u:e_{uv} \in E} y(u)^{(k-1)} \text{ and } x(v)^{(i)} = \sum_{u:e_{uv} \in E} y(u)^{(k-1)} \text{ for } k = 1,2,3, \ldots \tag{11}$$

The TextRank centrality is based on the eigenvector centrality measure and implements the concept of "voting". The TextRank score of a vertex v is initialized to a default value and computed iteratively until convergence using the following equation:

$$C_{PageRank}(v) = (1-d) + d \sum_{u \in N_{in}(v)} \frac{C_{PageRank}(u)}{|N_{out}(u)|} \tag{12}$$

where d is the dumping factor set between 0 and 1 (usually set to 0.85).

The TextRank is a modification of a PageRank defined for weighted graphs and used for ranking words in the texts. The equation is:

$$C_{TextRank}(v) = (1-d) + d \sum_{u \in N_{in}(v)} \frac{w_{uv} \times C_{TextRank}(u)}{\sum_{t \in N_{out}(u)} w_{ut}}. \tag{13}$$

4. Related Work on Keyword Extraction

Although the keyword extraction methods can be divided as (1) **document-oriented** and (2) **collection-oriented**, we are most interested in some of the other systematization in order to get a broad overview of the field. The approaches for keyword extraction can be roughly categorized into either (1) **unsupervised** or (2) **supervised**. Supervised approaches require an annotated data source, while the unsupervised require no annotations in advance. The massive use of social networks and Web 2.0 tools has caused turbulence in the development of new methods for keyword extraction. In order to improve the performance of methods on massive quantities of data (3) **semi-supervised** methods have come into research focus. Figure 1 shows the different techniques that are combined into supervised, unsupervised or both approaches.

Two critical issues of supervised approaches are demands to prepare the training data with manually annotated keywords and the bias towards the domain on which they are trained. For this reason in this work, the focus has been shifted towards more unsupervised methods, specifically graph-based methods which have been developed using only the statistics of the source which is reflected into the structure of the graph (network).

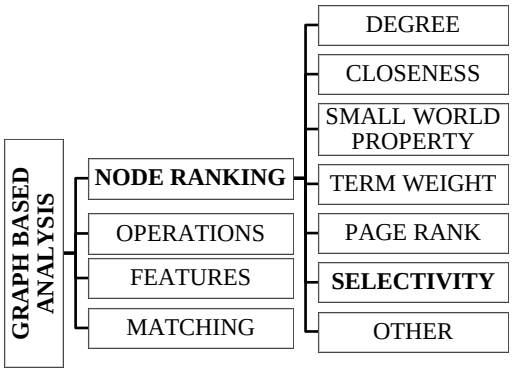

Figure 4. Classification of graph-based methods, modified from [49].

4.1. Supervised

The main idea of supervised methods is to transform keywords extraction into a binary classification task – word is either a keyword or not. Two typical and well-known systems for supervised keyword extraction, which set the boundaries of the research field are KEA (Witten et al., 1999 [57]) and GenEx (Turney, 1999 [53]) [53], [57]. The most important features for classifying a keyword candidate in these systems are the frequency and location of the term in the document. In short, GenEx uses Quinlan's C4.5 decision tree induction algorithm to his learning task, while KEA uses Naïve Bayes algorithm for training and keyphrase extraction. GenEx and KEA are extremely important systems because, in this field of keyword extraction, they set up the foundation for all other methods that were developed after, and have become the state-of-the-art benchmark for evaluating the performance of other methods.

Hulth (2003) in [20] explores the incorporation of linguistic knowledge into the extraction procedure and uses Noun Phrase chunks (NP) (rather than term frequency and n-grams), and adds the POS (Part-of-Speech) tag(s) assigned to the term as a feature. In more details, extracting NP-chunks gives better precision than n-grams, and by adding the POS tag(s) to the terms improves the results independent of the term selection approach applied.

Turney (2003) in [52] implements enhancements to the KEA keyphrase extraction algorithm by using statistical associations between keyphrases and enhances the coherence of the extracted keywords.

Song et al. (2003) represent the Information Gain-Based keyphrase extraction system called KPSpotter [50].

HaCohen-Kerner et al. (2005) in [16] investigate the automatic extraction and learning of keyphrases from scientific articles written in English. They use various machine learning (ML) methods and report that the best results are achieved with J48 (an improved variant of C4.5).

Medelyan and Witten (2006) propose a new method called KEA++, which enhances automatic keyphrase extraction by using semantic information on terms and phrases gleaned from a domain-specific thesaurus [32]. KEA++ is actually an improved version of the previously mentioned KEA devised by Witten et al. Zhang Y. et al.

The group of researchers in [62] (2006) propose the use of not only "global context information", but also "local context information". For the task of keyword extraction they engage Support Vector Machines (SVM). Experimental results indicate that the proposed SVM based method can significantly outperform the baseline methods for keyword extraction.

