C-Rank: A Concept Linking Approach to Unsupervised Keyphrase Extraction

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. Keyphrase extraction is the task of identifying a set of phrases that best represent a natural language document. It is a fundamental and challenging task that assists publishers to index and recommend relevant documents to readers. In this article, we introduce C-Rank, a novel unsupervised approach to automatically extract keyphrases from single documents by using concept linking. Our method explores Babelfy to identify candidate keyphrases, which are weighted based on heuristics and their centrality inside a co-occurrence graph where keyphrases appear as vertices. It improves the results obtained by graph-based techniques without training nor background data inserted by users. Evaluations are performed on SemEval and INSPEC datasets, producing competitive results with state-of-the-art tools. Furthermore, C-Rank generates intermediate structures with semantically annotated data that can be used to analyze larger textual compendiums, which might improve domain understatement and enrich textual representation methods.

Keywords: keyphrase extraction · complex networks · semantic annotation.

1 Introduction

Keyphrases are expressions intended to represent the content of a document and highlight its main topics. They may be single or multi-termed and may be provided by the author, which is uncommon in most of the non-scientific texts. Keyphrases are used by potential readers to decide whether or not the topics approached in the document are relevant to them. Furthermore, they may be used to recommend articles to readers, analyze research trends over time, among other NLP tasks [1]. However, the automatic keyphrase extraction is a challenging task as it varies from domains, suffers from the lack of context, and its result keyphrases may be formed by multiple words [1]. Therefore, despite the improvements achieved in the last years, it still is an active research topic that deserves further studies.

The keyphrase extraction task can be performed based on different approaches. Hasan and Ng [5] segmented the keyphrase extraction task in Supervised, that demands an annotated training set; and Unsupervised, that does not depend on annotated data, which is the line followed in this article.

The unsupervised methods can be developed to extract the keyphrases of a document based on different inputs other than the document text itself. The background data varies according to the method and can consider web-pages, specific-domain documents and general scientific texts [7]. Although the best results have been achieved by most of the methods using background data, it may demand information, training time or both, that the user does not necessarily possess. Therefore, approaches that do not require training nor other data to be inputted by the user should be investigated.

A predefined-domain-independent knowledge resource could improve the extraction results without requesting further data nor training from users. Babelnet[1] [11] is a wide-coverage multilingual semantic network automatically constructed that has about 16 million entries, which are called synsets. Each synset represents a given concept or a named entity and contains all its synonyms and translations in different languages. Despite the amount of relevant information contained in Babelnet, its usage would be limited without the Babelfy [10], which is a graph-based approach to simultaneously perform Entity Linking (EL) and Word Sense Disambiguation (WSD) on Babelnet.

In this article, we propose C-Rank as a novel approach to automatic perform unsupervised keyphrase extractions from free-text documents. For the best of our knowledge, it is the first method to explore concept linking to improve results in this task. C-Rank does not demand training nor other data provided by the user as it performs its linkages through Babelfy using as resources the BabelNet [11], Wikipedia[2] and WordNet [9] knowledge. C-Rank parses the inputted document text, runs Babelfy and constructs a co-occurrence graph with the annotated concepts as vertices. Next, it weights the vertices using their centrality in the graph, selects the top-ranked as candidates and modifies them using heuristic factors. Finally, C-Rank identifies vertices that belong to the same keyphrase and merge them, re-rank all the candidates and outputs the result. We extensively evaluate our approach with distinct gold standard datasets and demonstrate the effectiveness and benefits in our defined solution.

This article is organised as follows: Section 2 presents keyphrase extraction related works. Afterwards, Section 3 introduces C-Rank, our 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 [5] segmented unsupervised methods in four categories "Graph-based Ranking", that considers

[1] https://babelnet.org/

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

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”, considering that keyphrase extraction and text summarization tasks can benefit from each other and be performed simultaneously, combining “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.

