Graph-based Model for Keyphrases Extraction from Arabic Text (GMKE)

Amirah Al Shammari[1], Abdullah Al Ghamdi[2]

[1]Department of Computer Science, College of Computer, Jouf University, Al jouf, Saudi Arabia
Email: *afmalshammari[at]ju.edu.sa*

[2]Department of Information Systems, College of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
Email: *aalmalaise[at]kau.edu.sa*

Abstract: ***Keyword extraction is an important step in several natural language processing and information retrieval applications, including text summarization and search engine optimization. Keywords include the most essential information characterizing the document's content. As the number of available documents increases, it is extremely difficult for a user to read each one in depth. Therefore, knowing the subject of the documents without performing an in-depth analysis is essential, and an automatic method of keyword extraction is required. Arabic research is still leaking in this area. In this study, we introduce a graph-based model for extracting the keyphrase using the K-mean clustering algorithm and TF-IDF ranking for single document text. Experiments were conducted using the ArabicKPE dataset. The experimental findings show that our model gives encouraging results compared to TF-IDF approaches in the Arabic KPE domain based on Recall, precision, and F-measure.***

Keywords: Arabic NLP, keyphrase extraction, graph -based, clustering

1. Introduction

Keyphrases are a collection of words or phrases that provide a brief summary of a text. They are critical for several aspects of Natural Language Processing (NLP), such as Document Clustering (DC), Information Retrieval (IR), Text Mining (TM), and Classification[1], [2], [20], [21].Automatic Keyphrase Extraction (AKE) is the process of automatically determining the most common and significant words and phrases that characterize the subject of a document. Such keyphrases are regarded as comprehensive and statistically significant. AKE facilitates a more rapid and accurate search for certain document and identifies it among several others. Moreover, it can help enhance several natural language processing procedures [3].

Manual execution of keyphrase extraction is complicated due to data dimension growth. The most common and pervasive issue is that it is extremely time-consuming. Therefore, several approaches are used to conduct AEK. There are two fundamental types of machine learning: supervised and unsupervised, which can be either statistical or linguistic. Our model is one of the unsupervised statistical methods.

Most of these techniques implement the AKE in three steps: First, represent the text using an appropriate model, then extract a list of potential keyphrase candidates. Last, use the appropriate strategy to determine and choose the keyphrases that are most appropriate.

Although there are hundreds of thousands of algorithms and methods available for AKE approaches that are state-of-the-art pose a great deal of challenge [4], [5]. One of these challenges is to automatically extract from a text a limited set of keyphrases that can accurately describe the context and can simplify the rapid handling of information, particularly in situations that occur in real time[6]. This indicates that accurate solutions and models designed for high speeds are necessary for the extraction process today.

Another challenge is that the keyphrases that were extracted cover the most important aspects of the text. As there is no guarantee that the keyphrases will be extracted from the text's main subtopics. As discussed in [7], [8], and [9], many clustering-based approaches have been proposed to handle this challenge. A third challenge is that you have to think about how the sentence will affect the extract. It is necessary for the sentence selected for the keypresses to be the most significant one in the text. As mentioned in [5], [8], and [10], several strategies were presented to address this difficulty.

To address these limitations, researchers are now developing new, more effective models for this objective using a variety of methods and algorithms, whether for data representation or for finding the most appropriate words or phrases that represent the full text. Our approach is suggested to overcome these issues, which can be summarized as follows:

1) Presenting the text using a graph data structure.
2) Clustering of sentences to produce clusters of sentences that present subtopics of text in a single document.
3) Apply the statistical method TF-IDF to rank the most frequent key phrases/keywords in each cluster.
4) Select K of key phrases/keywords from each cluster.

The sections of this work are structured as follows: The problem statement, research questions, and objectives are illustrated in Section II. The discussion of the related work in Section III, our approach presented in Section IV, and Section V illustrating the experiments and results. Finally, Section VI provides a conclusion and future work.