Wang (2006) in [55] exploits different text features in order to determine whether a phrase is a keyphrase: TF and IDF, appearance in the title or headings (subheadings) of the given document, and the frequency appearing in the paragraphs of the given document in the combination with Neural Networks are proposed.

Nguyen and Kan (2007) [39] propose an algorithm for keyword extraction from scientific publications using linguistic knowledge. They introduce features that capture salient morphological phenomena found in scientific keyphrases, such as whether a candidate keyphrase is an acronym or weather it uses specific terminologically productive suffixes.

Zhang C. et al. (2008) in [61] implement a keyword extraction method from documents using Conditional Random Fields (CRF). The CRF model is a state-of-the-art sequence labeling method, which can use the features of documents more sufficiently and efficiently, and considers the keyword extraction as the string labeling task. The CRF model outperforms other ML methods such as SVM, Multiple Linear Regression model, etc.

Krapivin et al. (2010) in [23] use NLP techniques to improve various ML approaches (SVM, Local SVM, Random Forests) to the task of automatic keyphrase extraction from scientific papers. Evaluation shows promising results that outperform state-of-the-art Bayesian learning system KEA on the same dataset without the use of controlled vocabularies.

4.2. Unsupervised

HaCohen-Kerner (2003) in [17] presents a simple model that extracts keywords from abstracts and titles. The model uses unigrams, 2-grams and 3-grams, and a stopwords[2] list. The highest weighted group of words (merged and sorted n-grams) is proposed as keywords.

Pasquier (2010) in [43] describes the design of a keyphrase extraction algorithm for a single document using sentence clustering and Latent Dirichlet Allocation. The principle of the algorithm is to cluster sentences of the documents in order to highlight parts of text that are semantically related. The clustering is performed by using the cosine similarity between sentence vectors, K-means, Markov Cluster Process (MCP) and ClassDens techniques. The clusters of sentences, that reflect the themes of the document, are analyzed for obtaining the main topic of the text. The most important words from these topics are proposed as keyphrases.

Pudota et al. (2010) in [44] design a domain independent keyphrase extraction system that can extract potential phrases from a single document in an unsupervised, domain-independent way. They engaged n-grams, but they also incorporated linguistic knowledge (POS tags) and statistics (frequency, position, lifespan) of each n-gram in defining candidate phrases and their respective feature sets.

Hurt in [21] examines the differences between author generated keywords and automatically generated keywords using an inverse frequency and maximum likelihood algorithm. They express results in terms of novel linguistic measure “keyness”, which is defined as a log-likelihood measure of the relatedness of one or more specified words, keywords, to a corpus of literature. Testing of these two methods, they show that there are no statistically significant differences in the achieved results.

Very recent research by Yang et al. (2013) [60] focuses on keyword extraction based on entropy difference between the intrinsic and extrinsic modes, which refers to the fact that relevant words significantly reflect the author’s writing intention. Their method uses the Shannon’s entropy difference between the intrinsic and extrinsic mode, which refers to the occurrences of words as being modulated by the author’s purpose, while the irrelevant words are distributed randomly in the text. They indicate that the ideas of this work can be applied to any natural language without requiring any previous knowledge semantics or syntax of the language, especially for single documents of which there is no a priori information available.

4.3. Graph-Based

Ohsawa et al. (1998) in [40] propose an algorithm for the automatic indexing by co-occurrence graphs constructed from metaphors, called KeyGraph. This algorithm is based on

[2] Stopwords are the most frequent function words, which do not carry strong semantic properties, but are needed for the syntax of the language.

the segmenting of a graph, representing the co-occurrence between terms in a document, into clusters. Each cluster corresponds to a concept on which the author's idea is based, and top ranked terms by a statistic based on each term's relationship to these clusters are selected as keywords. KeyGraph proved to be a content sensitive, domain independent device of indexing.

Matsou et al. (2001) in [31] present early research where a text document is represented as an undirected and unweighted co-occurrence network. Based on the network topology, the authors proposed an indexing system called KeyWorld, which extracts important terms (pairs of words) by measuring their contribution to small-world properties. The contribution of the vertex is based on the closeness centrality calculated as the difference in small-world properties of the network with the temporarily elimination of a vertex combined with the inverse document frequency (idf).

Erkan and Radev (2004) in [13] introduce a stochastic graph-based method for computing the relative importance of textual units on the problem of text summarization by extracting the most important sentences. LexRank calculates sentence importance based on the concept of the eigenvector centrality in a graphical representation of sentences. A connectivity matrix based on intra-sentence cosine similarity is used as the adjacency matrix of the graphical representation of sentences. LexRank is shown to be quite insensitive to the noise in the data.