Despite achieving some of the best results, “Language Modeling” approaches require external data to be inputted by the user. Therefore, they will not be covered in this paper. In addition to 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 that do not demand a background textual set to be provided by the user.

Table 1. Unsupervised Keyphrase Extraction techniques.

Models	**Graph-based**	**Topic-based Clustering**	**Heuristics**	**Year**
Text-Rank [8]	X	-	X	2004
BUAP [12]	X	-	X	2010
Topic-Rank [4]		X	X	2013
C-Rank	**X**	**-**	**X**	2019

Mihalcea and Tarau [8] presented the Text-Rank 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 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 [13] 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.* [12] 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.

Bougouin, Boudin and Daille [4] 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 [8]. In the end, *TopicRank* outputs the most common keyphrases of the principal topics as a result.

The proposed approach in this paper, C-Rank, different from other state-of-the-art unsupervised keyphrase extraction approaches, does not request further data to be provided by the user. It relies on predefined background knowledge to leverage the meaning of terms in the extraction process. Instead of gather knowledge from statistical techniques, which demand a compendium that encompasses domain knowledge, C-Rank extracts information from a wide-coverage semantic-network.

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 should extract the keyphrases. It combines the knowledge of the document itself and contained inside BabelNet [11], collected through Babelfy [10]. 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 BabelNet 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.

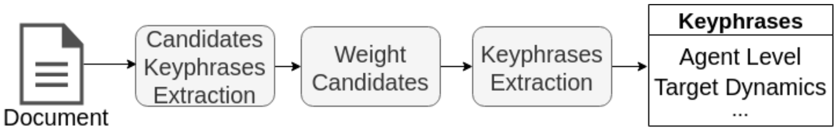

Fig. 1. C-Rank stages.

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 Babelnet, named here concept linking, resulting in a set of paragraphs annotated with babel synsets as illustrated in Figure 2.

Babelfy[3] is the adopted approach to link the document concepts with Babelnet, semantic annotating them. Despite receiving whole texts to process and annotate, some constraints occurred during the use of the Babelfy, as a maximum length of the input text and the service inability to process some special characters.

[3] http://babelfy.org/

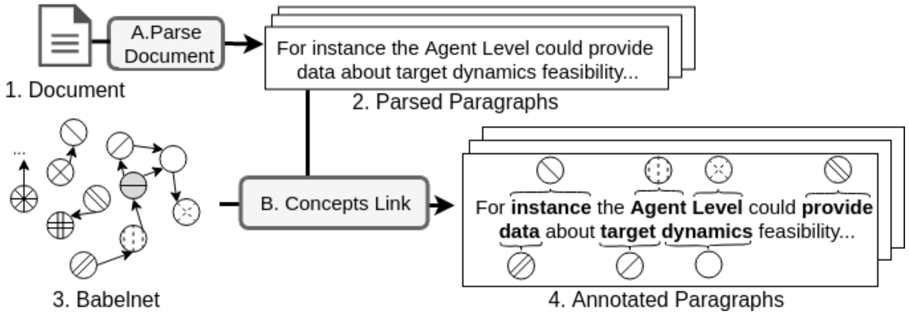
A.Parse Document
1. Document
For instance the Agent Level could provide data about target dynamics feasibility...
2. Parsed Paragraphs
B. Concepts Link
3. Babelnet
For instance the Agent Level could provide data about target dynamics feasibility...
4. Annotated Paragraphs

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

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, which identifies the correspondences between concepts and babel synsets. It can also determine multi-word concepts and its sub-concepts. For example, in "semantic network" the following synsets are 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, explored in Section 4.2, which are represented by dotted arrows.

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.

