Keyphrases extraction from single textual documents based on semantically defined background knowledge and co-occurrence graphs

Mauro Dalle Lucca Tosi[0000−0002−0218−2413] and
Julio Cesar dos Reis[0000−0002−9545−2098]

Institute of Computing, University of Campinas, Campinas - SP, Brazil
maurodlt@hotmail.com and jreis@ic.unicamp.br

Abstract. Keyphrases are short phrases composed of one or more words used to best represent a textual document and its main topics. They are fundamental to assist publishers in indexing documents and readers in identifying the most relevant ones. The keyphrase extraction task is a fundamental and challenging task designed to automatically extract a set of keyphrases from textual documents. In this article, we extend our research on C-Rank, an unsupervised approach that automatically extracts keyphrases from single documents. C-Rank uses a concept-linking approach that links concepts in common between single documents and an external background knowledge. It uses those concepts as candidate keyphrases, which are modeled in a co-occurrence graph, from which keyphrases are extracted based on heuristics and their centrality in the graph. We advance our study over C-Rank by evaluating it using different concept-linking approaches - Babelfy and DBPedia Spotlight. The evaluation was performed in five gold-standard datasets composed of distinct types of data - academic articles, academic abstracts, and news articles. Compared to other unsupervised approaches that do not demand external data to be provided by the user, C-Rank achieves state-of-the-art results extracting keyphrases from scientific documents.

Keywords: keyphrase extraction · complex networks · semantic annotation.

1 Introduction

Keyphrases are expressions composed of one or more words designed to describe the content of a textual document. They usually encompass the main topics related to the document that they represent. Usually, potential readers use them to judge whether those topics are relevant for them or not, which can lead them to read a document or just ignore it. Keyphrases can be automatically generated or provided by authors, which is not common on non-scientific-related texts. Moreover, they may be used to assist in various NLP tasks as the recommendation of articles to readers, the analyses of research trends over time, among others [2]. However, the automatic extraction of keyphrases is challenging. It is impacted

by the domain of the document analyzed, the lack of context, and the fact that the length of the keyphrases may vary [2]. Therefore, considering the relevance of keyphrases, even with the evolution of the keyphrase extraction algorithms over the last years, their automatic extraction is still a relevant research topic that requires further studies.

One may extract keyphrases based on different approaches. Hasan and Ng [13] divided the keyphrase extraction task in Supervised, that requires an annotated set for training; and Unsupervised, which does not depend on annotated data. In this article, we consider that annotated data is not always available and we further discuss and compare the usage of an unsupervised keyphrase extraction approach.

Unsupervised keyphrase extraction methods may receive distinct inputs, other than the document from which one is trying to extract the keyphrases. These inputs are used as a background data-source from which the method can obtain, usually statistical, information. The background data can be composed of web-pages, specific-domain documents, and general scientific texts [16]. Considering the extra information provided by the background data, unsupervised approaches that use it usually obtain best results compared to other methods. However, they have some drawbacks as requiring training, which can be time and computation consuming; and demanding from the user the input of the background data, which may be unavailable. Therefore, methods that are not based on training nor background data inputted by the user should be explored.

Since background data can improve the results of keyphrase extraction algorithms, a pre-defined-domain-independent knowledge resource could be used to improve the quality of the obtained keyphrases without requesting training nor additional data to be provided by the user. BabelNet [23] is a wide-coverage semantic network that was automatically constructed based on WordNet [21], Wikipedia[1], Wikidata [31], and other sources. It is composed of about 16 million entries, which are called synsets. Based on the amount of data that it encompass, the extraction of information from it would be impaired without Babelfy [22], which is a graph-based solution that simultaneously performs Entity Linking (EL) and Word Sense Disambiguation (WSD) on BabelNet. Another pre-defined-domain-independent knowledge resource is DBPedia [1], an open knowledge graph constructed based on Wikipedia information. It has data in 125 languages and a total of about 3 billion entries. Similar to Babelfy for the BabelNet, DBPedia has the DBPedia Spotlight. It is a tool that annotates and disambiguates mentions of DBPedia resources in natural language texts.

This article proposes a concept linking approach to automatic perform unsupervised keyphrase extractions from free-text documents. This work is an extended version of our previous work that presents C-Rank [30]. C-Rank does not demand training nor other data provided by users as it performs its linkages through a pre-defined-domain-independent knowledge resource. In this article, we further study and compare the usage of two semantic networks and concept linking approaches. We compare the usage of Babelfy, which uses as background

[1] http://www.wikipedia.org

knowledge the BabelNet [23] semantic network; and DBPedia Spotlight, which uses as background knolwedge the DBPedia knowledge graph. Furthermore, we evaluate both C-Rank versions - one using Babelfy and the other using DBPedia Spotlight - in distinct datasets. We extend the evaluation of C-Rank to encompass distinct types of data - academic articles, academic abstracts, and news articles. We attest the effectiveness of C-Rank to extract keyphrases from academic documents by comparing its results with the ones from an updated list of state-of-the-art keyphrase extraction approaches that do not demand external data to be provided by the user.

The remaining of this article is organized as follows: Section 2 presents keyphrase extraction related works. Afterwards, Section 3 presents both versions of C-Rank, a model to automatically extract keyphrases from documents. Section 4 reports on the used benchmark datasets in addition to the achieved results. Whereas Section 5 discusses our findings and compares C-Rank with existing methods, Section 6 concludes the article exhibiting the final considerations.

2 Related Work

This section presents unsupervised keyphrase extraction techniques and compares their approaches to obtain the phrases that best describe the content of a textual document. A survey conducted by Hasan and Ng [13] segmented unsupervised methods in four categories "Graph-based Ranking", that considers the co-occurrence of the phrases in a text as graph edges, in which its vertices represent the keyphrases, that are ranked based on the graph structure; "Topic-Based Clustering", which constructs a graph with the document topics as vertices and its relations as edges, then clusters it to identify the main topics discussed in the analyzed document;"Simultaneous Learning" considers that keyphrase extraction and text summarization tasks can beneficiate from each other and be performed simultaneously. It combines "Graph-based Ranking" with other summarization techniques to improve results; and "Language Modeling", that uses a background textual set to rank the relevance of a phrase in the analysed document, which is then compared with the same metric gathered in the background set.