Mihalcea and Tarau (2004) in [36] report upon a seminal research which introduced a state-of-the-art TextRank model. TextRank is derived from PageRank and introduced to graph based text processing, keyword and sentence extraction tasks. The abstracts are modeled as undirected or directed and weighted co-occurrence networks using a co-occurrence window of variable sizes (2-10). The lexical units are preprocessed: stopwords removed, words restricted with POS syntactic filters (open class words, nouns and adjectives, nouns). The PageRank motivated score of the importance of the vertex derived from the importance of the neighboring vertices is used for keyword extraction. The obtained TextRank performance compares favorably with the supervised machine learning n-gram based approach.

Mihalcea (2004) in [34] presents an extension to earlier work [36], where the TextRank algorithm is applied for the text summarization task powered by sentence extraction. In this task TextRank performed on a par with the supervised and unsupervised summarization methods, which motivated the new branch of research based on the graph-based extracting and ranking algorithms.

Xie (2005) in [59] studies different centrality measures in order to predict noun phrases that appear in the abstracts of scientific articles. The measures tested are: degree, closeness, betweenness and information centrality. Their results show that centrality measures improve the accuracy of the prediction in terms of both precision and recall. Furthermore, the method of constructing a noun-phrase (NP) network significantly influences the accuracy when using the centrality heuristic itself, but is negligible when it is used together with other text features in decision trees.

Huang et al. (2006) [19] propose an automatic keyphrase extraction algorithm using an unsupervised method also based on connectedness and betweenness centrality.

Palshikar (2007) in [41] proposes a hybrid structural and statistical approach to extract keywords from a single document. The undirected co-occurrence network, using a dissimilarity measure between two words, calculated from the frequency of their co-occurrence in the preprocessed and lemmatized document, as the edge weight, was shown to be appropriate for the centrality measures based approach for keyword extraction.

Wan and Xiao (2008) in [54] propose a small number of nearest neighbor documents to provide more knowledge to improve single document keyphrase extraction. A specified document is expanded to a small document set by adding a few neighbor documents close to the document using a cosine similarity measure, while the term weight is computed by TF-IDF. The local information in the specified document and the global information in all the neighboring documents are taken into consideration along with the expanded document set using a graph-based ranking algorithm.

Litvak and Last (2008) in [26] compare supervised and unsupervised approaches for keywords identification in the task of extractive summarization. The approaches are based on

the graph-based syntactic representation of text and web documents. The results of the HITS algorithm on a set of summarized documents performed comparably to supervised methods (Naïve Bayes, J48, SVM). The authors suggest that simple degree-based rankings from the first iteration of HITS, rather than running it to its convergence, should be considered.

Grineva et al. (2009) in [15] use community detection techniques for the extraction of key terms on Wikipedia's texts, modeled as a graph of semantic relationships between terms. The results show that the terms related to the main topics of the document tend to form a community, thematically cohesive groups of terms. Community detection allows the effective processing of multiple topics in a document and efficiently filters out noise. The results achieved on weighted and directed networks from semantically linked, morphologically expanded and disambiguated n-grams from the articles' titles. Additionally, for the purpose of testing noise stability, they repeated the experiment on different multi-topic web pages (news, blogs, forums, social networks, product reviews) which confirmed that community detection outperforms TF-IDF model.

Tsatsaronis et al. (2010) in [51] present SemanticRank, a network based ranking algorithm for keyword and sentence extraction from text. Semantic relation is based on the calculated knowledge-based measure of semantic relatedness between linguistic units (keywords or sentences). The keyword extraction from the Inspec abstracts' results reported a favorable performance of SemanticRank over state-of-the-art counterparts - weighted and unweighted variations of PageRank and HITS.

Litvak et al. (2011) in [27] introduce DegExt, a graph-based language independent keyphrase extractor, which extends the keyword extraction method described in [26]. They also compare DegEx with state-of-the-art approaches: GenEx [53] and TextRank [36]. DegEx surpasses both in terms of precision, implementation simplicity and computational complexity.

Boudin (2013) in [8] compares various centrality measures for graph-based keyphrase extraction. Experiments on standard data sets of English and French show that simple degree centrality achieves results comparable to the widely used TextRank algorithm; and that closeness centrality obtains the best results on short documents. Undirected and weighted co-occurrence networks are constructed from syntactically (only nouns and adjectives) parsed and lemmatized text using a co-occurrence window. Degree, closeness, betweenness and eigenvector centrality are compared to the PageRank motivated method proposed by Mihalcea (2004) in [36] as a baseline. Degree centrality achieves a similar performance as the much more complex TextRank. Closeness centrality outperforms TextRank on short documents (scientific papers abstracts).

