Indonesian Journal of Electrical Engineering and Computer Science
Vol. 37, No. 1, January 2025, pp. 206~213
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v37.i1.pp206-213

A three-phase model to keyword detection in Arabic corpora

Driss Namly[1], Karim Bouzoubaa[2], Ridouane Tachicart[3]
[1]Department of Computer Science, Mohammed V University in Rabat, Rabat, Morocco
[2]Mohammadia School of Engineers, Mohammed V University in Rabat, Rabat, Morocco
[3]Faculté des Sciences Juridiques, Economiques et Sociales d'El-Jadida, Chouaib Doukkali University, ElJadida, Morocco

Article Info

Article history:

Received May 31, 2024
Revised Sep 3, 2024
Accepted Sep 7, 2024

Keywords:

Arabic language
Information retrieval
Keyword detection
Machine learning
Natural language processing

ABSTRACT

The exponential growth of Arabic text data in recent years has created an urgent demand for sophisticated keyword detection techniques that are specifically tailored to the nuances of the Arabic language. This study addresses the critical need for efficient tools capable of swiftly and accurately identifying keywords within a collection of Arabic documents, particularly when analyzing multiple documents in a corpus. To meet this challenge, we present a novel corpus specifically designed for keyword detection in Arabic texts, along with an innovative approach that integrates three distinct candidate keyword lists: a frequency-based list, a vector space model list, and a machine learning-based list. This hybrid methodology leverages the strengths of each technique, enabling a more comprehensive and effective keyword identification process. We conducted extensive experimental validation to assess the performance and computational efficiency of our proposed pipeline. The results demonstrate that our approach consistently achieves robust performance across a variety of domains, with evaluation metrics indicating F1-scores that consistently surpass 91%. Overall, this study contributes to the advancement of automated keyword detection in Arabic, paving the way for enhanced information retrieval and text analysis capabilities.

Corresponding Author:

Driss Namly
Department of Computer Science, Mohammed V University in Rabat
Avenue Ibn Sina B.P 765, Agdal, 10090, Rabat, Morocco
Email: d.namly@um5r.ac.ma

1. INTRODUCTION

The rapid growth of Arabic text data has created an urgent need for advanced tools capable of quickly and accurately detecting the most significant terms within a corpus [1]. Automated keyword detection [2], [3] is a crucial process for various applications, including document summarization [4], search engine optimization [5], and information retrieval [6]. The literature review reveals a variety of methodologies utilized in automated keyword detection solutions, including statistical [7], [8], linguistic [9], [10], machine learning [11]-[15], graph-based [16]-[18], hybrid approaches [19], and large language models [20].

However, existing solutions face several limitations, especially when dealing with the complexities of the Arabic language. Current automated keyword detection methods often lack linguistic understanding [3], failing to recognize connections between words that share the same concept. For example, the Arabic words "لَحَضَرْتُنَّهَا، وَسَنَحْضُرُ، أَحَضَرَا، فَلِتَحْضُرِي، والحَضَر" (you would have attended it, and we will attend, are they attend, and you let attend, and the urban) share the same concept, "حَضَر" (to attend), but automated methods might miss this connection. They also struggle with word sense disambiguation, as non-diacritized Arabic words can have multiple meanings depending on the context [21]. This can result in the inclusion of irrelevant keywords. For instance, depending on the context, the word "ولم" (wlm) can have different meanings such as

"وَلُمَّ" (and be insane), "وَلَمٌّ" (and a group), "وَلَمَّ" (and gather), "وَلَمْ" (and not), "وَلِمَ" (and why), "وَلْم" (belt), or "وَلِمَ" (to eat in a feast). Furthermore, automated methods also often prioritize frequent terms, potentially overlooking important but less common or specialized keywords [22]. Additionally, when an automated keyword detection method looks at a single document in isolation, some words may not arise as keywords. Indeed, when analyzing a corpus, these same words might appear across multiple documents, indicating their importance or relevance within that collection. Another limitation is that automated solutions can mistakenly identify insignificant words as keywords while failing to detect certain important ones. Moreover, the performance of automated keyword detection solutions can vary depending on the processed language [21]. While they may work well in English, their effectiveness in languages like Arabic may be compromised.

To address the constraints observed in previous research, a common strategy involves using a combination of keyword detection methods. Statistical methods can quickly generate a wide range of keywords, which can subsequently undergo two rounds of refinement to improve the quality and relevance of the detected terms. The initial refinement involves employing a vector space model, followed by a second refinement utilizing a machine learning approach on a newly developed corpus. This tripartite approach achieves a harmonious balance between efficiency and accuracy, offering a resilient resolution to overcome the inherent constraints of automated keyword identification. Particularly, it addresses the challenge of detecting a comprehensive list of keywords for an entire corpus, rather than a keywords list for each individual document in the corpus.

2. METHOD

The safar keywords extraction tool (SKET), shown in Figure 1, has four main steps: preprocessing, frequency-based list, vector space list, and machine learning list. The final keywords list is created by merging the three separate lists from frequency, vector space, and machine learning methods. This merging uses a logical "and" to pick candidates that appear in all three keyword lists. In the following section, we will explore the details of each step in SKET.