2. Problem statement and objectives

A. Problem Statement:

Keyword extraction is a technique that can be used to improve the efficiency and effectiveness of various text processing tasks. It can be employed in various life applications, such as Information Retrieval, Text Summarization, Text Classification and Sentiment Analysis. Due to 5% of the internet users speaking Arabic, and with the significant growth of the volume of Arabic text on the Internet and in digital libraries. Introducing efficient approaches to extract keywords and keyphrases is now a necessity, especially in the domain of Arabic text. In spite of the availability of numerous approaches in the field, they still share a common disadvantage as the inability to cover the subtopics of the text and represent the relation between the words that are neighboring one another.

B. Research Questions

The research questions of our paper are:

- In what ways can clustering methods effectively address the subtopics present in textual data?
- What is the potential of utilizing a graph-based approach to extract key phrases from Arabic text?
- How can the combination of clustering and graph representation techniques be used to improve the efficiency of extracting keyphrases from Arabic text compared to traditional methods?

C. Objectives:

Based on the research questions that are declared in Subsection b, the objectives of our proposed approach are:

- Evaluate the effectiveness of a graph-based K-means clustering algorithm to cover the subtopics of the text.
- Assess the potential of graph-based approaches for effectively extracting keyphrases from Arabic text.
- Evaluate the efficiency of the proposed framework in terms of accuracy and computational cost.

3. Literature Review

GDREK [11] used the Graph-based Growing Self-Organizing Map (G-GSOM) for clustering the sentences and the Density Peaks (DP) algorithm for ranking; however, the two algorithms can be used to cluster the sentences and rank the sentences in parallel. The G-GSOM does not perform well with small datasets because it grows cluster by cluster. It is hard for the DP algorithm to identify clusters within which points share the same densities generally; usually, sentences within one document share many words and phrases, which make them have similar density. In our proposed approach, using K-means with a graph representation [12] of text as a clustering algorithm and using a simple statistical method (TF-IDF) as in [13] to detect the most frequent terms in each document will handle these problems in GDREK.

Suleiman et. al, [4]proposed an approach to extract key phrases, in which the text is represented as Word2vec, which skipped the physical relationships among the words. Word2vec suffers from high time complexity for constructing the matrix of words. For one word within 1000 words, we need the measuring of relation with 999 words. Word2vec can produce two different values for two similar words. In addition, their proposed approach does not cover the subtopics of the text.

AAKE [14] represented the text as a word list (Bag of Words -BoWs-) which skipped any relationships among the words. AAKE does not cover the subtopics of text. AAKE is based on nine features for the words, two of them (term frequency (TF) and sentence frequency (SF)) can enlarge the probability of extracting words due to their frequency. However, these words are not keywords, but they are popular used in the text as stopwords. ML approaches rely on vast, annotated text corpora, which are not always accessible.

Campos et. al, [5]introduced an approach for automatic keyword extraction from a single document. It does not cover the subtopics of the text. It skipped any relationships among the words due to its representation of the text as aBoWs. To detect keywords with more than one word, they applied an additional step using a sliding window of 3-g, which consumed additional time.

RVA [7] is a local word vector-guided keyphrase extraction approach that does not cover the subtopics of text. RVA is only used for single documents; however, applying it to multiple documents would be extremely time-consuming.

Awajan proposed an approach for keyword extraction from Arabic documents using term equivalence classes [8]. It does not have the ability to cover the subtopics of text. And it skipped any relationships among the words.

In the following, Table 1 summarizes how our proposed model handles the limitations of the literature.

Previous work	Limitations	How the proposed model will cover those limitations
[11]	• JDREK used the Graph-based Growing Self-Organizing Map (G-GSOM) for clustering the sentences and the Density Peaks algorithm for ranking; however, the two algorithms can be used to cluster the sentences and ranked the sentences in parallel. • The G-GSOM does not perform well with small dataset that it grows cluster by cluster. • It is hard for the DP algorithm to recognize clusters within which points share the same densities generally. The sentences within one document usually shared many words and phrases, which made them have similar density.	• Using K-means with a graph representation [12] of text as a clustering algorithm. • Using a simple statistical method (TF-IDF) as in [13] to detect the most frequent terms in each document.

