A Comparative Assessment of State-Of-The-Art Methods for Multilingual Unsupervised Keyphrase Extraction

Nikolaos Giarelis, Nikos Kanakaris, Nikos Karacapilidis

A comparative assessment of state-of-the-art methods for multilingual unsupervised keyphrase extraction

Nikolaos Giarelis [0000-0003-2611-3129], Nikos Kanakaris [0000-0001-9352-5807] and Nikos Karacapilidis [0000-0002-6581-6831]

Industrial Management and Information Systems Lab, MEAD
University of Patras, 26504 Rio Patras, Greece
giarelis@ceid.upatras.gr, nkanakaris@upnet.gr,
karacap@upatras.gr

Abstract. Keyphrase extraction is a fundamental task in information management, which is often used as a preliminary step in various information retrieval and natural language processing tasks. The main contribution of this paper lies in providing a comparative assessment of prominent multilingual unsupervised keyphrase extraction methods that build on statistical (RAKE, YAKE), graph-based (TextRank, SingleRank) and deep learning (KeyBERT) methods. For the experimentations reported in this paper, we employ well-known datasets designed for keyphrase extraction from five different natural languages (English, French, Spanish, Portuguese and Polish). We use the F1 score and a partial match evaluation framework, aiming to investigate whether the number of terms of the documents and the language of each dataset affect the accuracy of the selected methods. Our experimental results reveal a set of insights about the suitability of the selected methods in texts of different sizes, as well as the performance of these methods in datasets of different languages.

Keywords: Natural Language Processing, Keyphrase Extraction, Unsupervised Learning, Deep Learning, Graph-based Models, Empirical Research.

1 Introduction

Keyphrase (or keyword) extraction (KE) is a fundamental task in information management systems; it has been defined as the process of extracting keyphrases from a document, i.e. a set of phrases consisting of one or more words that are considered to be meaningful and representative for a document (Hasan and Ng, 2010). Various Information Retrieval (IR) and Natural Language Processing (NLP) tasks - such as text classification, text categorization, text summarization and generation of recommendations based on textual descriptions - greatly benefit from the use of KE methods (Wan and Xiao, 2008a). A variety of supervised and unsupervised KE methods have been proposed so far in the literature, with both categories demonstrating certain advantages and drawbacks. Supervised KE methods demonstrate higher F1 scores than their unsupervised counterparts, but fail to operate on large document collections

with no predefined keyphrases, mainly due to the sheer size of manual work needed by human annotators.

In this paper, we focus on a selected set of prominent unsupervised KE methods. This selection takes into account recent literature reviews (Papagiannopoulou and Tsoumakas, 2020; Campos et al., 2020) and a promising deep learning method. These methods are classified into three categories, upon the approach they build on, namely statistical, graph-based, or deep learning. Statistical methods considered include TF-IDF (Term Frequency - Inverse Document Frequency) (Hasan and Ng, 2010), RAKE (Rapid Automatic Keyword Extraction) (Rose et al. 2010), and YAKE (Yet Another Keyword Extractor) (Campos et al. 2020). Graph-based methods include TextRank (Mihalcea and Tarau, 2004) and SingleRank (Wan and Xiao, 2008b). Finally, the deep learning approach elaborated is KeyBERT (Grootendorst, 2020).

We assess the selected KE methods through a partial match evaluation framework proposed by Rousseau and Vazirgiannis (2015), which calculates the partial F1 score for each document and the final mean F1 score for each dataset. For our experimentations, we use a set of datasets consisting of multiple documents of different length, from five natural languages, namely English, French, Spanish, Portuguese and Polish.

The contribution of this paper lies in: (i) the assessment of prominent unsupervised KE methods based on three different approaches; (ii) the assessment of the selected methods on datasets of different size of documents, topics, and language; (iii) the investigation of whether the language of each dataset affects the accuracy of the selected methods. The remainder of the paper is organized as follows: Section 2 describes the unsupervised KE methods assessed. Section 3 presents the proposed partial match evaluation framework and the outcome of the comparative assessment of the selected methods. Concluding remarks and future work directions are outlined in Section 4.

2 Related Work: Unsupervised Keyphrase Extraction

According to Papagiannopoulou and Tsoumakas (2020), unsupervised KE methods follow a common three-step methodology. Firstly, they select the candidate lexical units by applying a set of heuristics, mostly to filter out unnecessary units from the input text. Secondly, they rank the aforementioned units by utilizing certain syntactic/semantic relationships with other candidate units. Finally, keyphrases are extracted based on the ranked list of candidate words. This section describes the most prominent KE methods that build on statistical, graph-based and deep learning methods. For all mathematical formulations given below, $|x|$ denotes the number of elements found in a set x.