Zhou et al. (2013) in [64] investigate a weighted complex network based keyword extraction incorporating the exploration of the network structure and linguistics knowledge. The focus is on the construction of a lexical network including the reasonable selection of vertices, the proper description of the relationships between words, a simple weighted network and TF-IDF. The reasonable selection of words from texts as lexical vertices from a linguistic perspective, the proper description of the relationship between words and the enhancement of vertex attributes attempt to represent texts as lexical networks more accurately. The Jaccard coefficient is used to reflect the associations or relationships of two words rather than the usual co-occurrence criteria in the process of network construction. The importance of each vertex to become a keyword candidate is calculated with closeness centrality. The compound measures that takes vertex's attributes (words length and IDF) are applied. Approach is compared with three competitive baseline approaches: binary network, simple weighted network and TF-IDF approach. Experiments for Chinese indicate that the lexical network constructed by this approach achieves preferable effect on accuracy, recall and F-score over the classic TF-IDF method.

Lahiri et al. (2014) in [24] extract keywords and keyphrases from co-occurrence networks of words and from noun-phrases collocations' networks. Eleven measures (degree, strength, neighborhood size, coreness, clustering coefficient, structural diversity index, page rank, HITS hub and authority score, betweenness, closeness and eigenvector centrality) are used for keyword extraction from directed/undirected and weighted networks. The obtained results

from four data sets suggest that centrality measures outperform the baseline term frequency – inverse document frequency (TF-IDF) model, and simpler measures such as degree and strength outperform computationally the more expensive centrality measures such as coreness and betweenness.

Abilhoa and de Castro (2014) in [1] propose a keyword extraction method representing tweets (microblogs) as graphs and apply centrality measures for finding the relevant keywords. They developed a technique named Twitter Keyword Graph where in the pre-processing step they use tokenization, stemming and stopwords' removal. Keywords are extracted from the graph cascade-like applying graph centrality measures – closeness and eccentricity. The performance of the algorithm is tested on a single text from the literature and compared with the TF-IDF approach and KEA algorithm. Finally, the algorithm is tested on five sets of tweets of increasing size. The computational time to run the algorithms proved to be a robust proposal to extract keywords from texts, especially from short texts such as micro blogs.

Beliga et al. (2014) in [4] propose the selectivity-based keyword extraction (SBKE) as a new unsupervised method for network-based keyword extraction. This approach is built with a new network measure - the vertex selectivity (defined as the average weight distribution on the edges of the single vertex) – see section 4. In [4] is also shown that selectivity slightly outperforms the standard centrality-based measures: in-degree, out-degree, betweenness and closeness. Vertices with the highest selectivity value are open-class words (content words) which are preferred keyword candidates (nouns, adjectives, verbs) or even part of collocations, keyphrases, names, etc. Selectivity is insensitive to non-content words or stopwords and therefore can efficiently detect semantically rich open-class words from the network and extract keyword candidates.

Centrality measures are discriminative properties of the importance of a vertex in a graph, and are directly related to the structure of the graph [1]. The Table 1 overviews network measures that are widely used in graph-based research on keyword extraction, together with additional measures from the NLP domain. Mark asterisk (*) denotes graph-based measures.

NAME	**DEFINITION**	**RESEARCH**
Degree*	Number of edges incident to a vertex.	[4], [8], [24], [59]
Strength*	Sum of the weights of the edges incident to a vertex.	[24]
Selectivity*	Fraction of the vertex strength and vertex degree (average strength).	[4]
Neighborhood size*	Number of immediate neighbors to a vertex.	[24]
Coreness*	Outermost core number of a vertex in the k-core decomposition of a graph.	[24]
Clustering Coefficient*	Density of edges among the immediate neighbors of a vertex.	[24]
Page Rank*	Importance of a vertex based on how many important vertices it is connected to.	[24], [51]
TextRank*	Modification of an algorithm derived from Google's PageRank is based upon the eigenvector centrality measure and implement the concept of "voting".	[8], [34]
HITS*	Importance of a vertex as a hub (pointing to many others) and as an authority (pointed to by many others).	[24], [26], [51]
Betweenness*	The fraction of shortest paths that pass through a vertex, calculated over all vertex pairs - the measure of how many shortest paths between all other node-pairs are traversing a node.	[4], [8], [19], [24], [59]