[4]	• They represented the text as Word2Vec, which skipped the physical relationships among the words. • Word2vec suffers from high time complexity for building the matrix of words. For one word within 1000 words, we need the measuring of relation with 999 words. • Word2vec can produce two different values for two similar words. • Does not cover the subtopics of the text.	• The proposed approach will represent the text based on graph structure to save the relationships among the words and detect the phrases. • The proposed approach will stem the words to their stems and represent them once. • The proposed approach will cluster the sentences and then extract the keywords and keyphrases from each cluster. • The proposed approach uses the TF-IDF method, which gains more information from longer documents compared to the embedding method.
[14]	• AAKE represented the text as a wordlist (Bag of Words) which skipped any relationships among the words. • AAKE does not cover the subtopics of text. • AAKE is based on nine features for the words, two of them (term frequency (TF) and sentence frequency (SF)) which can enlarge the probability of extracting words due to their frequency; however, these words are not the keywords, but they are commonly used in the text as the stopwords. • ML approaches rely on vast, annotated text corpora, which are not always accessible.	• The proposed approach will represent the text based on graph structure to save the relationships among the words and detect the phrases. • The proposed approach will cluster the sentences and then extract the keywords and keyphrases from each cluster. • The proposed approach will represent the TF-IDF as a statistical method that skips the words with high frequency; however, it will select the most frequent weighted words in the document's text.
[5]	• Does not cover the subtopics of the text. • Deals with the text as a Bag of Words which skipped any relationships among the words. • To detect the keywords with more than one word, they applied an addition step using a sliding window of 3-g, which consumed additional time.	• The proposed approach will represent the text based on graph structure to save the relationships among the words and detect the phrases. • The proposed approach will cluster the sentences and then extract the keywords and keyphrases from each cluster.
[7]	• RVA does not cover the subtopics of text. • RVA can be applied for single document; however, applied it with multi-documents required extreme time complexity.	• The proposed approach will represent the text based on graph structure to save the relationships among the words and detect the phrases. • The proposed approach will cluster the sentences and then extract the keywords and keyphrases from each cluster. • The proposed approach has the ability to extract the keywords from text in general (single or multiple).
[8]	• Use Term Equivalence Classes • Does not cover the subtopics of the text. • Deals with the text as a Bag of Words (BoWs) which skipped any relationships among the words.	• Our proposed approach has the ability to handle these limitations due to the representation of text using graphs and the clustering for sentences.

4. Methodology

GMKE is based on three main stages to extract the key phrases (KPs) and key words (KWs) of any text, as illustrated in Figure 1, which are: 1) text representation based on a graph, 2) sentence clustering using a graph-based K-means algorithm, and 3) selection of the most frequent key phrases and key words by applying the TF-IDF method from each resultant cluster.

In the following subsections, an explanation for each stage will be introduced.