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

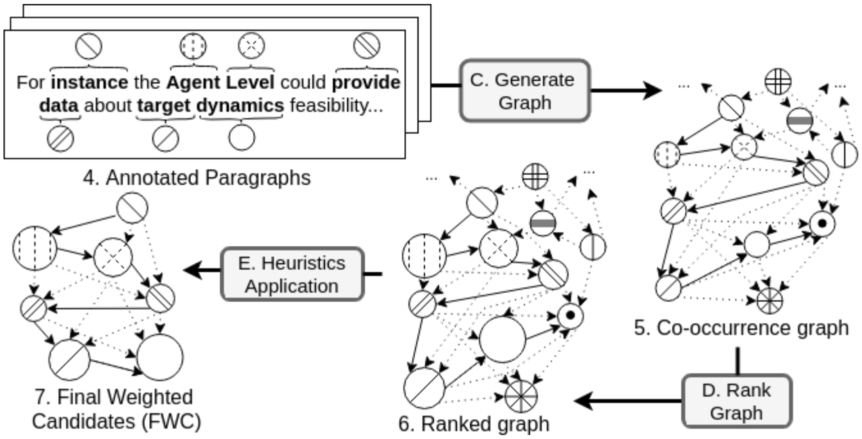
For instance the Agent Level could provide data about target dynamics feasibility...
4. Annotated Paragraphs
C. Generate Graph
5. Co-occurrence graph
D. Rank Graph
6. Ranked graph
E. Heuristics Application
7. Final Weighted Candidates (FWC)

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

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 [2].

C-Rank second stage also applies four heuristics into the graph, which were studied on a training set and are analyzed in Section 4.2. 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 $c_w = c_w^{\frac{1}{len(c)}}$, 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.

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.

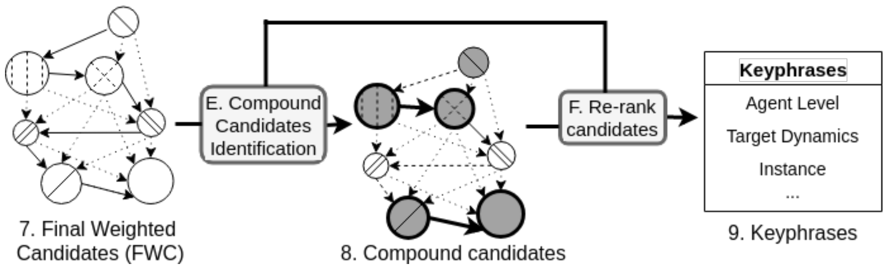

Fig. 4. C-Rank Third Stage: Keyphrase Extraction.

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 2), 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, Process F calculates the ranking weight of the compound candidates as expressed by Equation 3. 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 4, 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 $Keyphrases_d = FWC \cup NCC_{1:6}$.

$$NE_{i,j} = \frac{indirectEdge_{i,j}}{w(Out(i)) + w(In(j))} \tag{2}$$

$$CC_{i,j} = (v_i + v_j) * (\frac{NE_{i,j}}{\sum_{k \in NE} k}) \tag{3}$$

$$NCC_{i,j} = \frac{CC_{i,j}}{\sum_{t \in NE} t} \tag{4}$$

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. C-Rank was developed on python and is available online[4].

4.1 Datasets

We use two standard benchmark datasets to evaluate the results achieved during and after C-Rank development and compare them with other keyphrase extraction approaches, which explored the same datasets.

The first is the SemEval2010 [7] 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.

The INSPEC is the second dataset [6], composed of 3 sets of documents, a training set containing 3000 text files, a validation set with 1500 and a test set consisting of 500. Despite having a similar number of keyphrases assigned per document, the INSPEC dataset, different from the SemEval, is composed of academic abstracts, which makes its files shorter.

[4] https://github.com/maurodlt/C-Rank

4.2 Parameters Refinement

During C-Rank development several variables were defined. To determine the best possible values for those variables, we performed different analyses considering 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 4.3).

The co-occurrence window was the first variable defined in C-Rank development. Table 2 shows the results variance when the co-occurrence window changes and highlight in bold its optimal value.

Table 2. 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

Moreover, two heuristic variables values were defined, *LRC* and *CutOff* Threshold. Both of them were varied and provided optimal results when set to 87% and 18% respectively. Another analyzed C-Rank parameter was the centrality measure used to rank the co-occurrence graph. Table 3 shows the results when this metric changes.