Closeness*	Reciprocal of the sum of distances of all vertices to some vertex.	[1], [4], [8], [24], [31], [59], [64]
Community detection*	Community detection techniques are based on the principles which detect nodes with dense internal connections and sparser connections between groups.	[15]
Eigenvector Centrality*	Element of the first eigenvector of a graph adjacency matrix corresponding to a vertex.	[8], [24]
Information Centrality	Generalization of betweenness centrality – focuses on the information contained in all paths originating with a specific actor.	[59]
Structural Diversity Index	Normalized entropy of the weights of the edges incident to a vertex.	[24]
The Jaccard coefficient or Jaccard index	Reflects the association or relationship of two words taking into account not only the co-occurrence frequency, but also the frequency of both words in a pair.	[64]
Information Gain	The Kullback-Leibler divergence – a measure of expected reduction in entropy based on the "usefulness" of an attribute.	[50]
TF, IDF, TF-IDF	Term frequency, inverse document frequency.	[15], [21], [24], [31], [41], [44], [53], [54], [55], [57], [64]
n-gram	N-gram is a contiguous sequence of n items from a given sequence of text or speech.	[17], [20], [36], [44]
Cosine similarity	Determines similarity between two vectors.	[13], [43], [54]
SingleRank	Compute word scores for each single document based on the local graph for the specified document.	[54]
ExpandRank	Compute word scores for each single document based on the neighborhood knowledge of other documents.	[54]
Shannon's entropy difference	The difference between the intrinsic and extrinsic entropy.	[60]
Keyphraseness	The linear combination of features: phrase frequency, pos value, phrase depth, phrase last occurrence, phrase lifespan.	[29]
Other	Harmonic centrality, LIN centrality, Katz centrality, Wiener index, eccentricity, connectedness [59], POS tags [20], [44], CRF [61], LexRank [13], SemanticRank [51], SimRank, etc.	

Table 1. Measures and algorithms used for keyword extraction.

5. Selectivity-Based Keyword Extraction

5.1. Dataset

For the network based keyword extraction we use the data set composed of Croatian news articles [37]. The data set contains 1020 news articles from the Croatian News Agency (HINA), with manually annotated keywords (key phrases) by human experts. The set is divided as such: 960 annotated documents for learning of supervised methods, and 60 documents for testing. The test set of 60 documents is annotated by 8 different experts. We selected the first 30 texts from HINA's collection for our experiment.

5.2. Co-occurrence Network Construction

Each text can be represented as a complex network of linked words: each individual word is a vertex and the interactions amongst words are edges. Co-occurrence networks exploit simple neighbor relation; two words are linked if they are adjacent in the sentence [27]. The weight of the edge is proportional to the overall co-occurrence frequencies of the corresponding word pairs within a corpus. From the documents in the HINA data set we construct directed and weighted co-occurrence networks: one from the text in each document.

5.3. Results

We compute centrality measures for each vertex in a network constructed from 60 news articles: in-degree, out-degree, closeness, betweenness and selectivity. Then we rank all vertices (words) according to the values of each of these measures, obtaining the top 15 keyword candidates automatically from the network. It is obvious that top 15 ranked words according to the in/out degree centrality, closeness centrality and betweenness centrality are stopwords (conjunctions, prepositions, determiners, etc.) – see Table 2. It can also be noticed that centrality measures return almost identical stopwords. However, the selectivity measure ranked only open-class words: nouns, verbs and adjectives. We expect that among these highly-ranked words are keyword candidates. The same results are shown in the preliminary research on keyword extraction from multitopic web documents [47].

	IN-DEGREE	**OUT-DEGREE**	**CLOSENESS**	**BETWEENNESS**	**SELECTIVITY**
1.	biti (is/be)	biti (is/be)	biti (is/be)	biti (is/be)	Bratislava
2.	i (and)	i (and)	i (and)	i (and)	području (area)
3.	u (in)	u (in)	taj (that/this)	u (in)	utorak (Tuesday)
4.	na (on)	na (on)	a (but/and)	na (on)	zaleđe hinterland)
5.	da (that/to)	sebe (self)	sebe self)	sebe (self)	revolucije (revolution)
6.	koji (which)	za (for)	on (He)	da (that/to)	provjera (check)
7.	a (for)	taj (that/this)	da (that/to)	taj (that/this)	II. (roman number)
8.	a (but/and)	da (that/to)	u (in)	koji (which)	desetljeća (decades)
9.	taj (that/this)	od (from)	ali (but)	za (for)	Balkanu (Balkan)
10.	sebe (self)	s (with)	za (for)	hrvatski (Croatian)	sloboda (freedom)
11.	s (with)	a (but/and)	kako (how)	a (but/and)	universe
12.	od (of)	koji (which)	hrvatski (Croatian)	od (from)	trophy
13.	ne (not/no)	ne (not/no)	još (more/yet)	s (with)	stotina (hundred)
14.	hrvatski (Croatian)	hrvatski (Croatian)	sad (now)	ne (not/no)	Splitu (Split)
15.	o (on/about)	će (will)	godina (year)	iz (from)	razlika (difference)

Table 2. The top 15 ranked words according to the measures: in-degree, out-degree, closeness, betweenness and selectivity from th whole HINA dataset.

In short, it seems that selectivity is insensitive to stopwords and therefore can efficiently detect semantically rich open-class words from the network and extract better keyword candidates (which are probably names, parts of collocations or key phrases).

Simple measures such as selectivity promulgates the views and opportunities for the development of new graph-based methods which can yield successful keyword ranking, and at the same time circumvent the usage of demanding NLP procedures, which are deeply rooted in standard KE techniques. If we take into consideration the complexity and computational resources, then it is clear that the graph-based methods may have the advantage over traditional supervised and unsupervised methods. This is the reason why it makes sense to continue the work towards developing new graph-based methods.