In addition the four categories, Hasan and Ng also observed that many techniques merge their approaches with heuristics to push forward their results. Table 1 presents some of the best-unsupervised keyphrase extraction techniques, separating them based on their approaches, divided as "Graph-based", consolidating the "Graph-based Ranking" and the "Topic-Based Clustering" in one category; "Language Modeling", englobing the methods that depend on a background textual set; and "Concept linking", our proposed technique that uses a knowledge base to support the identification of keyphrases.

Mihalcea and Tarau [20] presented the TextRank algorithm as a way to represent a text as a graph. First, they tokenize their text and annotate it with part-of-speech tags. Second, Text-Rank creates a syntactic filter and uses the

Table 1. Unsupervised Keyphrase Extraction techniques and their approaches.

Models	Graph-based	Language Modeling	Concept Linking	Year
TextRank [20]	X	-	-	2004
BUAP [25]	X	-	-	2010
DERIUNLP [5]	-	X	-	2010
KP-Miner [10]	-	X	-	2010
Likey [28]	-	X	-	2010
Topic-Rank [8]	X	-	-	2013
SGRank [9]	X	X	-	2015
PositionRank [11]	X	-	-	2017
Shi *et al.* [29]	X	-	X	2017
Boudin [7]	X	-	-	2018
WikiRank [34]	X	-	X	2018
Yeom, Ko, Seo [33]	X	X	-	2019
C-Rank	**X**	**-**	**X**	**2019**

tokens that pass by it as graph vertices, that are connected by undirected and unweighted edges representing their co-occurrence in the text. Third, the technique ranks the graph vertices with a variation of Google's PageRank algorithm [26] and selects the best-ranked as the document keywords. Finally, the algorithm identifies sequences of those tokens in the text and treats them as multi-word keywords, recognized as part of the final result along with the other candidates that are represented by a single token.

On the other hand, instead of constructing a graph with individual words as vertices, Ortiz *et al.* [25] developed *BUAP* that identifies the most frequent sequences of words in a document as vertices of a graph and weights them using the PageRank algorithm. Then, BUAP outputs 15 keyphrases formed by the top-3 multi-term candidates, the top-ranked single-words, and up to 3 of their expanded-forms, if there are acronyms among them.

Bordea and Buitelaar [5] along with El-Beltagy, Samhaa and Rafea [10] respectively introduced the *DERIUNLP* and the *KP-Miner* algorithms. Both are Language Modeling methods using a background textual collection to obtain word statistics applied on a variation of the TF-IDF metric. The first one uses the syntactic description of the words to identify noun phrases and annotate them to one of the manually assigned skill types. Those phrases are considered the candidate keyphrases, ranked based on TF-IDF along with their number of words and frequency in their skill type collection. DERIUNLP then excludes the most general keyphrases, expand their Acronyms and uses a final factor to penalize phrases according to their first appearance in the text.

The second algorithms, *KP-Miner* firstly selects its candidates by extracting phrases used at least "n" times, without punctuation or stop-words in it and had their first appearance within a pre-defined threshold part of the text. Then it obtains the candidate's weights applying a TF-IDF variation, also considering as factors each phrase number of words and the location of their first appearance. At the final stage, KP-Miner re-weights the keyphrases after reducing the number

of times the most important phrases appeared in the text based on if they are sub-phrases of other relevant candidates, dealing with their overlap.

Different from the other mentioned Language Modeling approaches, Paukkeri and Honkela [28] presented *Likey*, that instead of a TF-IDF variation uses a rank metric to weight candidates, that is based on the number of phrases with the candidate length, and normalized regarding the same metric applied on a reference corpus. Then, keyphrases with more than a word face a post-processing step that excludes candidates with any term ranked with a value smaller than a pre-defined threshold to remove stop-words.

Bougouin, Boudin and Daille [8] developed the *TopicRank* algorithm. First, it tokenizes, part-of-speech tags the text and clusters it into topics, weighted with the same ranking algorithm used in Text-Rank [20]. In the end, *TopicRank* outputs the most common keyphrases of the principal topics as a result. On the other hand, Danesh, Sumner and Martin [9] combined the "Graph-based" and the "Language Modeling" approaches to create the SGRank, which extracts all possible candidate phrases from the text and excludes the ones containing: punctuations, stop-words, a frequency below a pre-defined threshold, and any word that is not a noun, verb or adjective. Next, it calculates a TF-IDF variation of the candidates, which considers as 1 the IDF value of all phrases compounded by more than a word, because of the TF-IDF inaccuracy when applied to those terms. Then, the top candidates are re-ranked based on their length and where they appeared at first in the document. Finally, SGRank dismisses the key phrases under a threshold and builds a co-occurrence graph with the final candidates as vertices, weighted based on a modified version of the PageRank [26] algorithm and outputted as a result.

Using only the content of a single document to extract its keyphrases, Florescu and Caragea [11] proposed the PositionRank algorithm. It is similar to the TextRank, but it uses a biased version of the PageRank that considers the position and the frequency of the words' occurrences in the text to improve its results. Similarly, Boudin [7] introduces a TopicRank-based approach that also uses the position of the words in the text to improve the quality of its keyphrases. It promotes the first occurring candidate of each topic. By doing this, they improved the state-of-the-art of the keyphrases extraction using only the content of a single document.

Shi *et al.* [29] proposed to construct knowledge graphs to represent single documents in order to extract their keyphrases. Different from graph-based co-occurrence methods or statistic-based methods, their approach considers semantic relations among concepts. Shi *et al.* [29] also link their keyterm candidates with entities from a pre-defined background knowledge-graph (DBPedia). After this linkage, they construct, cluster, and rank a keyterm graph, from which they extract the document's keyphrases. By their experiments, this approach improved the state of the art of the unsupervised keyphrase extraction of single documents. Likewise, Yu, Ng [34] introduced WikiRank. It is an algorithm that also uses pre-defined background knowledge to assist in the extraction of keyphrases. Instead of relying on a compendium of documents inputted by the

user (language-modeling approaches) or only on the content of the article itself (simple graph-based approaches), it extracts keyphrases based on concepts in common between the document analyzed and Wikipedia. WikiRank uses the identified concepts to construct a semantic graph, it them applies some heuristics, and extracts the document's keyphrases.

Yeom, Ko, and Seo [33] proposed a keyphrase extraction algorithm to overcome the limitations faced by statistical models in extracting keyphrases from single documents; and the drawbacks of some graph-based approaches, which considers only a small amount of information contained in single documents. They propose to merge those approaches, modeling a document as a graph, but scoring its keyphrases considering a statistical model, a modified version of the C-value model. This approach, similarly to the ones present by Florescu and Caragea [11] and Boudin [7], uses the occurrence position of the words in the text to weight the keyphrase candidates. However, instead of using the occurrence position from the analyzed document only, it considers the occurrence position of words in a background compendium as well.

