Keyword Extraction from Texts: A Comparative Study of Frequency-Based and Graph-Based Methods

Hortencia Alejandra Ramírez Vázquez
School of Engineering and Science
Tecnológico de Monterrey
Monterrey, Mexico
a01750150@tec.mx

Elvira Ballinas García
School of Engineering and Science
Tecnológico de Monterrey
Monterrey, México
A01171994@tec.mx

Luis Humberto Sánchez Vaca
School of Engineering and Science
Tecnológico de Monterrey
Monterrey, Mexico
a01638029@tec.mx

***Abstract*—This paper aims to analyze texts and extract the main keywords by two keyword extraction methods. The core functionality involves extracting words from the text and comparing them with an online dataset. The extraction process will use two methods: word frequency analysis, and graph-based approaches. The core functionality involves identifying keywords within the document and comparing them with an online dataset to analyze further and validate the key terms.**

***Index Terms*—text analysis, key extraction, NLP**

I. Introduction

Natural Language Processing (NLP) is a subfield of computer science and artificial intelligence (AI) that enables computers to understand and communicate with human language. NLP allows computers to recognize, understand and generate natural language by combining computational linguistics with statistical modeling, machine learning and deep learning. [1]

Models that use NLP find relations between the constituent parts of language using different techniques for data preprocessing, feature extraction, and modeling. Data preprocessing is used because the text needs to be converted into a format the model can understand, thus improving its performance. For feature extraction, there are different measures to describe a document in relation to the corpus. These features are the inputs for the modeling. This model will vary depending on the task that it is trying to accomplish. [2]

NLP can be used for different purposes. For example, text classification can be used to perform sentiment analysis, where tags are assigned to text to put them in categories. In machine translation, the computer translates a text to a different language. Natural language generation, in this case, the NLP algorithms, are used to analyze data and produce content based on it. Text extraction, where the computer summarizes texts to find important parts, can be used for keyword extraction. [3]

Keywords and keyphrases are crucial for text analysis, as they facilitate the synthetic comprehension of ideas. A keyword/key phrase is present in a text if it appears as a single contiguous word. It can be extracted using linguistic techniques that use frequency, relevance, or structural patterns [4]. However, manually extracting keypoints from a text can be a taxing task [5]. Luckily, there have been significant advances in the keyphrase/keyword extraction field. Some standard methods involve the following:

- **Conditional Random Fields**, is a kind of features extraction and pre-processing method that implements sentence segmentation and labelling. CRF++ tool and POS-tag is used to extract keywords from de pdf [6].
- **Machine learning aproaches** are very important for KE. For example, Support Vector Machines (SVM) is used to construct a classification model. First, this method extracts relevant features for each phrase based on the frequency of appearance of the word, part of speech, and position of the text. SVM is trained in labeled data where words are marked as keywords or not. Finally, this model can predict which words are relevant [7]. Usually, these kinds of machine learning algorithms require model training data.
- **Graph-based methods** allow the exploration of relationships and structural information very effectively. In this case, the document is modeled as a graph in which nodes and their relations represent words are represented by links. Regarding keyword extraction, various centrality measures are proposed for ranking the words in a text, which refers to indicators that identify the most important vertices within a graph [8].
- **Linguistic approaches** rely on the language characteristics of words, sentences, and entire documents. These methods look at various aspects like the meaning of words (lexical), sentence structure (syntactic), the more profound meaning or context (semantic), and how ideas flow and connect throughout the text (discourse analysis) [8].
- **Convolutional Neural Networks (CNNs)** are used to extract information from documents, often in combination with Natural Language Processing (NLP) techniques. A notable example is Chargrid, which applies optical character recognition (OCR) to identify text and then

maps it to a grid format for analysis. This approach is beneficial for documents where information is organized within structured shapes, such as rectangles or tables, enabling more accurate extraction and interpretation of content [9].

- **Recurrent Neural Networks (RNNs)** process data sequentially, meaning they can capture each word's context in relation to the words around it. This is crucial for keyword extraction, as keywords often depend on the context of a sentence. RNNs, especially when combined with attention mechanisms, can improve keyword extraction by focusing on the more relevant parts of the text [10]. The attention mechanism helps the network "pay more attention" to specific words or phrases likely to be keywords based on their importance in the overall context.