6. Conclusion and Future Trends

Keywords provide a compact representation of a document's content. Graph-based methods for keyword extraction are inherently unsupervised, and have the fundamental aim to build a network of words (phrases or linguistic units) and then rank the vertices exploiting the measures of the network structure, usually centrality motivated. This paper is a detailed systemization of existing approaches for keyword extraction: the review of related work on supervised and unsupervised methods with a special focus on the graph-based methods. The paper consolidates the most commonly used centrality measures that are essential in graph-based methods: in/out-degree, closeness, betweenness, etc. In addition, the existing work of Croatian extraction is included as well.

This work provides an insight into the related work of graph-based keyword extraction methods which successfully consolidated various techniques of natural language processing and complex network analysis. Combinations of these techniques establish a solid platform regard to the objectives of keyword (term) extraction and scope of the specific application. Such hybrid techniques represent new convenient ways to circumvent anomalies that occur in VSM and other traditionally used models.

Graph-based methods for keyword extraction are simple and robust in many ways: (1) they do not require advanced linguistic knowledge or processing, (2) they are domain independent and (3) they are language independent. Such graph-based KE techniques are certainly applicable for various tasks: text classification, summarization, search, etc. Due to the aforementioned benefits it is reasonable to expect that graph-based extraction will attract the attention of the research community in the future. It can be expected that many text and document analyses will incorporate graph-based keyword extraction.

Acknowledgments

This work has been supported in part by the University of Rijeka under the LangNet project (13.13.2.2.07).

References

[1] Abilhoa, W. D; de Castro, L. N. A keyword extraction method from twitter messages represented as graphs. *Applied Mathematics and Computation*, 240:308-325, 2014.

[2] Ahel, R; Dalbelo-Bašić, B; Šnajder, J. Automatic keyphrase extraction from Croatian newspaper articles. In *Proceesings of The Future of Information Sciences, Digital Resources and Knowledge Sharing*, pages 207-218, 2009.

[3] Bekavac, M; Šnajder, J. GPKEX: Genetically Programmed Keyphrase Extraction from Croatian Texts. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, ACL, pages 43-47, Sofia, 2013.

[4] Beliga, S; Meštrović, A; Martinčić-Ipšić, S. Toward Selectivity-Based Keyword Extraction for Croatian News. In CEUR Proceedings (SDSW 2014), Vol. 1301, pages 1-14, Riva del Garda, Trentino, Italy, 2014.

[5] Berry, M. W; Castellanos, M. *Survey of Text Mining II*, Springer, 2008.

[6] Berry, M. W; Kogan, J. *Text Mining: Applications and Theory*, Wiley, UK, 2010.

[7] Borge-Holthoefer, J; Arenas, A. Semantic Networks: Structure and Dynamics, *Entropy*, 12(5):1264-1302, 2010.

[8] Boudin, F. A comparison of centrality measures for graph-based keyphrase extraction. In International Joint Conference on Natural Language Processing (IJCNLP), pages 834-838, Nagoya, Japan, 2013.

[9] Chang, J.-Y; Kim, I.-M. Analysis and Evaluation of Current Graph-Based Text Mining Researches. *Advanced Science and Technology Letters 42 (Mobile and Wireless 2013)*, pages 100-103, Jeju Island, Korea, 2013.

[10] Chen, P; Lin, S. Automatic keyword prediction using Google similarity distance. *Expert Systems with Applications*, 37(3): 1928-1938, 2010.

[11] Divjak, B; Lovrenčić, A. *Diskretna matematika s teorijom grafova*. TIVA Tiskara Varaždin, 2005.

[12] Dobša, J. Information retrieval using latent semantic indexing. *Journal of Information and Organizational Sciences*, 26(1-2):13-23, 2002.

[13] Erkan, G; Radev, D. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal od Articial Intelligence Research (JAIR)*, 22(1):457-479, 2004.

[14] Feldman, R; Sanger, J. The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data, New York: Cambridge University Press, 2007.

[15] Grineva, M; Grinev, M; Lizorkin, D. Extracting Key Terms from Noisy and Multi-theme Documents. In *Proceedings of the 18th International Conference on World Wide Web*, pages 661-670, NY, USA, 2009.

[16] HaCohen-Keren, Y; Gross, Z; Masa, A. Automatic Extraction and Learning of Keyphrases from Scientific Articles. *Computational Linguistics and Intelligent Text Processing.* In Proceedings of 6th Int. Conference CICLing 2005, LNCS 3406, pages 657-669, Mexico City, Mexico, 2005.

[17] HaCohen-Kerner, Y. Automatic Extraction of Keywords from Abstracts. Knowledge-Based Intelligent Information and Engineering Systems, In Proceedings of 7th International Conference KES 2003, (LNCS 2773), pages 843-849, 2003.