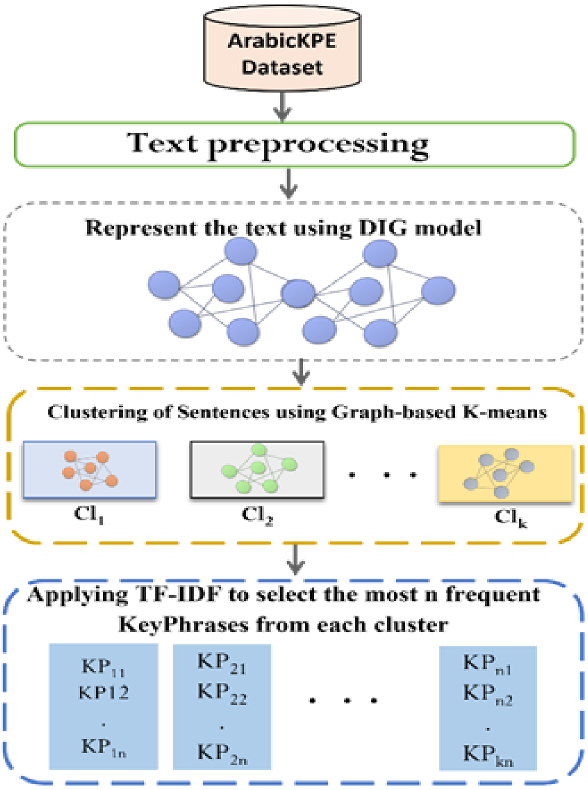

ArabicKPE Dataset
Text preprocessing
Represent the text using DIG model
Clustering of Sentences using Graph-based K-means
Cl_1
Cl_2
. . .
Cl_k
Applying TF-IDF to select the most n frequent KeyPhrases from each cluster
KP_{11}
KP12
.
KP_{1n}
KP_{21}
KP_{22}
.
KP_{2n}
. . .
KP_{n1}
KP_{n2}
.
KP_{kn}

Figure 1: GMKE Model

A. Text Preprocessing:

In Text Mining, Natural Language Processing, and Information Retrieval, pre-processing is a crucial and essential stage. Due to the fact that text often contains some special formats like number formats, date formats, and the most common words that are unlikely to help extract keyphrases (such as prepositions, articles, and pronouns), these can be eliminated. This step contains many tasks, such as text normalization, tokenization, the removal of stop words, and stemming. To perform GMKE using the ArabicKPE dataset [9], the document text is preprocessed to represent the text in the form of separated words. Preprocessing tasks include removing Arabic diacritics, normalizing the various forms of Arabic letters into a single shape as the Alef letter [آ,إ,أ]to [ا]. Lastly, using the Stanford CoreNLP Toolkit [10], segment the text into single tokens. The texts are then separated into sentences, and the tokens are associated with the word embedding representation.

B. Text representation:

Many approaches are proposed to represent the text for NLP tasks, but the graph data structure is the most effective to detect the relationships between words in order to extract the phrases from the text with any length. GMKE is based on the Document Index Graph (DIG) model [15]. DIG is used by many previous approaches to represent the text as in [16],[17],[18], and [19]. Using DIG, each word will be represented as one unique node (not repeated). These nodes have full data about each word including its number of occurrences, positions in text, the next words to it, and their importance in each position. This structure enables any approach to extract the phrases which means the relationships between words are well detected. In DIG, while building the accumulative graph, the shared phrases between all sentences are detected, which requires less time and space. These shared phrases are stored in a matrix of list data structure. Therefore, by the end of graph construction, all the shared phrases will be ready, and the similarity matrix has already been calculated and stored. According to the representing document structure, the DIG is a directed graph G = (V, E), where V is the vertices of graph and represent the words of text while E are the edges of graph which represent the relationships between words.

C. Sentence clustering:

In this stage, all the sentences of the document will be clustered into k clusters. Each cluster represents a subtopic in the document. From each cluster, GMKE will select several key phrases and key words. Clustering is done using the k-means algorithm based on a graph representation of the text. Each cluster will be initialized by a sentence (sub-graph) from the graph of the document.

Each sub-graph of a sentence is extracted from an accumulative graph. Then, it compares all existing clusters of sentences to find the most similar one. The first cluster is created and initiated by the first sentence in the text corpus, and it is assigned to the cluster as one vertex.

This stage results in several clusters, where each of them contains many similar sentences that represent a sub-topic in the text. The number of clusters (k) is defined based on experiments and according to the nature of the dataset.