Besides the discussed keyphrase extraction approaches presented above, Mahata *et al.* [17] introduce Key2Vec, a novel technique that surpassed the results achieved by all the other unsupervised methods. Key2Vec represents words as dense real-valued vectors, that need training in their phrase embedding model, executed using over than 1 million documents. Therefore, despite the great results it suffers from both problems priory discussed, requiring time and data for their training.

The presented approach in this paper, named C-Rank, different from other state-of-the-art unsupervised keyphrase extraction approaches, does not request further data to be provided by users. It relies on pre-defined background knowledge to leverage the meaning of terms in the extraction process. Instead of gather knowledge from statistical techniques, which demand a compendium that encompass domain knowledge, C-Rank, similarly to Shi *et al.* [29], extracts information from a wide-coverage semantic network, but different from them, do not cluster its concepts in topics to generate its keyphrases.

3 C-Rank

C-Rank is an unsupervised algorithm that automatically extracts keyphrases from single documents without the support of a background textual collection. In this sense, the user needs to insert only the text from which the system must extract the keyphrases. It combines the knowledge of the document itself and contained inside a wide-coverage semantic network. C-Rank works in three stages illustrated in Figure 1, and detailed in the following subsections. The first stage pre-processes the input document and annotate it with pre-defined semantic network concepts. The second stage takes those concepts as vertices and generates a co-occurrence graph, which is ranked and trimmed based on heuristics and its centrality, producing candidate keyphrases. The third and final

stage identifies candidates that belong to the same phrase, which are merged and re-ranked, generating the final keyphrases list as output.

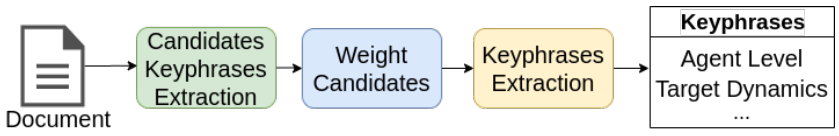

Fig. 1. C-Rank stages. Adapted from [30].

3.1 Extraction of Candidate Keyphrases

The first stage receives as input a textual document that is initially parsed to have its concepts linked with a pre-defined semantic network, process named here *concept linking*. This process results in a set of paragraphs annotated with disambiguated concepts as illustrated in Figure 2.

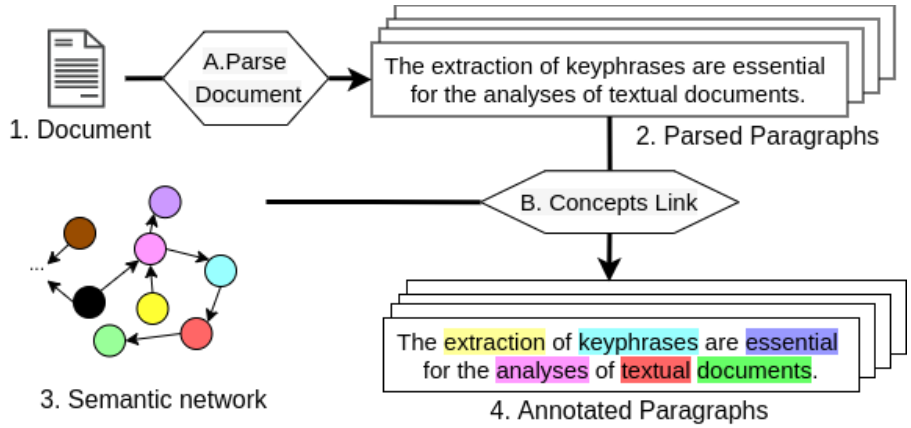

Fig. 2. C-Rank First Stage: Extraction of Candidate Keyphrases.

We studied the usage of two well-known semantic networks in the concept linking process, BabelNet [23] and DBPedia [1]. We chose those networks for two main reasons. First, their generality, which let them encompass knowledge from various distinct domains, not limiting their usage only for content of specific areas. Second, the simplicity of using their solutions to identify correspondent concepts between natural language texts and their networks.

Babelnet[2] [23] is a wide-coverage multilingual semantic network automatically constructed that has about 16 million entries, which are called babel

[2] https://babelnet.org/

synsets. Each synset represents a given concept or a named entity and contains all its synonyms and translations in different languages. Although the amount of relevant information contained in Babelnet, its usage would be limited without the Babelfy[3] [22], which is their graph-based solution to perform simultaneously Entity Linking (EL) and Word Sense Disambiguation (WSD) on Babelnet. The Babelfy description mentions it as an EL and a WSD solution, even though it also performs concept linking, that is fundamental to C-Rank as many keyphrases are not formed by named entities.

DBPedia[4] [1] is an open knowledge graph (OKG) constructed based on the Wikipedia information. It is composed of data in 125 languages, reaching about 3 billion entries. Similarly to Babelfy for BabelNet, DBPedia has the DBpedia Spotlight[5] [19], which automatically disambiguates and annotates mentions of DBPedia resources (correspondent concepts) in unstructured texts.

Babelfy and DBPedia Spotlight are the adopted approaches that we attest in respectively linking the document concepts with Babelnet and DBPedia, semantic annotating them. Both of them have HTTP APIs, which allow to perform the concept linking process without downloading their database. However, despite receiving whole texts to process and annotate, some constraints occurred during the use of the HTTP APIs, as a maximum length of the input text and their inability to process some special characters.

In order to overcome these limitations, we parsed the input document - process A, Figure 2 - segmenting it in sets of paragraphs with at most 5000 words, and removing all non-letter characters, except for “!”, “.”, “?”, “-” and “ ”, which are important as they segment sentences and words.

Afterward, the process B - Figure 2 - links the parsed paragraphs using Babelfy or DBPedia Spotlight, which identifies correspondent concepts between the parsed paragraphs and the chosen semantic network. It can also determine multi-word concepts and its sub-concepts. For example, in “semantic network” the following synsets can be linked “semantic”, “network”, and “semantic network”. In our approach, we only use the multi-word concept annotation because the sub-concepts would always appear more in the document and they would be positively biased in C-Rank next stages.

3.2 Weight Candidates

The second stage receives the annotated paragraphs as input, generating a weighted directed co-occurrence graph based on it. This is ranked and trimmed with heuristics to output a graph of candidate keyphrases (*cf.* Figure 3).