[18] Hotho, A; Nürnberger, A; Paaß, G. A Brief Survey of Text Mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20(1): 19-62, 2005.

[19] Huang, C; Tian, Y; Zhou, Z; Ling, C. X; Huang, T. Keyphrase extraction using semantic networks structure analysis. In *IEEE International Conference on Data Mining*, ICDM '06, pages 275-284, Hong Kong , 2006.

[20] Hulth, A. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. *In Proceedings of EMNLP 2003*, pages 216-223, Stroudsburg, USA, 2003.

[21] Hurt, C. D. Automatically Generated Keywords: A Comparison to Author-Generated Keywords in the Sciences. *Journal of Information and Organizational Sciences*, 34(1):81-88, 2010.

[22] Jones, K. S. Informaion retrieval and artificial inteligence. *Artificial Intelligence*, 114(1-2): 257-281, 1999.

[23] Krapivin, M; Autayeu, A; Marchese, M; Blanzieri, E; Segata, N. Keyphrases Extraction from Scientific Documents: Improving Machine Learning Approaches with Natural Language Processing. In Proceedings of 12th International Conference on Asia-Pacific Digital Libraries, ICADL 2010, LNAI 6102, pages 102-111, Gold Coast, Australia 2010.

[24] Lahiri, S; Choudhury, S. R; Caragea, C. Keyword and Keyphrase Extraction Using Centrality Measures on Collocation Networks, Cornell University Library, *arXiv preprint arXiv:1401.6571*, 2014.

[25] Langville, A; Meyer, C. A survey of eigenvector methods of web information retrieval. *SIAM Review*, 47(1):135-161, 2005.

[26] Litvak, M; Last, M. Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, MMIES '08, pages 17-24, Manchester, UK, 2008.

[27] Litvak, M; Last, M; Aizenman, H; Gobits, I; Kandel, A. DegExt – A Language-Independent Graph-Based Keyphrase Extractor. *Advances in Intelligent Web Mastering – 3*, AISC, 86:121-130, 2011.

[28] Liu, H; Fengguo H. What role does syntax play in a language network?. *EPL (Europhysics Letters)*, 83(1): 18002, 2008.

[29] Manyika, J; Chui, J; Brown, B; Bughin, J; Dobbs, R; Roxburgh, C; Byers, A. H. *Big data: The next frontier for innovation, competition, and productivity*, McKinsey Global Institute, 2011.

[30] Masucci, A; Rodgers, G. Diferences between normal and shufled texts: structural properties of weighted networks. *Advances in Complex Systems*, 12(01):113-129, 2009.

[31] Matsuo, Y; Ohsawa, Y; Ishizuka, M. Keyworld: Extracting keywords from document s small world. In *Discovery Science*, pages 271-281, 2001.

[32] Medelyan, O; Witten, I. H. Thesaurus Based Automatic Keyphrase Indexing. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 296-297, New York, USA, 2006.

[33] Meštrović, A; Martinčić-Ipšić, S; Čubrilo, M. Weather forecast data semantic analysis in f-logic. *Journal of Information and Organizational Sciences*, 31(1): 115-129, 2007.

[34] Mihalcea, R. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics*, ACL 2004, 2004.

[35] Mihalcea, R; Radev, D. Graph-based Natural Language Processing and Information Retrieval, Cambridge University Press, 2011.

[36] Mihalcea, R; Tarau, P. TextRank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 104-411, Barcelona, Spain, July 2004.

[37] Mijić, J; Dalbelo Bašić, B; Šnajder, J. Robust Keyphrase Extraction for a Large-Scale Croatian News Production System. In *Proceedings of International Conference on Formal Approaches to South Slavic and Balkan Languages*, pages 59-66, Zagreb, 2010.

[38] Newman, M. E. J. *Networks: An Introduction*, Oxford University Press, 2010.

[39] Nguyen, T. D; Kan, M.-Y. Keyphrase extraction in scientific publications. In *Proceedings of ICADL 2007*, pages 317-326, Hanoi, Vietnam, 2007.

[40] Ohsawa, Y; Benson, N. E; Yachida, M. KeyGraph: Automatic Indexing by Co-Occurrence Graph Based on Building Construction Metaphor. In *Proceedings of the Advances in Digital Libraries Conference*, ADL '98, pages 12, Washington, DC, USA, 1998.

[41] Palshikar, G. K. Keyword Extraction from a Single Document Using Centrality Measures. *Pattern Recognition and Machine Intelligence*, LNCS 4815, pages 503-510, 2007.

[42] Paralič, J; Marek P. Some approaches to text mining and their potential for semantic web applications. *Journal of Information and Organizational Sciences*, 31(1):157-169, 2007.