2.1 Statistical Methods

TF-IDF is one of the most common baseline methods in the literature. This method computes a TF-IDF score for each term of a document, based on its frequency in this document and the number of other documents that include it. It is:

$$TF - IDFt \; = \; TFt \; \times \log\left(\frac{|D|}{|d \in D : t \in d \,|}\right) \tag{1}$$

where $TF\text{-}IDF_t$ is the homonym score for term *t*, TF_t is its term frequency, $|D|$ the number of documents, and $|d \in D: t \in d\,|$ the number of documents where *t* is included. Due to the increased runtime in large datasets, since for each term every document in the collection must be traversed and iterated upon its terms, we have slightly altered this method by employing the *TfidfVectorizer* class of *scikit-learn*; instead of $|D|$, we consider the total number of sentences in a document and the total number of sentences where *t* appears in.

RAKE is a prominent statistics-based method (Rose et al. 2010), which uses a list of stopwords and a set of phrase / word delimiters that are used in a combined manner in order to divide the text into candidate keyphrases, while maintaining the sequence of terms as they occur in text. By using these candidate keyphrases, the method builds a term co-occurrence matrix, which is used to calculate the significance of keyphrase as the sum of three metric scores, namely *keyphrase frequency*, *keyphrase degree* (the number of other candidate keyphrases that appear alongside the considered keyphrase), and *ratio of degree to frequency*.

A third method of this category is *YAKE* (Campos et al., 2020), which apart from term frequency utilizes new statistical metrics that consider context and terms spread throughout the document. YAKE first splits the text into individual terms and then calculates a score *S(t)* for each individual term *t.* This score relies on five metrics: T_{case} (casing aspect of a term, which considers uppercase terms and terms with their first letter capitalized, excluding those at the beginning of a sentence, to be more significant than others), T_{pos} (the positional of a term, which favors words found near the start of the document), TF_{norm} (term frequency normalization), T_{rel} (term relatedness to context, which computes the number of different terms that occur on the left and right side of the term), and $T_{difsent}$ (which measures how often a term appears in different sentences). *S(t)* is computed using the formula:

$$S(t) = \frac{T_{rel} * T_{pos}}{T_{case} + \frac{TF_{norm}}{T_{rel}} + \frac{T_{difsent}}{T_{rel}}} \tag{2}$$

As soon as this equation is calculated for each term, a sequence of 1, 2 … *n-gram* candidate keyphrases is produced by utilizing a sliding window of n-grams. For each candidate keyphrase (*ck*), a score *S(ck)* is calculated. It is noted that for smaller values of *S(ck)*, the quality of the *ck* is increased.

$$S(ck) = \frac{\prod_{t \in ck} S(t)}{TF(ck) * (1 + \sum_{t \in ck} S(t))} \tag{3}$$

2.2 Graph-based methods

Graph-based unsupervised KE methods represent a document as a graph, where candidate keyphrases are represented as nodes and the connections between them as edges. After the construction of the document graph, these methods rely on graph

measures that consider various graph structural properties to rank the candidate phrases and select the *top-N* among them.

TextRank (Mihalcea and Tarau, 2004) is one of the most well-known KE methods. It starts by assigning part-of-speech (POS) tags for each term in the text, then the nouns and adjectives are selected for the candidate list. Each candidate keyphrase is added to the graph as a node. Edges are added between terms that are present in a sliding window of *N* terms. In the case of undirected and unweighted edges, the TextRank score ($S(v_i)$) for each node (v_i) is described by the following recursive formula:

$$S(v_i) = (1 - d) + d \times \sum_{v_j \in \Gamma(v_i)} \frac{1}{|\Gamma(v_j)|} S(v_j) \quad (4)$$

where d is the damping factor, set to 0.85 as proposed in (Hasan and Ng, 2010) and $\Gamma(v_j)$ denotes the set of neighboring nodes of v_j. When Equation (4) converges, the nodes are sorted in descending order by their calculated scores.