In order to construct the graph, process C (in Figure 3) uses the paragraphs linked concepts as vertices and their co-occurrence to generate the direct edges, that connects directly subsequent vertices represented as solid arrows in Figure 3; and indirect ones, linking concepts within a predefined window width, which are

[3] http://babelfy.org/
[4] https://wiki.dbpedia.org/
[5] https://www.dbpedia-spotlight.org/

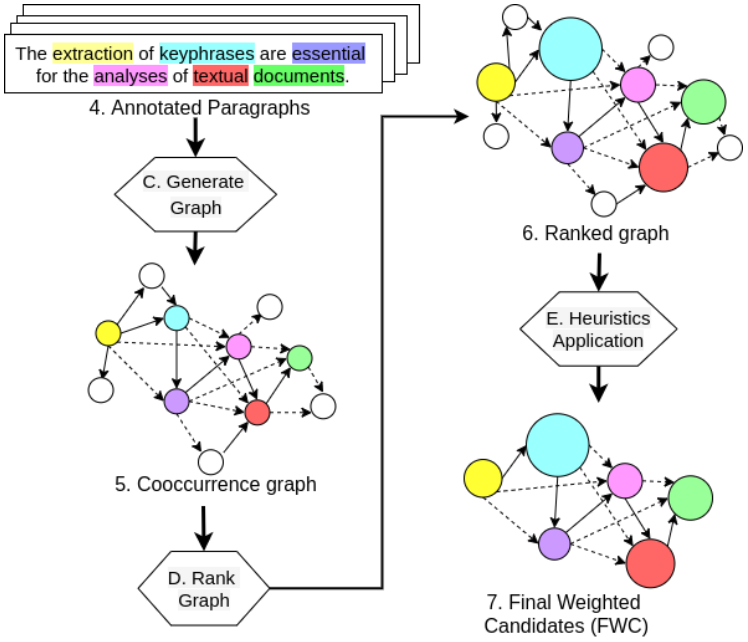

Fig. 3. C-Rank Second Stage: Weight Candidates.

represented by dotted arrows. The window width is also referred as co-occurrence window and is further explored in Section 4.3, .

The graph has their vertices weighted based on the number of times their concepts appear inside the document. This also occurs with the edges, that are weighted based on the distances between the concepts which its vertices represent (*cf.* Equation 1), in which $weightEdge_{i,j}$ is the weight of the edge that connects vertice i with vertice j; $In(j)$ refers to the set of edges that arrive at j; $window$ is the predefined window width; and $distance(i, j)$ stands for the co-occurrence distance in text between the concepts that the vertices i and j represent.

$$weightEdge_{i,j} = \sum_{i \in In(j)} 1 - log_{window} distance(i, j) \tag{1}$$

Process D (in Figure 3) ranks the vertices of the co-occurrence graph to obtain the candidate keyphrases. It uses the centrality degree value normalized by the maximum possible degree of a node. Although being a simple measure, the degree centrality achieves higher results in the identification of keyphrase on graph-based approaches, compared to other traditional ranking techniques [3].

C-Rank second stage also applies four heuristics into the graph, which were studied on a training set and are analyzed in Section 4.3. However, one must previously determine how to label each concept of the graph before applying the heuristics, because the same idea can be expressed divergently. As an example, “Artificial Intelligence” and “AI”, both represent the same concept, despite being

written differently. We label each concept based on its first occurrence in the document because we understand that it may cover the concept extended form instead of its initials, considering that usually, in a text, a concept is introduced before its abbreviation.

The first heuristic identifies the Part-of-speech (POS) of each candidate label and discards those that have any word different from a noun, a verb or an adjective, which are the most common keyphrase POS tags. The second one cuts the 87% lower-ranked candidates (LRC) if the analyzed document is long - has more than 1000 words - in order to reduce noise. The third heuristic re-ranks the candidates favoring those formed by multiple words, as they are more likely to be chosen to become keyphrases; it uses the Equation 2, in which c_w represents the candidate weight and $len(c)$ its number of words. The fourth and final heuristic discards all candidates that first appeared after a *CutOff* threshold of the text, defined to 18% for long documents, as keyphrases usually are introduced at the beginning of a text.

$$c_w = c_w^{\frac{1}{len(c)}} \tag{2}$$

3.3 Keyphrase Extraction

C-Rank third stage (Figure 4) receives the Final Weighted Candidates graph (FWC), identifies the concepts that belong together in the same keyphrase and outputs a re-ranked list of the input document keyphrases.

A "Compound Candidates" is the given definition of the candidates that belong to the same keyphrase, which are formed by the union of two different concepts. As in Figure 4, the vertices with dotted patterns, which are labeled as "agent" and "level", despite being subsequently in the text, representing a single thought, were linked separately in stage one, Figure 2. The compound candidates identification mitigates this issue and merges these concepts together, allowing them to be multi-word concepts.

The compound candidates identification considers the vertices relations in the graph to determine whether two terms represent a single concept and, therefore, belong together in the same keyphrase. To conclude that two candidates are compound, their subsequent co-occurrence must occur multiple times, which will vary depending on the text length. Therefore, compound candidates must be linked through a direct edge weighting at least $2+(totalWords_d/1000)$, being $totalWords_d$ the number of words in the document d. This minimum weight ensures that the compound candidates appear at least twice in short texts and require a higher frequency in larger ones.

Next, the third stage weights the compound candidates to have a comparison metric to re-rank them based on the other candidate keyphrases. Process F in Figure 4, calculates the normalized edges weight that connects the compound candidates (*cf.* Equation 3), in which $NE_{i,j}$ refers to the normalized edge that links vertice i with vertice j; $w(Out(i))$ is the sum of all edges weights outgoing vertice i; and $w(In(j))$ is the sum of all edges weights incoming vertice j. Then,