Regardless of all the existing methods, the keyphrase extraction problem still performs poorly compared to other NLP tasks [11]. Algorithms often struggle to retrieve the context for keyphrases due to language ambiguity or paragraph structures. Some strategies to overcome these limitations include using graph theory and machine learning conjunctively to improve the contextual understanding of keywords/phrases or incorporating deep-learning models to enhance semantic analysis.

II. Method and Data

For this paper, the dataset 500N-KPCrowd-v1.1 was utilized to train the keyword extraction methods. This dataset covers various topics, including art and culture, business, crime, fashion, health, politics, science, sports, and technology.

The ground truth is built using Amazon's Mechanical Turk service to recruit and manage taggers. Multiple annotators were required to look at the same news story and assign a set of keywords from the text itself. The final ground truth consists of keywords selected at least by 90% of the taggers.

Let,

$$T = \{t_1, t_2, t_3, ..., t_n\} \tag{1}$$

be the set of texts, where each t_i represents a text,

$$K = \{k_1, k_2, k_3, ..., k_m\} \tag{2}$$

be the set of keywords, where each k_j represents a keyword.

Where, each text can contain a subset of relevant keywords from K. Such that,

$$\{k1, k2, k3, ..., k_i\} \subseteq t_j$$

The methodology for keyword extraction was implemented by developing, testing, and evaluating the extraction algorithms, as shown in 1.

The objective of this paper is to compare two different keyword extraction methods: a classical extraction based on word frequency, and the topic rank, which is a graph-based method. Each method will be assessed under the same conditions to clearly compare its effectiveness.

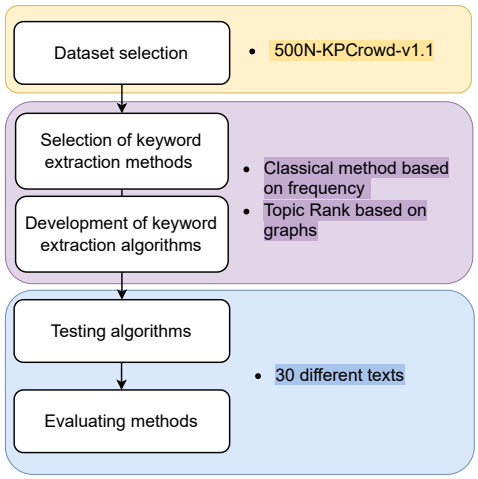

Fig. 1. Keyword extraction methodology

A. Classic keyword extraction method based on word frequency

This algorithm is based on the frequency of each word appearing in a given text. By counting the occurrences of each word, the algorithm identifies the most frequently mentioned terms, which are considered the keywords of the text. The higher the word frequency, the more relevant it is for keyword extraction.

- **Text Preprocessing:** The first step was to normalize the text by converting all words to lowercase to ensure that keyword extraction is not affected by case differences. Next, special characters (such as punctuation or non-alphabetic symbols) were removed using regular expressions, leaving only words and spaces.
- **Tokenization:** The text was tokenized. It is split into individual words to work with manageable units of text.
- **Stopwords removal:** Removing stopwords refers to those common words that do not contribute significant meaning to keyword extraction, such as articles, prepositions, and pronouns.
- **Priority Word Filtering:** Each filtered word is compared to a set of priority words extracted from the dataset. Only the words that match these priorities are kept for further analysis.
- **Frequency calculation:** This step is based on calculating the frequency of each word in the filtered text.

B. Topic Rank

TopicRank [12] is an unsupervised graph method for extracting keyphrases from the topics of a document. It identifies noun phrases that represent the main topics of a document. A graph is created that provides a topical representation of the document. Keyphrases are extracted by selecting the most representative candidate from each top-ranked topic.