D. Select the frequent keyphrases and keywords:

In this stage, GMKE builds the matrix of all words and phrases with length three, then calculates the term frequency (TF) or inverse document frequency (IDF) (TF-IDF) for each of them. TF-IDF is a numerical statistic used to reflect how important a word is to a sentence in a collection or corpus of sentences or documents. The term frequency (TF) represents the number of times a word appears in a text, while the inverse document frequency (IDF) is a measure of how rare the word is across the entire sentence collection. A high TF-IDF score indicates that a word is both frequent in a particular sentence and rare across the whole document sentences.

The frequency of a term can be computed using the formula as follows: TF = (Number of times the phrase occurs in the sentence) / (Total number of terms in the document).

The IDF can be determined using the following form: IDF = log(N / df), where N is the total number of sentences in the document and df is the number of sentences in which the term appears. The logarithm function is used to scale the values, as the difference in frequencies between sentences can be very large.

Therefore, the best value of TF-IDF will depend on the specific use case and the desired outcome. A higher value indicates a higher importance of the word in the specific document, while a lower value indicates that the word is common or not very informative.

As a result, the matrix of key words and key phrases will introduce the rank of them for each cluster, and then GMKE will choose the key word or key phrase with the highest value in each cluster. These are the key words and key phrases in the document. If the number of extracted key words and key phrases is less than three, GMKE will select more than one from each cluster.

5. Experiments and Results

In this section, the experiments and results of GMKE are presented and discussed. To develop the proposed approach, we depend mainly on the Python programming language. Python has many libraries to perform Natural Language Processing (NLP) operations, which are critical to the majority of upcoming data science.

a) Measures

To evaluate the performance of the GMKE, a series of experiments on a corpus of Arabic documents are conducted and measured using precision, recall, and F1-score. More details are illustrated below:

1) **Recall:** It is the number of correct extracted sentences divided by the number of sentences that should have been returned correctly. The value of recall is calculated using equation 4.1.

$$Recall = \frac{Returned\ sentences\ \cap Correct\ sentences}{Correct\ sentences} \tag{4.1}$$

2) **Precision:** It is the number of correctly extracted sentences divided by the number of all returned results. The value of recall is calculated using equation 4.2.

$$Precision = \frac{Returned\ sentences\ \cap Correct\ sentences}{Returned\ sentences} \qquad (4.2)$$

3) **F-measure** is also known asF1-score and F score. It is a metric that combines Precision and Recall using equation 4.3.

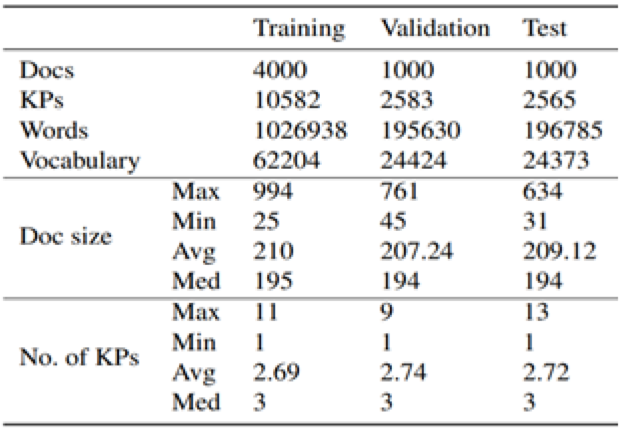
$$F - meaure = 2\,.\,\frac{Precesion\ .\ Recall}{Precesion +\ Recall} \qquad (4.3)$$

Our results illustrate the efficacy of the suggested method for extracting Keyphrases from Arabic documents, with encouraging results compared to the state-of-the-art methods. In this section, we will provide a detailed description of the dataset, the results obtained, and a comprehensive analysis of the results.

b) Dataset

The used dataset is ArabicKPE [9] with the same splits that were supplied by the authors: 4887 documents for training, 944 for validating the model, and 941 documents used for testing. Table 2 illustrates full statistical details about ArabicKPE.