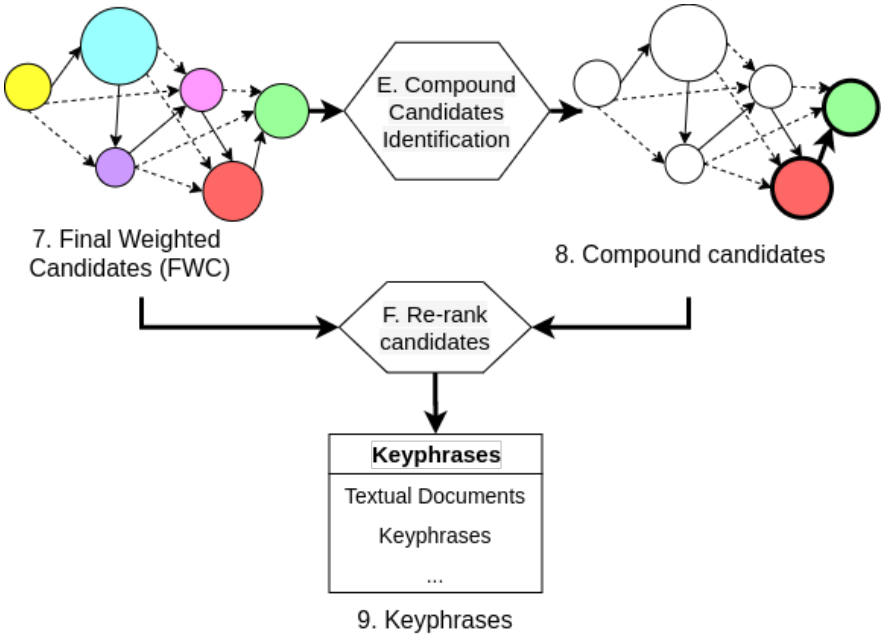

Fig. 4. C-Rank Third Stage: Keyphrase Extraction.

Process F calculates the ranking weight of the compound candidates as expressed by Equation 4. It has $CC_{i,j}$ as the ranking value of the compound candidate formed by the vertices i and j, and v_i, v_j are the weight of the vertices i and j from the FWC. At last step, Process F normalizes the compound candidates by their sum as presented by Equation 5, in which $NCC_{i,j}$ is the normalized $CC_{i,j}$; and outputs a sorted list of the input document keyphrases $Keyphrases_d$, generated by the union of the Final Weighted Candidates and the top-ranked Compound Candidates. However, the NCC values difference decrease because of the normalization and after its union with the FWC they tend to cluster, standing out over the rest of the data. To overcome this issue, we join only 6 top-ranked Compound Candidates with the Final Weighted Candidates, a value defined from observations and tests over the keyphrases data-sets, thus $Keyphrases_d = FWC \cup NCC_{1:6}$.

$$NE_{i,j} = \frac{indirectEdge_{i,j}}{w(Out(i)) + w(In(j))} \tag{3}$$

$$CC_{i,j} = (v_i + v_j) * (\frac{NE_{i,j}}{\sum_{k \in NE} k}) \tag{4}$$

$$NCC_{i,j} = \frac{CC_{i,j}}{\sum_{t \in NE} t} \tag{5}$$

4 Experimental Evaluation

This section presents the analysis performed for C-Rank development and its evaluation, along with the protocols utilized during the corresponding experiments. We first introduce the used datasets, then report on the refinement performed in the heuristics values followed by the achieved results with *C-Rank* compared to other unsupervised keyphrase extraction techniques discussed in Section 2.

4.1 Datasets

We use five standard benchmark datasets to evaluate the results achieved by C-Rank and compare them with other keyphrase extraction approaches, which explored the same datasets. We chose those datasets because of their co-occurrence among most of the compared keyphrase extraction techniques. Furthermore, we selected datasets that could express how C-Rank deals with different types of textual entries: Full academic articles, academic abstracts, and news articles.

The first is the SemEval2010 [16] dataset, divided in a trial, a train and a test set containing 40, 100 and 144 documents, respectively. Each one is an academic article belonging to one of four distinct ACM classifications. All the records are annotated with two sets of keyphrases as the author-assigned, which were part of the original document; and the reader-assigned, that were manually annotated by Computer Science students.

NUS [24] is the second dataset. It is composed of 211 full academic articles in its test set, which is the one used in our analysis. Similarly to the SemEval2010, it also has its keyphrases divided in author-assigned and readers-assigned. In our experiment, we use both sets of keyphrases to determine our accuracy, recall, and f-score.

The INSPEC [14] is the third dataset, composed of 3 sets of documents, a training set containing 3,000 text files, a validation set with 1,500 and a test set consisting of 500. Despite having a similar number of keyphrases assigned per document, the INSPEC dataset, different from the SemEval and the NUS datasets, is composed of academic abstracts, which makes its files shorter. Each of its records is annotated with controlled and uncontrolled keywords, even though only the last ones are usually used during evaluations because they have more keyphrases inside the texts, 76.2% against 18.1%.

DUC [32] is our fourth dataset, which is composed by 308 news articles. This dataset has only one set of keyphrases, which were annotated by readers.

At last, 500N-KPCrowd [18] is our fifth dataset, also composed by news articles. It has 500 documents, divided for training and test. Furthermore, its documents were annotated with keyphrases generated by readers. We note that, in our evaluation with 500N-KPCrowd, we use only its test set, composed of 50 news articles.

Table 2 exhibits statistics of the datasets described above. The SemEval and the NUS datasets are similar, with academic articles with around 8000 words each, annotated with in average 11 and 14.7 keyphrases respectively. The

INSPEC dataset is composed of academic abstracts with in average 135 words each. Even though, the average number of keyphrases per abstract is 9.8, just slightly smaller than the other sets composed of full articles. Then, we present the DUC and KPCrowd news sets. Despite having the same entry type (news documents), they are very distinct. DUC has documents with almost two times the number of words of KPCrowd. On the other hand, KPCrowd's documents have on average 46.2 keyphrases each, which is extremely high compared to DUC, with 8.1, and the other sets.

Table 2. Datasets statistics. It exhibits the type of documents of each dataset, the number of documents in the sets that we used in our evaluation, and the average number of words and keyphrases of the documents of each dataset.

Dataset	**Entry**	**#documents**	**#words**	**#keyphrases**
SemEval2010	academic articles	100	7961.2	14.7
NUS		211	8398.3	11.0
INSPEC	academic abstracts	500	134.6	9.8
DUC	news	308	847.2	8.1
500N-KPCrowd		50	465.3	46.2

At last, we note that achieving 100% of f-score in those datasets is practically impossible. Take as example the SemEval dataset, 15% of the reader-assigned keyphrases are not in their original composition, against 19% of the author-assigned set. Therefore, considering the orientation of the dataset developers of extracting only keyphrases that actually appear in the text, the maximum recall one can achieve in this dataset is 81% for the author's and 85% for the reader's keyphrases. Another example is the INSPEC, as previously mentioned, only 76.2% of its controlled keywords appear in the texts.