SingleRank (Wan and Xiao, 2008b) is similar to TextRank, with three key differences (Hasan and Ng, 2010). Firstly, TextRank supports weighted graphs with a slightly different formula than the one stated above (each weighted edge has the same pre-defined weight); on the contrary, in SingleRank each edge has a weight equal to the number of times the connected terms co-occurred in the same sliding window. Secondly, while in TextRank only the highest ranking terms are considered in the candidate keyphrase forming process, low ranked terms can also participate in SingleRank. This causes candidate keyphrases to not be ranked up by individual terms, rather by the sum of all terms forming a keyphrase. The resulting score is then used in descending order to obtain the *top-N* highest scored candidate keyphrases. Thirdly, SingleRank employs a larger window size (usually 10), instead of smaller window sizes used by TextRank (with 2 as minimum). The mathematical formulation of the SingleRank weighted score ($WS(v_i)$) is nearly identical to the weighted version of TextRank, the major difference being that the weight of an edge between two nodes v_i and v_j is replaced by the number of co-occurrences (c_{ij}) between these nodes.

$$WS(v_i) = (1 - d) + d \, x \sum_{v_j \in \Gamma(v_i)} \frac{c_{ij}}{\sum_{v_k \in \Gamma(v_j)} c_{jk}} WS(v_j) \quad (5)$$

2.3 Deep Learning Methods

Recent advances in deep learning enabled researchers to augment classical KE methods, which utilize only graph and statistical measures, by employing word embeddings as a means to capture the semantic relationships between terms in the text, and thus improve the quality of the extracted keyphrases.

KeyBERT (Grootendorst, 2020) relies on BERT-based pre-trained models of word embeddings to augment the quality of the extracted keyphrases. BERT, which stands for Bidirectional Encoder Representations from Transformers, is the original model developed by Google researchers (Devlin et al. 2019). It was made to improve state-

of-the-art NLP tasks. In the scope of this paper, we utilize a similar multilingual pretrained model for unsupervised KE, as described below.

Firstly, for each document the model creates a list of candidate keyphrases, by using the `CountVectorizer` class of scikit-learn. This class implements a simple bag-of-words implementation, which measures the frequency of these keyphrases.

Secondly, a document embedding vector based on the words of the document and an embedding vector for each candidate keyphrase are produced. These embeddings are produced by utilizing the *sentence-transformer* package, introduced in (Reimers and Gurevych, 2019), which is built by using the popular `pytorch` deep learning python library (`pytorch.org`). The aforementioned package comes with many pretrained BERT-based models; in this paper, we opt for the pretrained model called `distiluse-base-multilingual-cased-v2`, which is based on Distilbert (Sanh et al. 2019). Distilbert is a multilingual knowledge distilled model made after the original multilingual Universal Sentence Encoder (MUSE) (Yang et al. 2020). While the original MUSE model supports only 16 languages, this distilled model supports more than 50 languages.

Thirdly, after the production of the required embedding vectors, for each candidate keyphrase, a pairwise cosine similarity score is calculated between the former and the embedding vector of the document. Afterwards the keyphrases are sorted based on their similarity score, in descending order, as a way of ranking them. The basic idea is that keyphrases, which have a vector representation highly similar to the one of the document, are the most representative of the document. In contrast with other methods, KeyBERT includes an extra diversification step of the results. This diversification step of the results is applied using either the Maximal Marginal Relevance or Max Sum Similarity measure. Both of these measures require certain parameters to balance out the number of similar keyphrases without reducing the overall accuracy of the model.

Maximal Marginal Relevance. As mentioned in the previous section, to remedy the shortcomings of highly similar results, a diversification step is applied using the Maximal Marginal Relevance (MMR) measure described in (Bennani-Smires et al., 2018). This measure, which is also leveraged by KeyBERT, is:

$$\text{MMR} = argmax_{C_i \in C \setminus K} \Big[\lambda * \widetilde{cos}_{sim}(C_i, \quad doc) - (1-\lambda)\, max_{C_j \in K}\, \widetilde{cos}_{sim}(C_i, C_j) \Big] \quad (6)$$

where C is the set of candidate phrases, K is the set of extracted keyphrases, doc is the document embedding vector, C_i, C_j are the embedding vectors of candidate keyphrases i, j respectively, $\widetilde{cos}_{sim}$ the normalized cosine similarity function, applied between two vectors, and λ is a parameter that controls the relevance and the diversity of the candidate keyphrases. A value of λ = 0.5, ensures balance among them. Grootendorst (2020), suggests a value of λ = 0.7 to ensure more diversification in the final list of extracted keywords.