Table 2: Statistical details about ArabicKPE



		Training	Validation	Test
Docs		4000	1000	1000
KPs		10582	2583	2565
Words		1026938	195630	196785
Vocabulary		62204	24424	24373
Doc size	Max	994	761	634
	Min	25	45	31
	Avg	210	207.24	209.12
	Med	195	194	194
No. of KPs	Max	11	9	13
	Min	1	1	1
	Avg	2.69	2.74	2.72
	Med	3	3	3

The ArabicKPE dataset is a collection of Arabic documents specifically created for keyphrase extraction. It is one of the largest and most extensively used benchmark datasets for evaluating the accuracy of keyphrase extraction methods in the Arabic language. The dataset consists of a diverse set of documents covering a range of topics, including news articles, scientific papers, and legal texts.

The documents in the ArabicKPE dataset are annotated with keyphrases, which are manually extracted by expert annotators. This provides a gold standard for evaluating the performance of keyphrase extraction algorithms. The keyphrases in the dataset cover a wide range of lengths and variations, making it suitable for testing the robustness and versatility of key-phrase extraction methods. Additionally, the dataset is divided into a training set and a test set, allowing for the evaluation of key-phrase extraction algorithms using cross-validation techniques. Overall, the ArabicKPE dataset provides a valuable resource for researchers and practitioners working on keyphrase extraction in the Arabic language and is widely used in the evaluation and comparison of keyphrase extraction algorithms.

6. Results

GMKE and TF-IDF are compared based on the Precision, Recall and F1-Score. The results shown in Table 3were obtained by extracting the top 3 key phrases and key words from the document and the top 5 key phrases and key words.

Table 3: Comparison results

	Top 3			Top 5		
Approach	Precision	Recall	F1-Score	Precision	Recall	F1-Score
GMKE	0.28	0.37	0.32	0.19	0.43	0.26355
TF-IDF	0.17	0.26	0.2	0.085	0.26	0.13

As illustrated in Figure 2 and Figure 3, GMKE overcomes the TF-IDF due to many reasons as:

- The clustering of sentences improves the factor of variety in the extracted keyphrases and keywords.
- The representations of text based on graph which enhances the opportunity to detect keyphrases.

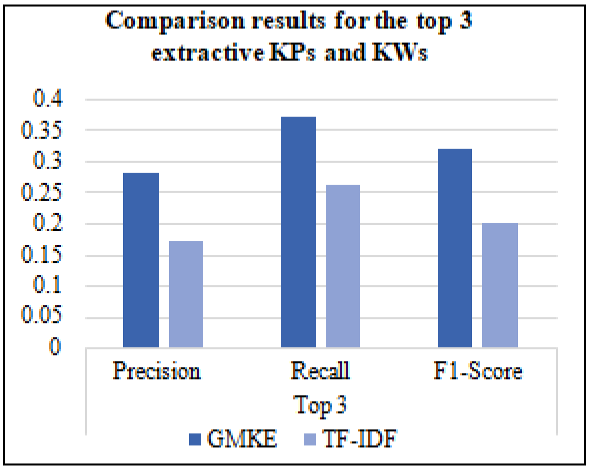


Figure 2: Results for the top 3 extractive KPs and KWs

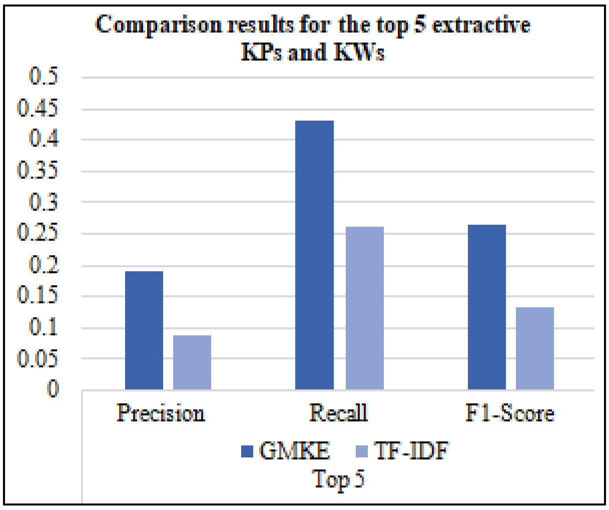


Figure 3: Results for the top 5 extractive KPs and KWs

By noting the accuracy of two approaches, it is worth noting that it decreased as the number of extracted KPs and KWs increased. This means that the relation between the number of extracted KPs and KWs and accuracy is an inverse relationship.