4.2 Implementation Aspects

For the C-Rank development several technologies were adopted to favor the algorithms reproducibility and simplicity. C-Rank was developed on python and is available online[6].

It uses networkx [12], a software package designed to study complex networks, which builds, manages and calculates the centrality of the algorithm graphs. Those graphs apply concepts as vertices, which can be the babel synsets: gathered using the pybabelfy[7], an interface to use the Babelfy HTTP API on python; or DBPedia Spotlight concepts: gathered using the DBpedia Spotlight HTTP API directly. Then, to identify the POS tags of each babel synset, the NLTK [4] was used, as it is a toolkit designed to process natural language.

[6] Omitted due to blind review
[7] https://github.com/aghie/pybabelfy

The C-Rank evaluation was performed using the Python stemmer available on the same website[8] as the one used on the SemEval 2010 Task 5 [15], modified to work on Python 3. It is a Porter Stemmer that produces slightly different results than the NLTK implementation. The SemEval results were calculated using the stemmed words and the Perl code provided by SemEval 2010 Task 5[9], whether the NUS, INSPEC, DUC, and KPCrowd results were determined with the implementation in python available along with C-Rank and the pre-processed datasets made available by Boudin[10].

4.3 Parameters Refinement

We defined several variables in the C-Rank development. To determine the best possible values for those variables, we performed different analyses considering the Babelfy concept-linking approach and the SemEval training set, which lead us to our defined heuristics and parameters. These were explored to evaluate *C-Rank* in the test set of the datasets (*cf.* Subsection 4.4). The following experiments present the results concerning the refinement of parameters, showing C-Rank f-score on the SemEval training set varying the variables, their values, and the number of analyzed keyphrases.

The co-occurrence window, or predefined window width, was the first variable defined in C-Rank development. Table 3 shows the results variance when the co-occurrence window changes and highlight in bold its optimal value.

Table 3. Micro-average F-scores achieved extracting the Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 trainning dataset varying the graph co-occurrence window.

Co-occurrence Window	**Top-5(%)**	**Top-10(%)**	**Top-15(%)**	**Average(%)**
2	15,23	19,98	20,49	18,6
3	**15,83**	**20,58**	**20,76**	**19,1**
4	15,83	20,48	20,81	19,0
5	15,83	20,53	20,95	19,1
10	15,63	20,26	20,95	18,9
100	14,96	19,82	21,08	18,6

Two heuristic variables values were defined, LRC and *CutOff* Threshold. Both of them have their variations exhibited along with their results change in Table 4 and Table 5. The highlighted numbers point out the optimal values which are used in the course of this article and the last line of each table presents the C-Rank results without the usage of the corresponding heuristic, when LRC is set to 0 and $CutOff$ to 100.

[8] https://tartarus.org/martin/PorterStemmer/
[9] http://semeval2.fbk.eu/semeval2.php?location=tasks#T6
[10] https://github.com/boudinfl/ake-datasets

Table 4. Micro-average F-scores achieved extracting the Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 dataset varing the percentage of *LRC*.

LRC **(%)**	**Top-5 (%)**	**Top-10(%)**	**Top-15(%)**	**Average(%)**
95	15,09	18,02	18,34	17,2
90	15,36	19,82	20,4	18,5
89	15,83	20,26	20,63	18,9
88	16,04	20,58	20,72	19,1
87	**15,83**	**20,58**	**20,76**	**19,1**
86	15,91	20,31	20,72	19,0
85	15,97	20,37	21,04	19,1
84	15,83	20,26	21,22	19,1
83	15,56	20,37	21,22	19,1
75	15,36	19,82	21,36	18,8
0	13,93	18,18	20,03	17,4

Table 5. Micro-average F-scores achieved extracting the Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 dataset varing the *CutOff* Threshold percentage.

CutOff **Threshold (%)**	**Top-5 (%)**	**Top-10 (%)**	**Top-15 (%)**	**Average (%)**
5	14,68	19,98	20,53	18,4
10	15,15	20,64	21,08	19,0
16	15,63	20,42	20,81	19,0
17	15,77	20,64	20,86	19,1
18	**15,83**	**20,58**	**20,76**	**19,1**
19	15,63	20,58	20,72	19,0
20	15,63	20,42	20,72	18,9
25	15,63	19,71	20,44	18,6
50	14,68	18,73	19,67	17,7
100	14,2	18,29	19,71	17,4

Another analyzed C-Rank parameter was the centrality measure used to rank the co-occurrence graph. Table 6 shows the results when this metric varies. In order to evaluate the Concept Linking usage and the proposed heuristics, Table 7 exhibits the variances of results applying our defined contributions. It clearly shows the improvement achieved when implementing the proposed techniques of concept linking and our elaborated heuristics.

4.4 Results

A final evaluation was performed to quantify the effectiveness of C-Rank using both proposed concept-linking approaches. We perform this evaluation by comparing the results of using Babelfy and DBPedia Spotlight in the SemEval, NUS, INSPEC, DUC, and KPCrowd test datasets. Furthermore, those results allow us to compare *C-Rank* with other keyphrase extraction techniques. We present the results as follows. Table 8 shows the results using Babelfy as concept-linking approach in the mentioned datasets; Table 9 presents the obtained results using

Table 6. Micro-average F-scores achieved extracting the Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 trainning dataset varying the co-occurrence graph centrality measure.

Centrality measures	**Top-5**	**Top-10**	**Top-15**	**Average**
Closeness Centrality	15,29	18,95	19,62	18,0
Betweenness Centrality	15,36	19,49	20,76	18,5
Eigenvector	12,37	15,95	17,43	15,3
Pagerank	15,83	19,87	20,58	18,8
Degree Centrality	**15,83**	**20,58**	**20,76**	**19,1**

Table 7. Micro-average F-scores achieved extracting the Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 trainning dataset applying our proposed contributions.

Contributions	**Top-5**	**Top-10**	**Top-15**	**Average**
Using Concept Linking & Heuristics	**15,83**	**20,58**	**20,76**	**19,1**
Using only Concept Linking	8,56	11,58	12,69	10,9
Using only Heuristics	11,15	14,58	15,42	13,7
Without Concept Linking & Heuristics	7,07	9,55	10,68	9,1

DBPedia Spotlight as concept-linking approach in the same datasets; Table 10 shows the comparison of C-Rank with other unsupervised keyphrases extraction techniques that do not demand external data to be inputted by the user. The F-score values we use in our comparison are the ones observed by Papagiannopoulou and Tsoumakas [27] or, if not available, the ones described in the articles that respectively introduce the analyzed techniques.