To perform keyphrase extraction, the following steps are done:

- **Text Preprocessing:** First, the text needs to be preprocessed using techniques such as sentence segmentation, word tokenization, and Part-of-Speech tagging.
- **Topic identifications:** It is necessary to find the keyphrase candidates that represent the main topics of a document. In this case, the most extended sequences of nouns and adjectives are extracted as candidates.
- **Graph-based ranking:** With the keyphrase candidates, the document is represented by a complete graph. In this graph, topics are vertices, and edges are weighted according to the strength of the semantic relation between vertices. Once the graph is created, TextRank [13] ranks the topics, assigning a significance score to each topic.
- **Keyphrase selection:** Finally, selecting only the most representative keyphrase candidates is necessary. This selection helps avoid redundancy and get a good coverage of topics.

The implementation of this method used in this document is based on [14].

III. Evaluation

We will use a dataset consisting of 30 texts to test and evaluate the keyword extraction methods. These texts will be processed to extract keywords, and the performance of each extraction method will be evaluated using three key metrics: precision, recall, and F1-score.

Precision, also known as positive predictive value, helps measure the models' confidence or ability to make positive predictions accurately. In other words, its value represents the ratio of relevant (or real) instances over the total predictions. Mathematically, it is defined as follows [15],

$$P = \frac{tp}{pp}, \tag{3}$$

where tp stands for true positives, and pp are all the predicted positives, which can be calculated by:

$$pp = tp + fp, \tag{4}$$

where fp is the count of false positives.

Recall is the metric that allows us to assess the models' sensitivity. It quantifies the models' ability to identify all relevant instances. It is defined as [15],

$$R = \frac{tp}{rp} \tag{5}$$

where rp are real positives and can be calculated as

$$rp = tp + fn, \tag{6}$$

where fn refer to false negatives.

Finally, F1 - score measures precision and recall in the same metric. It helps visualize a balance between both metrics and helps make an unbiased comparison between models. It is defined as,

$$F1 = 2\frac{P * R}{P + R}. \tag{7}$$

IV. Results

Thirty texts on topics related to science and fashion were selected to assess the performance of the keyword extraction process. The implementation of these methods can be accessed through this google colab.

The comparison between the two keyword extraction methods is presented in Table I. This table shows the values obtained for each of the three metrics — recall (R), precision (P), and F1-score (S) — for the 30 texts analyzed. These metrics provide a comprehensive evaluation of each method's performance, with recall measuring the ability to identify relevant keywords correctly, precision assessing the accuracy of the extracted keywords, and F1-score offering a balanced evaluation of both recall and precision. The results allow for a direct comparison of how each method performs in terms of extracting key terms from the selected texts.

The results shown in the table show that the word frequency-based method achieved low recall values with an average of 0.14, but high precision, around 0.61. Additionally, most of the F1-score values were around 0.21. Results showed that this model has low recall values while high precision values.

The TopicRank method showed medium to low recall values, with an average of 0.18. Precision got better results in some cases, reaching more than 0.4, but on average, it only got 0.23. This resulted in a medium to low performance for the F1-score, achieving an average of 0.19. These results should take into account that the method extracted at most 50 keywords.

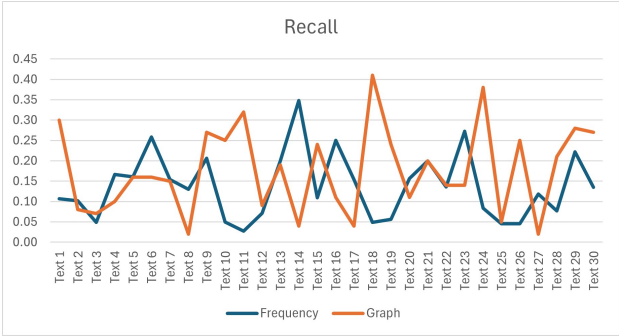

Fig. 2. Comparison of recall metric for three keyword extraction methods - words frequency, graphs, and RNNs - applied to 30 texts of the dataset 500N-KPCrowd-v1.1.

A comparison of the two methods based on recall, precision, and F1-Score metrics is presented in Figures 2, 3, and 4. The recall chart indicates that both the word frequency method and the graph-based method yield values below 0.5. This metric is not constant for either method, with values ranging between 0.0 and 0.45. While the word frequency method achieves higher precision values, the graph-based method performs lower in this metric. Finally, the F1 score demonstrates the relation between recall and precision. For both methods, the results of this metric is rather low, with neither being significantly better than the other.