Max Sum Similarity. The second measure for applying diversification to the candidate keyphrases is *Max Sum Similarity* (Grootendorst, 2020). This measure selects similar keyphrases to the document, which when considered in pairs are mostly dissimilar to one another. This measure gains its name from the summing of the vector cosine similarities for each pair of terms found in every pair of candidate phrases. The most dissimilar pairs with the maximum sum of distance between their vector representations are considered. To control the number of dissimilar pairs found in the final list of extracted keywords, the author uses a parameter for his method called *nr_candidates*, which selects the number of unique candidate phrases.

3 Experiments

For the implementation and evaluation of the selected KE methods, we used the Python programming language. The full code, datasets, and evaluation results of our experiments are freely available at https://github.com/NC0DER/KeyphraseExtraction.

3.1 Datasets

To test how well multilingual unsupervised KE methods work, we chose five datasets from five different natural languages, which can be found online at https://github.com/NC0DER/KeyphraseExtraction/tree/main/Datasets. Specifically:

- For English, we opted for the validation subset (500 documents) out of the entire *Hulth* dataset (Hulth 2003), which contains 2000 abstracts of computer science papers. Specifically, we used the uncontrolled keyphrases, since they appear more often in the text.
- For French, we opted for *WikiNews* (Bougouin et al. 2013), which contains 100 documents from French news articles published from May to December 2012.
- For Portuguese, we opted for *110-PT-BN-KP* (Marujo et al. 2012), which contains 110 transcripted text documents from 8 broadcast news programs talking about various subjects such as politics, sports, finance and other.
- For Polish, we opted for *pak2018* (Campos et al. 2020) which contains 50 abstracts from scientific articles.
- For Spanish, we opted for a small subset of *Cacic* and *Wicc* (Aquino and Lanzarini, 2015) datasets. Wicc is composed of 1640 computer science scientific articles published between 1999 and 2012, while Cacic contains 888 scientific papers between 2005 and 2013. When we manually inspected all those datasets, we noticed that both Cacic and Wicc had a low number of keyphrases found as-is in the text; for this reason, we selected a small subset out of both datasets (57 and 78 documents, respectively); these documents were selected because their associated keyphrase files had at least one keyphrase present in each document.

3.2 Experimental Setup

For our experiments, we start by calculating TfidfVectorizer (see Section 2.1). For the other two statistical methods, we employ a popular implementation of RAKE (https://github.com/fabianvf/python-rake) and the official implementation of YAKE (https://github.com/LIAAD/yake). For graph-based methods, we use the implementations of TextRank and SingleRank available at https://github.com/DerwenAI/pytextrank and https://github.com/boudinfl/pke (Boudin, 2016). For the deep learning method, we use the official implementation of KeyBERT (Grootendorst, 2020) available at https://github.com/MaartenGr/KeyBERT.

Regarding their parametric setup, all methods are set to produce n-grams of size ranging from 1 to 3. For each method, the *top-10* keyphrases are extracted and then compared with the manually assigned keyphrases, as analytically described in Section 3.3. A list of parameters, which are set for each KE method, can be seen below:

Table 1. Parameter configurations for each of the unsupervised KE methods.

Method	Parameters	Approach
TfidfVectorizer	ngram_range = (1, 3), top_n = 10	Statistical
RAKE	top_n = 10	Statistical
YAKE (seqm)	n = 3, top_n = 10, dedupLim = 0.9, dedupFunc = 'seqm', windowsSize = 1	Statistical
TextRank	top_n = 10	Graph-based
SingleRank	top_n = 10	Graph-based
KeyBERT (mmr)	ngram_range = (1, 3), top_n = 10, method = 'mmr', diversity = 0.7	Deep Learning
KeyBERT (maxsum)	ngram_range = (1, 3), top_n = 10, method = 'maxsum', diversity = 0.7	Deep Learning

On a sidenote, the parameters (method, diversity) of KeyBERT refer to the diversification measures explained in Section 2.3. YAKE uses the term deduplication function (dedupFunc) for its diversification measure. In their work, Campos et al. (2020) consider various such functions, with the best being the sequence matcher (seqm), after extensive evaluation. For both methods, we use the recommended parameters of their respective authors for optimal use.