Table 8. Micro-average precision, recall and f-score on the extraction of Top-5, Top-10, and Top-15 keyphrases on the SemEval2010, NUS, Inspec, DUC, and KPCrowd test datasets using Babelfy as concept-linking approach.

#Keyphrases	**Evaluation**	**SemEval**	**INSPEC**	**DUC**	**KPCrowd**	**NUS**
5	Precision	28,00	31,48	9,48	30,80	16,02
	Recall	9,55	16,02	5,91	3,33	7,23
	F-score	**14,24**	**21,23**	**7,31**	**6,02**	**10,00**
10	Precision	24,20	23,08	8,29	36,20	13,98
	Recall	16,51	23,49	10,21	7,83	12,63
	F-score	**19,63**	**23,28**	**9,15**	**12,88**	**13,27**
15	Precision	20,27	17,05	6,77	36,93	12,57
	Recall	20,74	25,93	12,46	11,99	17,04
	F-score	**20,50**	**20,57**	**8,77**	**18,10**	**14,47**

Table 9. Micro-average precision, recall and f-score on the extraction of Top-5, Top-10, and Top-15 keyphrases on the SemEval2010, NUS, Inspec, DUC, and KPCrowd test datasets using the DBPedia Spotlight as concept-linking approach.

#Keyphrases	Evaluation	SemEval	INSPEC	DUC	KPCrowd	NUS
5	Precison	14,00	25,98	14,54	30,12	14,10
	Recall	4,77	9,85	8,84	3,25	6,04
	F-Score	**7,12**	**14,29**	**11,00**	**5,86**	**8,45**
10	Precison	12,10	22,77	11,43	26,74	11,50
	Recall	8,25	11,52	13,75	5,32	9,85
	F-Score	**9,81**	**15,3**	**12,48**	**8,88**	**10,61**
15	Precison	9,47	22,12	9,83	4,96	9,37
	Recall	9,69	11,74	17,28	21,10	12,03
	F-Score	**9,58**	**15,34**	**12,54**	**10,95**	**10,53**

Table 10. Comparison of f-score achieved by extracting 10 keyphrases on the SemEval2010, NUS, Inspec, DUC, and KPCrowd test datasets.

Method	SemEval	NUS	INSPEC	DUC	KPCrowd
TextRank [20]	5.6	-	12.7	-	-
BUAP [25]	14.4	-	-	-	-
Topic-Rank [8]	13.4	12.6	23.5	-	14.8
PositionRank [11]	13.1	14.6	25.3	-	14.5
Shiet *et al.* [29]	-	-	-	≈33	-
Boudin [7]	14.6	14.7	24.5	-	15.6
WikiRank [34]	-	9.1	27.01	27.53	-
C-Rank (Babelfy)	19.63	13.27	23.28	9.15	12.88
C-Rank (Spotlight)	9.81	10.61	15.30	12.48	8.88

5 Discussion

In this article, we extended our research on the C-Rank algorithm, which performs unsupervised keyphrase extraction without relying on background data inserted by users. We studied how the usage of distinct background semantic networks in the concept-linking approach and the applied heuristics can assist in the identification of better concepts to represent textual documents. Moreover, we compared the obtained results with algorithms with similar characteristics.

Our approach explored external background knowledge from Babelnet and DBPedia. We found a relevant impact with the use of the concept linking in the keyphrase extraction (*cf.* Table 7). We observed that, comparing Tables 8 and 9, the background semantic network used in the concept-linking process directly impacts the results obtained by our algorithm. Using the BabelNet network we achieved the best results in most datasets. This could be impacted by the fact of we setting the heuristics values based on observations performed using only

the SemEval dataset. The obtained results are justified by fact that, compared to DBPedia Spotlight, Babelfy returns significant more correspondent concepts between the textual documents and the semantic network. In some cases, the DBPedia Spotlight identified less concepts than the number of keyphrases we were extracting, negatively impacting its performance.

During the graph construction, the results varying the maximum co-occurrence distances between concepts (*cf.* Table 3) showed that the variance between results is low. Therefore, if performance is an issue, despite the lower f-scores, setting the co-occurrence window to 2 is equivalent of using only the direct-edges during all the algorithm, which would decrease the computational cost without much impact in the resultant values.

Despite the variance of results, the heuristic values do not impact the algorithm effectiveness. The centrality measure, on the other hand, can significantly decrease the results. As demonstrated by Boudin [6] and corroborated in Table 6, despite being simple, the degree centrality achieves higher results than other popular metrics usually used in keyphrase extraction algorithms, as the Pagerank.

Shiet *et al.* [29] previously introduced a method similar to C-Rank, also using a background semantic network to assist in the keyphrase extraction. Despite their state-of-the-art results, their evaluation was performed using only the DUC dataset. Based on Table 10, we observed that the methods that achieve the best results in one dataset not necessarily achieve results as good as that in others. As an example, the WikiRank algorithm achieves the best F-score in the INSPEC and the second best in the DUC datasets. However, it had the worst F-score in the NUS dataset. Therefore, based on the Shiet *et al.* evaluation, we cannot attest their method excellence in extracting keyphrases from documents that are not similar to the ones used in the DUC dataset (news articles with around 850 words each).

For the best of our knowledge, using Babelfy as concept-linking approach, the algorithms not relying on user data and yielding the best results were outperformed by C-Rank with statistical significance in the SemEval 2010 dataset. Furthermore, C-Rank obtained the best average F-score considering the datasets composed of scientific documents (SemEval, NUS, and INSPEC). On the other hand, C-Rank results on the datasets composed of news articles were not satisfactory compared to other methods. Our first concern was about the reduced size of news documents compared to academic ones. As C-Rank obtained competitive results in the INSPEC datasets - the smallest dataset - we discarded this possibility. Then, we address the main part of this difference by a possible variation between the usual layout of news articles compared to academic ones.

The Concept Linking approach is a novel aspect of our algorithm that might be further explored in the keyphrase extraction and other NLP tasks. It not just brings background knowledge that assists in the keyphrases identification, but further produces intermediate structures with concepts and entities semantically annotated that can be used to improve domain understatement and enrich other textual representation structures. Considering the achieved results, further inves-

tigations on the use of concept linking in related NLP tasks could support and complement current solutions. Additionally, as future work, we plan to study how to use semantic relations among concepts to enrich C-Rank graphs and assist in identification of keyphrases composed of multiple concepts.