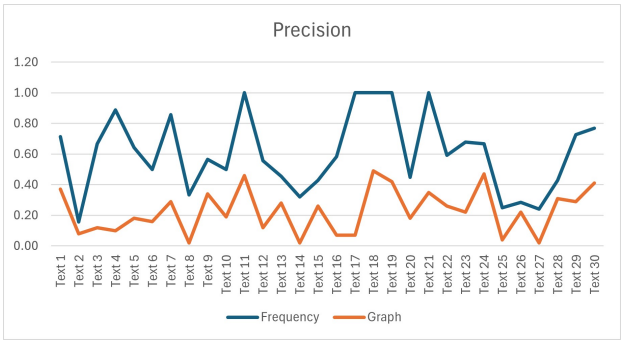

Fig. 3. Comparison of precision metric for three keyword extraction methods - words frequency, graphs, and RNNs - applied to 30 texts of the dataset 500N-KPCrowd-v1.1.

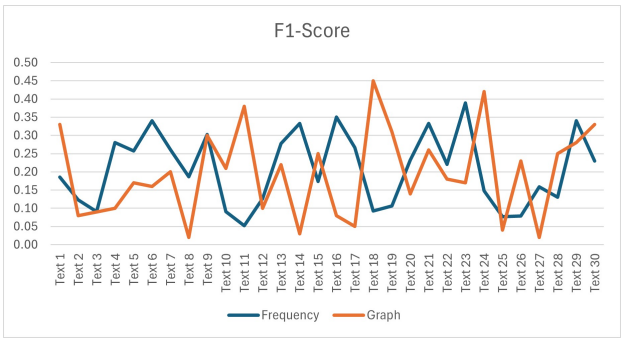

Fig. 4. Comparison of F1-scores metric for three keyword extraction methods - words frequency, graphs, and RNNs - applied to 30 texts of the dataset 500N-KPCrowd-v1.1.

	Frequency			Graph		
	R	**P**	**S**	**R**	**P**	**S**
Text 1	0.11	0.71	0.19	0.30	0.37	0.33
Text 2	0.10	0.16	0.12	0.08	0.08	0.08
Text 3	0.05	0.67	0.09	0.07	0.12	0.09
Text 4	0.17	0.89	0.28	0.10	0.10	0.10
Text 5	0.16	0.64	0.26	0.16	0.18	0.17
Text 6	0.26	0.50	0.34	0.16	0.16	0.16
Text 7	0.15	0.86	0.26	0.15	0.29	0.20
Text 8	0.13	0.33	0.19	0.02	0.02	0.02
Text 9	0.21	0.57	0.30	0.27	0.34	0.30
Text 10	0.05	0.50	0.09	0.25	0.19	0.21
Text 11	0.03	1.00	0.05	0.32	0.46	0.38
Text 12	0.07	0.56	0.13	0.09	0.12	0.10
Text 13	0.20	0.45	0.28	0.19	0.28	0.22
Text 14	0.35	0.32	0.33	0.04	0.02	0.03
Text 15	0.11	0.43	0.17	0.24	0.26	0.25
Text 16	0.25	0.58	0.35	0.11	0.07	0.08
Text 17	0.15	1.00	0.27	0.04	0.07	0.05
Text 18	0.05	1.00	0.09	0.41	0.49	0.45
Text 19	0.06	1.00	0.11	0.24	0.42	0.31
Text 20	0.16	0.45	0.23	0.11	0.18	0.14
Text 21	0.20	1.00	0.33	0.20	0.35	0.26
Text 22	0.14	0.59	0.22	0.14	0.26	0.18
Text 23	0.27	0.68	0.39	0.14	0.22	0.17
Text 24	0.08	0.67	0.15	0.38	0.47	0.42
Text 25	0.05	0.25	0.08	0.05	0.04	0.04
Text 26	0.05	0.29	0.08	0.25	0.22	0.23
Text 27	0.12	0.24	0.16	0.02	0.02	0.02
Text 28	0.08	0.43	0.13	0.21	0.31	0.25
Text 29	0.22	0.73	0.34	0.28	0.29	0.28
Text 30	0.14	0.77	0.23	0.27	0.41	0.33
Average	0.14	0.61	0.21	0.18	0.23	0.19

TABLE I
COMPARISON OF METRICS VALUES - RECALL (R), PRECISION (P), AND F1-SCORE (S).