[43] Pasquier, C. Single Document Keyphrase Extraction Using Sentence Clustering and Latent Dirichlet Allocation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 154-157, Los Angeles, California 2010.

[44] Pudota, N; Dattolo, A; Baruzzo, A; Tasso, C. A New Domain Independent Keyphrase Extraction System. *Digital Libraries*, CCIS 2010, 91: 67-78, 2010.

[45] Saratlija, J; Šnajder, J; Dalbelo-Bašić, B. Unsupervised topic-oriented keyphrase extraction and its application to Croatian. *Text, Speech and Dialogue*, LNCS 6836, pages 340-347, 2011.

[46] Sebastiani, F. Machine learning in automated text categorisation. *ACM Computing Survays*, 34(1):1-47, 2002.

[47] Šišović, S; Martinčić-Ipšić, S; Meštrović, A. Toward Network-based Keyword Extraction from Multitopic Web Documents. In *Proceedings of 6th International Conference on Information Technologies and Information Society (ITIS2014)*, Šmarješke toplice, pages 18-27, Slovenia, 2014.

[48] Solé, R. V; Corominas, B; Valverde, S; Steels, L. Language networks: Their structure, function, and evolution. *Complexity*, 15(6):20-26, 2010.

[49] Sonawane, S. S; Kulkarni, P. A; Graph based Representation and Analysis of Text Document: A Survey of Techniques. *International Journal of Computer Applications*, 96(19):1-8, 2014.

[50] Song, M; Song, I.-Y; Hu, X. KPSpotter: a flexible information gain-based keyphrase extraction system. In *Proceedings of 5th International Workshop of WIDM 2003*, pages 50-53, New Orleans, Louisiana, USA, 2003.

[51] Tsatsaronis, G; Varlamis, I; Nørvag, K. SemanticRank: ranking keywords and sentences using semantic graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1074-1082, Beijing, China, 2010.

[52] Turney, P. D. Coherent Keyphrase Extraction via Web Mining. In *Proceedings of the 18th International Joint Conference on AI*, IJCAI'03, pages 434-439, San Francisco, CA, USA, 2003.

[53] Turney, P. D. Learning to Extract Keyphrases from Text. In *Technical Report ERB-1057*, National Research Council of Canada, Institute for Information Technology, 1999.

[54] Wan, X; Xiao, J. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 855-860, Chicago, Illinois, 2008.

[55] Wang, J; Peng, H; Hu, J.-S. Automatic Keyphrases Extraction from Document Using Neural Network. In *Advances in Machine Learning and Cybernetics*, 4th International Conference ICMLC 2005, LNCS 3930, pages 633-641, Guangzhou, China, 2006.

[56] Washio, T; Motoda, H. State of the Art of Graph-based Data Mining. *SIGKDD Explorations Newsletter*, 5(1):59-68, 2003.

[57] Witten, I. H; Paynter, G. W; Frank, E; Gutwin, C; Nevill-Manning, C. G. Kea: Practical Automatic Keyphrase Extraction. In *Proceedings of the 4th ACM*

Conference of the Digital Libraries, DL ’99, pages 254-255, Berkeley, CA, USA, 1999.

[58] Wu, J-L; Agogino, A. M; Automating Keyphrase Extraction with Multi-Objective Genetic Algorithms, In *Proceedings of the 37th HICSS*, IEEE, 4(4):104-111, 2003.

[59] Xie, Z. Centrality Measures in Text Mining: Prediction of Noun Phrases that Appear in Abstracts. In *Proceedings of the ACL Student Research Workshop*, ACLstudent ’05, pages 103-108, Ann Arbor, Michigan, 2005.

[60] Yang, Z; Lei, J; Fan, K; Lai, Y. Keyword extraction by entropy difference between the intrinsic and extrinsic mode. *Physica A: Statistical Mechanics and its Applications*, 392(19):4523-4531, 2013.

[61] Zhang, C; Wang, H; Liu, Y; Wu, D; Liao, Y; Wang, B. Automatic Keyword Extraction from Documents Using Conditional Random Fields. In *Journal of Computational Information Systems*, 4(3):1169-1180, 2008.

[62] Zhang, K; Xu, H; Tang, J; Li, J. Keyword Extraction Using Support Vector Machine. *Advances in Web-Age Information Management*, LNCS 4016, pages 85-96, Hong Kong, China, 2006.

[63] Zhang, Y; Milios, E; Zincir-Heywood, N. A Comparison of Keyword- and Keyterm-based Methods for Automatic Web Site Summarization. In *Technical Report: Papers from the AAAI’04 Adaptive Text Extraction and Mining*, pages 15-20, San Jose, 2004.

[64] Zhou, Z; Zou, X; Lv, X; Hu, J. Research on Weighted Complex Network Based Keywords Extraction. *Chinese Lexical Semantics*, LNCS 8229, pages 442-452, 2013.