6 Conclusion

The extraction of keyphrases plays an essential role in the interpretation and analyses of textual documents. Most existing approaches rely on external data as input by users, not extracting keyphrases from single textual documents. This external data is usually a background textual set, which can have its documents annotated with their respective keyphrases (used for supervised training), or not (used for language modeling approaches). However, as users may not posses a background textual set related to the documents they want to extract the keyphrases, approaches that extract them from single documents are essential. In this article, we extended the research on *C-Rank*, an unsupervised keyphrase extraction approach that explores concept linking and graph-based techniques to extract keyphrases from single documents. *C-Rank* does not require external data to be inputted by users as it boosts its efficiency linking concepts identified in the input textual document with their correspondents in wide-coverage semantic networks. It uses the correspondent concepts as vertices of a co-occurrence graph, which is ranked based on heuristics and centrality measures, from which it extract the keyphrases. The conducted experiments quantified the efficiency of *C-Rank* in different scenarios. We evaluated the usage of DBPedia and BabelNet as background semantic networks for concept-linking process. We found that the usage of the latter one produced better results when extracting keyphrase from documents of most of the datasets in our evaluation. Comparing *C-Rank* results using datasets composed of different types of documents (academic articles, academic abstracts, and news), we observed that *C-Rank* did not achieve competitive results compared to similar approaches for extracting keyphrases from "news" articles. On the other hand, it outperformed, with statistical significance, all the compared unsupervised techniques that do not demand extra information to be provided by users on the SemEval2010 benchmark dataset. Compared to the other similar approaches, it obtained the best average f-score considering scientific-related datasets (SemEval2010, NUS, and INSPEC). On this basis, we recommend the usage of *C-Rank* to extract keyphrases from single scientific-related documents. As future work, we plan to investigate how the usage of semantic relations from external semantic networks can assist in the automatic identification of keyphrases.

Acknowledgements

This work was financially supported by the São Paulo Research Foundation (FAPESP) (grants #2017/02325-5 and #2013/08293-7)[11] and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)
2. Augenstein, I., Das, M., Riedel, S., Vikraman, L., McCallum, A.: SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 546–555. Association for Computational Linguistics (2017)
3. Beliga, S., Meštrović, A., Martinčić-Ipšić, S.: An overview of graph-based keyword extraction methods and approaches. Journal of information and organizational sciences **39**(1), 1–20 (2015)
4. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
5. Bordea, G., Buitelaar, P.: Deriunlp: A context based approach to automatic keyphrase extraction. In: Proceedings of the 5th international workshop on semantic evaluation. pp. 146–149. Association for Computational Linguistics (2010)
6. Boudin, F.: A comparison of centrality measures for graph-based keyphrase extraction. In: International Joint Conference on NLP. pp. 834–838 (2013)
7. Boudin, F.: Unsupervised keyphrase extraction with multipartite graphs. arXiv preprint arXiv:1803.08721 (2018)
8. Bougouin, A., Boudin, F., Daille, B.: Topicrank: Graph-based topic ranking for keyphrase extraction. In: International Joint Conference on Natural Language Processing (IJCNLP). pp. 543–551 (2013)
9. Danesh, S., Sumner, T., Martin, J.H.: Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In: Conference on lexical and computational semantics. pp. 117–126 (2015)
10. El-Beltagy, S.R., Rafea, A.: Kp-miner: Participation in semeval-2. In: Proceedings of the 5th international workshop on semantic evaluation. pp. 190–193 (2010)
11. Florescu, C., Caragea, C.: Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1105–1115 (2017)
12. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008)
13. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: A survey of the state of the art. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 1: Long Papers). vol. 1, pp. 1262–1273 (2014)

[11] The opinions expressed in here are not necessarily shared by the financial support agency.

14. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on Empirical methods in natural language processing. pp. 216–223. Association for Computational Linguistics (2003)
15. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In: 5th International Workshop on Semantic Evaluation. pp. 21–26 (2010)
16. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: Automatic keyphrase extraction from scientific articles. Language resources and evaluation **47**(3), 723–742 (2013)
17. Mahata, D., Kuriakose, J., Shah, R.R., Zimmermann, R.: Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. vol. 2, pp. 634–639 (2018)
18. Marujo, L., Gershman, A., Carbonell, J., Frederking, R., Neto, J.P.: Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. arXiv preprint arXiv:1306.4886 (2013)
19. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th international conference on semantic systems. pp. 1–8 (2011)
20. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. pp. 404–411 (2004)
21. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to wordnet: An on-line lexical database. International journal of lexicography **3**(4), 235–244 (1990)
22. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Computational Linguistics **2**, 231–244 (2014)
23. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence **193**, 217–250 (2012)
24. Nguyen, T.D., Kan, M.Y.: Keyphrase extraction in scientific publications. In: International conference on Asian digital libraries. pp. 317–326. Springer (2007)
25. Ortiz, R., Pinto, D., Tovar, M., Jiménez-Salazar, H.: Buap: An unsupervised approach to automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th international workshop on semantic evaluation. pp. 174–177. Association for Computational Linguistics (2010)
26. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (November 1999), previous number = SIDL-WP-1999-0120
27. Papagiannopoulou, E., Tsoumakas, G.: A review of keyphrase extraction. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery p. e1339 (2019)
28. Paukkeri, M.S., Honkela, T.: Likey: unsupervised language-independent keyphrase extraction. In: Proceedings of the 5th international workshop on semantic evaluation. pp. 162–165. Association for Computational Linguistics (2010)
29. Shi, W., Zheng, W., Yu, J.X., Cheng, H., Zou, L.: Keyphrase extraction using knowledge graphs. Data Science and Engineering **2**(4), 275–288 (2017)
30. Tosi, M.D.L., dos Reis, J.C.: C-rank: A concept linking approach to unsupervised keyphrase extraction. In: Research Conference on Metadata and Semantics Research. pp. 236–247. Springer (2019)
31. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)

32. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: AAAI. vol. 8, pp. 855–860 (2008)
33. Yeom, H., Ko, Y., Seo, J.: Unsupervised-learning-based keyphrase extraction from a single document by the effective combination of the graph-based model and the modified c-value method. Computer Speech & Language **58**, 304–318 (2019)
34. Yu, Y., Ng, V.: Wikirank: Improving keyphrase extraction based on background knowledge. arXiv preprint arXiv:1803.09000 (2018)