V. DISCUSSION

This section discusses the study's main findings, focusing on the implications, limitations, and potential improvements of the proposed methods.

For the word frequency-based extraction method, 17 texts achieved high precision values with an average of 0.61, indicating that most of the detected words are real keywords. However, all texts showed low recall with values around 0.14, suggesting that the model can only detect a few amount of all the keywords. These metrics indicate that the model can identify the most general keywords, but it skips many of them. This conclusion is further supported by the low F1 score, which reflect a poor balance between recall and precision due to the lack of detecting all the real keywords.

The word frequency method is useful for a general understanding of the text's context. Compared to other methods that combine word frequency and association [16], which reached a maximum precision of 0.29 with a recall of 0.25, our method had better precision and below values of recall. Additionally, the Rapid Automatic Keyword Extraction (RAKE) method [17], also based on word frequency, achieved a precision of 0.33, and 0.41 for recall, which indicates that while those models identify too many irrelevant keywords and also failed in detecting a great number of actual keywords.

These results indicate that our word frequency method achieves comparable accuracy to other methods employing similar approaches. Furthermore, this comparison highlights that while the word frequency-based method can be effective for general keyword extraction, although it may still struggle with precision and relevance compared to more advanced techniques.

On the other hand, the TopicRank method achieved lower precision results than the frequency-based method. It performed well in nine texts with a value above 0.3, of which five were above 0.4. However, on average, it resulted in 0.23, around a third of the other methods. So, in most cases, the keywords found by this method were not relevant. Regarding recall, TopicRank performed slightly better than the frequency method, with an average of 0.18, with almost half of the text above 0.2. This indicates that this method can't find most of the actual keywords. Both metrics result in a low F1-score with a value of 0.19 on average, indicating that the method is ineffective in finding keywords.

Comparing the results obtained in this work with other papers such as [12] where the method tested here is introduced. The results of the WikiNews dataset for SingleRank showed a precision of 0.19, recall of 20.7, and F1-score of 0.19. For TopicRank, it had a precision of 0.35, recall of 0.37, and F1-score of 0.35. This shows a similar performance compared to SingleRank, but a poor comparison against the same method

may suggest a lousy implementation. Compared to the NE-Rank method [18] that integrated the nodes' weight into the formula and claimed to result in a more natural ranking of text instead of only focusing on the co-occurrence relationship, we obtained a lower precision since it got 0.34 in their testings. Finally, comparing with a method based on using knowledge graphs [19] that got a 0.21 in precision, 0.55 in the recall, and 0.29 in F1-score, we can see that even though our precision was slightly better, it had a much significant recall wit a higher F1-score, making it more effective than the one used in this work.

As a result of this comparison, we can see that our implementation of the TopicRank method was not the best. Further research should be done to improve its performance and compared using the same datasets as the original work to ensure a good ground truth to compare and improve this method with other graph-based methods that have achieved better results in the keyphrase extraction problem.

VI. Conclusion and future work

The current study evaluated the effectiveness of two different methods of key-word extraction: word-frequency based analysis and topic rank. The first one, obtained good results for precision, however, it failed to find all the expected key-words, with its average recall being rather low. This is generally the case for key-word extraction problems, where it is hard for models to extract all the expected key-words. The case of topic rank was interesting, as the implementation, compared to the paper that introduced this method [12], did not give the results we were expecting. This is a strong indicator of areas for optimization or the need for more robust processing and tuning.

In general, we can conclude we can trust that the predicted key-words that the frequency model yields are actually valid and relevant keywords, however, it is not the best method to extract all the relevant key-words, making it less suitable for problems that require comprehensive keyword identification, as it fails to capture the broader semantic context of a text. On the other hand, while the topic rank method had a slightly better performance in recall, and a better balance between precision and recall, its overall performance was suboptimal, with lower F1 and precision results. This suggests that our method is not reliable to identify human keywords, perhaps due to limitations in the graph construction and the ranking algorithm. Moreover, the results highlight a common challenge in keyword extraction: handling the trade-off between precision and recall. While the frequency model is a good starting point for the general problem of keyword extraction, topic rank shows promise for extracting contextually significant keywords, albeit with room for improvement.