In order to evaluate the Concept Linking usage and the proposed heuristics, Table 4 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.

Table 3. 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 4. 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

4.3 Results

A final evaluation was performed to determine the effectiveness of *C-Rank* results achieved in both SemEval and INSPEC test datasets, which allows comparing *C-Rank* among other keyphrase extraction approaches. Table 5 presents the obtained results and Table 6 shows the comparison with other unsupervised keyphrases extraction techniques.

Table 5. Micro-average precision, recall and f-score on the extraction of Top-5, Top-10, and Top-15 keyphrases on the SemEval2010 and the Inspec test datasets.

	Top-5			**Top-10**			**Top-15**		
	P.	**R.**	**F1.**	**P.**	**R.**	**F1.**	**P.**	**R.**	**F1.**
SemEval	28	9,6	14,2	24,2	16,5	**19,6**	20,3	20,7	20,5
Inspec	32	16	21,2	23,1	23,5	**23,3**	17,1	25,9	20,6

Table 6. Comparison among micro-average precision, recall, and f-score achieved by extracting 10 keyphrases on the SemEval2010 and the INSPEC test datasets. † indicates statistical significance improvement using a 2-sided paired t-test at $p < 0.05$.

	SemEval			**INSPEC**		
	Precision	Recall	F-score	Precision	Recall	F-score
Text Rank	7,9	4,5	5,6†	14,2	12,5	12,7†
BUAP	17,8	12,4	14,4†	-	-	-
Topic-Rank	14,9	10,3	12,1†	27,6	31,5	27,9
C-Rank	24,2	16,5	**19,6**	23,1	23,5	**23,3**

5 Discussion

For the best of our knowledge, 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.

Our approach explored external background knowledge from Babelnet, which is an important characteristic. We found a relevant impact with the use of the concept linking in the keyphrase extraction (*cf.* Table 4). 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.

During the graph construction, the results varying the maximum co-occurrence distances between concepts (*cf.* Table 2) 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 the results, the heuristic values do not impact the algorithm performance. The centrality measure, on the other hand, can significantly decrease the results. As demonstrated by Boudin [3] and corroborated in Table 3, despite being simple, the degree centrality achieves higher results than other popular metrics usually used in keyphrase extraction algorithms, as the Pagerank. Considering the achieved results, further investigations on the use of concept linking in related NLP tasks could support and complement current solutions.

6 Conclusion

Keyphrase extraction plays a key role in the interpretation and analyses of textual documents. Existing proposals heavily rely on training datasets and external input. In this paper, we introduced *C-Rank*, an unsupervised keyphrase extraction algorithm that explored concept linking and graph-based techniques. Our approach enables the analysis of single documents and does not demand a textual compendium to be inserted by users. Our technique explored background knowledge from a wide-coverage semantic-network in a novel approach to obtain candidate Keyphrase and rank them. It used the concepts linked with the network as vertices of a co-occurrence graph, which is ranked based on heuristics and centrality measures. The conducted experiments showed the benefits of the elaborate features in the technique. The evaluation revealed that *C-Rank* outperformed, with statistical significance, all the unsupervised techniques that do not demand extra information to be provided by users on the SemEval2010 benchmark dataset. As future work, we plan to evaluate C-Rank against different types of data, other than scientific-related articles. Furthermore, we will investigate the C-Rank intermediate semantic structures produced in tasks related to the identification of domain topics based on a set of textual documents.

Acknowledgements

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

References

1. 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)
2. 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)
3. Boudin, F.: A comparison of centrality measures for graph-based keyphrase extraction. In: International Joint Conference on NLP. pp. 834–838 (2013)
4. 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)
5. 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)
6. 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)
7. 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)
8. 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)
9. 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)
10. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Computational Linguistics **2**, 231–244 (2014)
11. 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)
12. 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)
13. 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

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