7. Conclusion and Future Work

GKME is a model for keyphrase extraction that uses a graph-based K-mean clustering algorithm and TF-IDF ranking for single document text. It gives many aspects of successfully extracting keywords and keyphrases. Experiments are performed on the ArabicKPE dataset. GMKE gives encouraging results compared to other approaches based on Recall, precision, and F-measure.

Here, some techniques and suggestions for modifications are proposed to improve the performance of GMKE in the future:

1) Using more features about sentences and ranking them before extracting KP.
2) Conduct GMKE on other datasets that support multiple sub-topics.

References

[1] S. Abdulateef, N. A. Khan, B. Chen, and X. Shang, “Multidocument Arabic Text Summarization Based on Clustering and Word2Vec to Reduce Redundancy,” *Information*, vol. 11, no. 2, p. 59, Jan. 2020, doi: 10.3390/info11020059.

[2] L. Abualigah, M. Q. Bashabsheh, H. Alabool, and M. Shehab, “Text Summarization: A Brief Review,” in *Studies in Computational Intelligence*, vol. 874, Springer, 2020, pp. 1–15. doi: 10.1007/978-3-030-34614-0_1.

[3] Ahadi, A., Singh, A., Bower, M., & Garrett, M. (2022, March 15). Text Mining in Education—A Bibliometrics-Based Systematic Review. Education Sciences, 12(3), 210. https://doi.org/10.3390/educsci12030210

[4] D. Suleiman, A. A. Awajan, and W. Al Etaiwi, “Arabic Text Keywords Extraction using Word2vec,” 2019 2nd International Conference on New Trends in Computing Sciences, ICTCS 2019 - Proceedings, Oct. 2019, doi: 10.1109/ICTCS.2019.8923034.

[5] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, “A text feature based automatic keyword extraction method for single documents,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10772 LNCS, pp. 684–691, 2018, doi: 10.1007/978-3-319-76941-7_63/COVER.

[6] M. R. Alfarra, A. M. Alfarra, and J. M. Alattar, “Graph-based Fuzzy Logic for Extractive Text Summarization (GFLES),” Proceedings - 2019 International Conference on Promising Electronic Technologies, ICPET 2019, pp. 96–101, Oct. 2019, doi: 10.1109/ICPET.2019.00025.

[7] E. Papagiannopoulou and G. Tsoumakas, “Local word vectors guiding keyphrase extraction,” Inf Process Manag, vol. 54, no. 6, pp. 888–902, Nov. 2018, doi: 10.1016/J.IPM.2018.06.004.

[8] A. Awajan, “Keyword Extraction from Arabic Documents using Term Equivalence Classes,” ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), vol. 14, no. 2, Apr. 2015, doi: 10.1145/2665077.

[9] M. Helmy, R. M. Vigneshram, G. Serra, and C. Tasso, “Applying Deep Learning for Arabic Keyphrase Extraction,” Procedia Comput Sci, vol. 142, pp. 254–261, Jan. 2018, doi: 10.1016/J.PROCS.2018.10.486.

[10] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. Mcclosky, “The Stanford CoreNLP natural language processing toolkit,” aclanthology.org, pp. 55–60.

[11] M. Alfarra, A. M. Alfarra, and A. Salahedden, “Graph-based density peaks ranking approach for extracting keyphrases (GDREK),” IEEE 7th Palestinian International Conference on Electrical and Computer Engineering, PICECE 2019, Mar. 2019, doi: 10.1109/PICECE.2019.8747175.