3.3 Evaluation

To evaluate the selected methods, we adopt the partial match framework pro-posed by Rousseau and Vazirgiannis (2015). The rationale behind this framework is that while

KE methods often form the correct keyphrase, when tested under exact matching the tests often yield low results. According to this framework, the following metrics are defined:

$$Partial\ Precision = \frac{number\ of\ partially\ matched\ keyphrases}{total\ amount\ of\ extracted\ keyphrases} \quad (7)$$

$$Partial\ Recall = \frac{number\ of\ partially\ matched\ keyphrases}{total\ amount\ of\ assigned\ keyphrases} \quad (8)$$

We also note that the partial F1 score (*pF1*), which is the harmonic mean be-tween the partial precision and recall, is defined as:

$$pF1 = \frac{2 * Partial\ Precision * Partial\ Recall}{Partial\ Precision + Partial\ Recall} \quad (9)$$

The number of partially matched keyphrases corresponds to the number of extracted keyphrases that partially match with those assigned by human authors. The total number of extracted keyphrases is equal to the number of *top-N* extracted keyphrases, which is set to 10 in our experiments. The total number of assigned keyphrases correspond to the number of keyphrases manually assigned by human annotators of the specific dataset.

Table 2. Statistics of each dataset; Words per Document (W/D), Text Category based on W/D (Mean) and Number of Documents.

Dataset	W/D (Mean)	W/D (Max)	W/D (Min)	Text Category	Number of Documents
Cacic (57)	3894	6301	1713	Long (Full texts)	57
Wicc (78)	1863	4347	10	Long (Full texts)	78
110-PT-BN-KP (110)	301	955	13	Medium (News Articles)	110
WikiNews (100)	282	1026	126	Medium (News Articles)	100
Hulth Validation (500)	119	285	16	Short (Abstracts)	500
pak2018 (50)	97	170	55	Short (Abstracts)	50

Before we compare the keyphrases between the human annotators and those extracted from the KE methods, we lowercase all keyphrases, remove punctuation marks and apply stemming. For stemming, we use the *Snowball Stemmer* found in the *NLTK toolkit* (https://www.nltk.org/), due to its ability to stem texts from different languages, such as English, French, Spanish, Portuguese and others. Since this stemmer does not support the Polish language, we use the *pystempel* stemmer for the Polish dataset

(https://pypi.org/project/pystempel/). The parameter configurations are summarized in Table 1. Statistics of each dataset are presented in Table 2, while the experimental results are shown in Table 3. The code of all experimentations reported in this paper can be found on the following GitHub repository: https://github.com/NC0DER/KeyphraseExtraction.

Table 3. Partial F1 score at 10 extracted keywords (pF1@10), per KE method, for each diversification measure. Bold font indicates the best combination of method (and measure if it uses any) in brackets.

Dataset (pF1@10)	TfidfVectorizer	Rake	Yake (Seqm)	KeyBERT (MMR)	KeyBERT (MaxSum)	TextRank	SingleRank
Cacic (57)	0.077	0.212	0.213	0.206	**0.316**	0.244	0.266
Wicc (78)	0.043	0.201	0.231	0.233	**0.270**	0.256	0.266
110-PT-BN-KP (110)	0.187	0.252	0.333	0.248	**0.380**	0.297	0.336
WikiNews (100)	0.242	0.387	**0.578**	0.373	0.567	0.490	0.522
Hulth Validation (500)	0.378	0.593	0.541	0.493	0.580	0.618	**0.629**
pak2018 (50)	0.085	0.177	0.173	0.158	0.159	0.202	**0.205**

As shown in Table 3, KeyBERT achieves the highest F1 score for the Spanish (Cacic, Wicc) and Portuguese (110-PT-BN-KP) datasets. For the English (Hulth Validation) and Polish (pak2018) datasets, the graph-based methods achieve the best results. For the French (WikiNews) dataset, YAKE has the best performance. YAKE also achieves the best results among all statistical methods. It is also noted that, throughout all datasets, SingleRank outperforms TextRank.

Moreover, our experimentations indicate that the best method for long texts is KeyBERT (MaxSum) and for short texts is SingleRank. We also conclude that SingleRank is able to model the correlations between the words more accurately than other methods for short texts. However, in long texts, significant keyphrases that do not appear as often as others are not extracted. This is due to the fact that graph-based methods rely on co-occurrence of terms, thus a suboptimal ranking of non-frequent keyphrases is produced. Furthermore, we conclude that KeyBERT increases the quality of extracted keyphrases on long texts for two reasons: (i) it utilizes word embeddings, which are able to capture contextual similarity between terms; (ii) it employs a selected diversification method, which leads to a richer set of keyphrases.