Hybrid approaches might be a good future improvement, creating a model that effectively combines the strengths of both frequency and graph based approaches. Furthermore, more advanced techniques such as deep learning models like RNNs, could greatly improve the contextual understanding and therefore, semantic relevance of the predicted data. RNNs, however, might be tricky to use, as they can be highly resource-consuming, and training the model might take hours. Also, adding more sophisticated pre-processing steps can help categorize and manage the data better. Future research should also focus on generalizing these findings by evaluating the techniques on different datasets that have different text structures and domains.

VII. Acknowledgment

Firstly, we would like to thank the Laboratory of Artificial Intelligence and Decision Support (LIAAD) for their work in collecting the dataset used for this paper.

We would also like to acknowledge our professors, Raul Monroy and Hector Ceballos, for their guidance and teachings.

Finally, we want to acknowledge the equal contributions of Hortencia Ramírez, Elvira Ballinas, and Luis Vaca for their collaboration in the research and implementation of the methods used and the analysis and interpretation of the results presented in this work.

References

[1] J. Holdsworth, "What is nlp (natural language processing)?," 2024. Accesed September 25, 2024.

[2] DeepLearning.AI, "A complete guide to natural language processing," 2023. Accesed September 25, 2024.

[3] A. S. Gillis, B. Lutkevich, and E. Burns, "What is natural language processing (nlp)?," 2024. Accesed September 25, 2024.

[4] M. Umair, T. Sultana, and Y.-K. Lee, "Pre-trained language models for keyphrase prediction: A review," *ICT Express*, vol. 10, no. 4, pp. 871–890, 2024.

[5] C. Q. Nguyen and T. T. Phan, "An ontology-based approach for key phrase extraction," in *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pp. 181–184, 2009.

[6] H. M. Hasan, F. Sanyal, D. Chaki, and M. Ali, "An empirical study of important keyword extraction techniques from documents," 12 2017.

[7] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232–247, 2016.

[8] S. Beliga, A. Meštrović, and S. Martincic-Ipsic, "An overview of graph-based keyword extraction methods and approaches," *Journal of Information and Organizational Sciences*, vol. 39, pp. 1–20, 07 2015.

[9] O. Bensch, M. Popa, and C. Spille, "Key information extraction from documents: Evaluation and generator," 2021.

[10] Q. Zhang, Y. Wang, Y. Gong, and X. Huang, "Keyphrase extraction using deep recurrent neural networks on twitter," in *Conference on Empirical Methods in Natural Language Processing*, 2016.

[11] M. Song, Y. Feng, and L. Jing, "A survey on recent advances in keyphrase extraction from pre-trained language models," *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 2153–2164, 2023.

[12] A. Bougouin, F. Boudin, and B. Daille, "TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction," in *International Joint Conference on Natural Language Processing (IJCNLP)*, (Nagoya, Japan), pp. 543–551, Oct. 2013.

[13] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (D. Lin and D. Wu, eds.), (Barcelona, Spain), pp. 404–411, Association for Computational Linguistics, July 2004.

[14] F. Boudin, "pke: an open source python-based keyphrase extraction toolkit," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, (Osaka, Japan), pp. 69–73, December 2016.

[15] W. DM, "Evaluation: From precision recall and f-factor to roc informedness markedness correlation," *Journal of Machine Learning Technologies*, vol. 2, p. 6, 2011.

[16] Z. Xu and J. Zhang, "Extracting keywords from texts based on word frequency and association features," *Procedia Computer Science*, vol. 187, pp. 77–82, 2021. 2020 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI2020.

[17] S. Rose, D. Engel, N. Cramer, and W. Cowley, *Automatic Keyword Extraction from Individual Documents*, pp. 1 – 20. 03 2010.

[18] A. Bellaachia and M. Al-Dhelaan, "Ne-rank: A novel graph-based keyphrase extraction in twitter," in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, pp. 372–379, 2012.

[19] W. Shi, W. Zheng, J. X. Yu, *et al.*, "Keyphrase extraction using knowledge graphs," *Data Science and Engineering*, vol. 2, no. 4, pp. 275–288, 2017.