[12] R. M. Albadrani, M. A. Al-Hagery, M. Tahar, and B. Othman, “A Proposed Clustering Technique for Arabic Text Summarization,” International Journal of Science and Research, doi: 10.21275/SR221009042717.

[13] M. R. Alfarra and A. Alfarra, “Graph-Based Technique for Extracting Keyphrases in a Single-Document (GTEK),” Proceedings - 2018 International Conference on Promising Electronic Technologies, ICPET 2018, pp. 92–97, Nov. 2018, doi: 10.1109/ICPET.2018.00023.

[14] E. H. Omoush and V. W. Samawi, “Arabic Keyword Extraction using SOM Neural Network,” INTERNATIONAL JOURNAL OF ADVANCED STUDIES IN COMPUTER SCIENCE AND ENGINEERING IJASCSE, vol. 5, 2016.

[15] K. M. Hammouda and M. S. Kamel, “Efficient phrase-based document indexing for web document clustering,” IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 10, pp. 1279–1296, 2004, doi: 10.1109/TKDE.2004.58.

[16] S. Hingmire et al., “Document Classification by Topic Labeling Document Classification by Topic Labeling,” no. July, 2013.

[17] M. F. Hussin, M. R. Farra, and Y. El-Sonbaty, “Extending the growing hierarchal SOM for clustering documents in graphs domain,” in Proceedings of the International Joint Conference on Neural Networks, 2008, pp. 4028–4034. doi: 10.1109/IJCNN.2008.4634377.

[18] Vijaya Shetty, S., Akshay, S., Shritej Reddy, B. S., Rakesh, H., Mihir, M., & Shetty, J. (2022). Graph-Based Keyword Extraction for Twitter Data. In Emerging Research in Computing, Information, Communication and Applications: ERCICA 2020, Volume 2 (pp. 863-871). Springer Singapore.

[19] M. Alfarra, A. M. Alfarra, and A. Salahedden, “Graph-based growing self-organizing map for single document summarization (GGSDS),” IEEE 7th Palestinian International Conference on Electrical and Computer Engineering, PICECE 2019, Mar. 2019, doi: 10.1109/PICECE.2019.8747236.

[20] H. K. Duan, M. A. Vasarhelyi, M. Codesso, and Z.

Alzamil, “Enhancing the government accounting information systems using social media information: An application of text mining and machine learning,” International Journal of Accounting Information Systems, vol. 48, p. 100600, Mar. 2023, doi: 10.1016/j.accinf.2022.100600.

[21] K. Thakur and V. Kumar, “Application of Text Mining Techniques on Scholarly Research Articles: Methods and Tools,” New Review of Academic Librarianship, vol. 28, no. 3, pp. 279–302, May 2021, doi: 10.1080/13614533.2021.1918190.

Author Profile

Amirah Al Shammari is a Lecturer in the Computer Science department at Jouf University, Saudi Arabia. She received her master’s degree in science in Informatics from Qassim University Saudi Arabia. She is currently studying Ph. D in King Abdulaziz University (KAU), Jeddah, Saudi Arabia. And her area of research is cloud computing, machine learning and data
mining and analytics.

Abdullah Saad AL-Malaise AL-Ghamdi is a Professor, Software & Systems Engineering and AI, and associated with Faculty of Computing and Information Technology (FCIT), King Abdulaziz University (KAU), Jeddah, Saudi Arabia. He received his Ph.D. degree in Computer Science from George Washington University, USA, in 2003. He is a member of the Scientific Council and holds the position as a Secretary General of Scientific Council, KAU. In addition, he is working as Head of Consultant’s Unit at Vice-President for Development Office, as a Consultant to Vice-President for Graduate Studies & Scientific Research, KAU. Previously, he has worked as Head of IS Department, Vice Dean for Graduate Studies and Scientific Research, and Head of Computer Skills Department at FCIT. His main research areas are Software Engineering and Systems, Artificial Intelligence, Data Analytics, Business Intelligence, and Decision Support System