Finally, we observe that the language of a dataset does not affect the accuracy of any of the selected methods. As seen in Tables 2 and 3, for datasets belonging to the same text category, even for different natural languages, the selected methods are ranked similarly.

4 Conclusions

We have comparatively assessed a set of unsupervised multilingual KE methods across different datasets. Our experimental results reveal that the deep learning method (KeyBERT) employed is more suitable for long sized texts, whereas the graph-based methods are more suitable for short sized texts. A known technical limitation of KeyBERT is that it does not work for extremely short texts, i.e. texts with less than *2 * top-N* unique terms.

A limitation of this work is certainly the limited number of employed datasets. Additional datasets will be considered in future work, aiming to further validate the outcomes of this paper. Future work directions also include: (i) the use of larger pre-trained BERT models, aiming to improve the contextual similarity between terms; (ii) the fine-tuning of these models for domain-specific applications; (iii) the comparative evaluation of additional unsupervised deep learning KE methods, including EmbedRank (Bennani-Smires et al., 2018), Key2Vec (Mahata et al., 2018) and Reference Vector Algorithm (Papagiannopoulou and Tsoumakas, 2018).

Acknowledgments. The work presented in this paper is supported by the inPOINT project (https://inpoint-project.eu/), which is co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (Project id: T2EDK- 04389).

References

1. Aquino, G. O., Lanzarini, L. C.: Keyword identification in Spanish documents using neural networks. Journal of Computer Science and Technology, 15(2), 55-60 (2015).
2. Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., Jaggi, M.: Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In: Proceedings of the 22nd Conference on Computational Natural Language Learning, pp. 221–229. Association for Computational Linguistics, Brussels, Belgium (2018).
3. Boudin, F.: pke: an open source python-based keyphrase extraction toolkit. In: Proceedings of the 26th International Conference on Computational Linguistics: System Demonstrations, pp. 69–73. The COLING 2016 Organizing Committee, Osaka, Japan (2016).
4. Bougouin, A., Boudin, F., Daille, B.: TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing, pp. 543–551. Asian Federation of Natural Language Processing, Nagoya, Japan (2013).
5. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt A.: YAKE! Keyword extraction from single documents using multiple local features. Inf. Sci. 509, 257–289 (2020).
6. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019).

7. Grootendorst, M.: KeyBERT: Minimal keyword extraction with BERT. v0.1.3, Zenodo (2020).
8. Hasan, K., Ng, V.: Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art. In: Coling 2010: Posters, pp. 365–373. Coling 2010 Organizing Committee, Beijing, China (2010).
9. 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, USA (2003).
10. Mahata, D., Kuriakose, J., Shah, R. R., Zimmermann, R.: Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 634–639. Association for Computational Linguistics, New Orleans, Louisiana (2018).
11. 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. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), pp. 399 – 403. European Language Resources Association (ELRA), Istanbul, Turkey (2012).
12. Mihalcea, R., Tarau, P., TextRank: Bringing Order into Texts. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411. Association for Computational Linguistics, Barcelona, Spain (2004).
13. Papagiannopoulou, E., Tsoumakas, G.: Local word vectors guiding keyphrase extraction. Information Processing & Management, 54(6), 888–902 (2018).
14. Papagiannopoulou, E., Tsoumakas, G.: A Review of Keyphrase Extraction. WIREs Data Mining Knowledge Discovery, 10(2) 10:e1339 (2020).
15. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 3982–3992, Association for Computational Linguistics (2019).
16. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. Text Mining: Applications and Theory, Editors: Michael W. Berry Jacob Kogan, John Wiley & Sons, Ltd (2010).
17. Rousseau, F., Vazirgiannis, M.: Main core retention on graph-of-words for single-document keyword extraction. In Proceedings of the 37th European Conference on IR Research (ECIR 2015), pp. 382–393, Vienna, Austria (2015).
18. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (NeurIPS) (2019).
19. Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Abrego, G. H., Yuan, S., Tar, C., Sung, Y. H., Strope, B., Kurzweil, R.: Multilingual Universal Sentence Encoder for Semantic Retrieval. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 87–94, Association for Computational Linguistics (2020).
20. Wan, X., Xiao, J.: CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling), pp. 969–976, Coling 2008 Organizing Committee, Manchester, UK (2008a).

21. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of the 23rd national conference on Artificial intelligence (AAAI), pp. 855–860, Chicago, Illinois, USA (2008b).