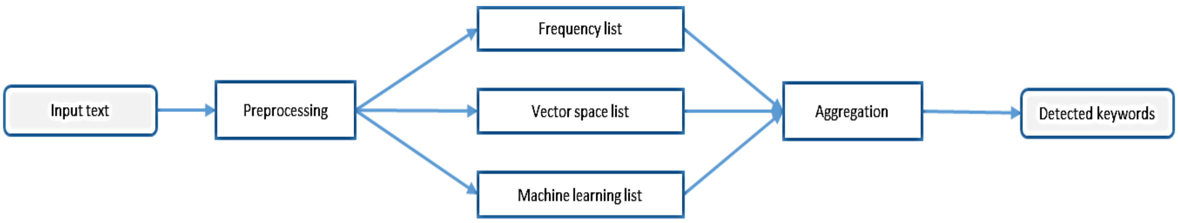

Figure 1. Proposed pipeline for keyword detection

2.1. Preprocessing

During this stage, the main objective is to prepare the data for pipeline utilization by removing noise from the original text. This process includes tokenization and removing punctuation marks, special characters, non-Arabic letters, and digits. Subsequently, stop-words are removed even though they are commonly found, as they usually make up 30 to 50% of the text data size and are not deemed as keywords. The tokens that remain are lemmatized, as it is advised to work with the lemma of the words rather than the stem, root, or the word itself.

The preprocessing steps are performed using SAFAR tools [23]. The preprocessing process includes several critical stages: first, the documents undergo tokenization, which breaks the text into manageable units. This is followed by normalization, where special characters, non-Arabic letters, and tokens that consist solely of digits or contain both digits and letters are removed. Subsequently, stop-words are eliminated to focus on the most meaningful terms, and the remaining tokens are lemmatized using the SAFAR lemmatizer [24]. This processing step plays a crucial role in addressing challenges associated with linguistic understanding and non-diacritized Arabic. This is because lemmatization brings together tokens with the same meaning under a single lemma, thereby enhancing linguistic understanding. Moreover, the lemmatizer used produces diacritized lemmas, further contributing to the overall effectiveness of the process. Once preprocessing is complete, we validate the keyword detection process through three experiments conducted on 80% of the corpus, while the remaining 20% is used for evaluation.

2.2. Frequency list

In the subsequent phase of SKET, a frequency-based approach is employed to generate a roster of potential keywords by considering their frequency. In order to achieve this, term frequency (TF) is utilized as a probability function for every lemmatized token. The tokens with the highest probability are recognized as potential keywords. The TF statistic gauges the appearance frequency of a token's in the corpus and is computed by dividing the number of occurrences of the token "f" by the total number of tokens "N" as follows:

$$TF = \frac{f}{N} \tag{1}$$

Consequently, in order to identify the initial list, unique lemmas are first identified and their frequencies in the whole corpus are counted, we sort them from high to low, then, we select the first 50 tokens as keyword's candidates.

2.3. Vector space list

In order to prevent the detection of insignificant words, and to prevent the detection in single document in isolation, we improve the results obtained from the frequency-based step by integrating a vector space approach into the SKET framework. This step aims to refine the detection of candidate keywords by pinpointing tokens that exhibit a consistent distribution throughout the entire corpus. Tokens that are frequently encountered but are confined to isolated paragraphs within a single document are deemed inappropriate as keywords for the overall corpus. By employing this strategy, we seek to improve the precision and relevance of the keywords extracted from the textual data.

From an organizational standpoint, a corpus consists of a collection of diverse documents, each containing multiple paragraphs. In this framework, a token is considered uniformly distributed across the corpus if it appears in the majority of both documents and paragraphs. To operationalize this concept, we apply the conventional vector space model to represent the occurrence of tokens instead of entire text documents. In this revised model, each token is conceptualized as a vector with distinct components indicating its presence or absence in each paragraph. To identify potential keywords, we have established a prototype token that is present in every paragraph. The similarity between a token and the prototype is assessed by measuring the distance in this vector space, utilizing cosine similarity to compare the two vectors. Tokens that exhibit proximity to the prototype are identified as candidate keywords.

During the construction of the model, we estimate the value of a response variable based on two explanatory variables, X_1 and X_2, which represent the presence rate in a specific section of a document and the presence rate across paragraphs, respectively. The resulting function is formulated as a weighted sum of these explanatory variables by (2).

$$P(X) = w_1 \times X_1 + w_2 \times X_2 \tag{2}$$

In order to determine the weights w_1 and w_2, the significance of each variable is evaluated by testing different weight combinations and analyzing their impact on the inferred function. The weight pairs tested include [0.5, 0.5], [0.3, 0.7], [0.7, 0.3], [0.1, 0.9], and [0.9, 0.1] for [w_1, w_2]. Through this experimentation, it is found that the maximum average variation in the inferred function is 2.33%, observed between the pairs [0.5, 0.5] and [0.7, 0.3]. As a result, it is decided to allocate identical weights to every explanatory factor. The final inferred function is a sum of the explanatory variables with equal weights by (3).

$$P(X) = \frac{X_1 + X_2}{2} \tag{3}$$

To form our list of candidates, we organize the tokens based on their probabilities in a descending manner. Then, we select the top 50 tokens as potential keywords.

2.4. Machine learning list

In this stage, the efficacy of the frequency and vector space methodologies of SKET is enhanced through the application of a classification algorithm, which aids in identifying this third set of keyword candidates. This stage aims to avoid the omission of crucial but less common or specialized keywords by integrating domain-specific knowledge. The integration of domain expertise is expected to enhance the accuracy and efficiency of the keyword identification process across different datasets.

To determine the most suitable algorithm for SKET, various classifiers are evaluated based on their accuracy and execution time (ET). Considering ET is essential as keyword detection tasks often involve realtime processing or handling large amounts of data. A metric is calculated that combines ET and proportion of errors (PE = 1-Accuracy). The inverse efficiency score (IES) is defined as the ET adjusted for the number of errors made. A lower value of IES indicates superior performance.

$$IES = \frac{ET}{1-(PE)} \tag{4}$$

Table 1 presents the results of the classifiers assessed using a constrained dataset comprising 10 news articles obtained from the hespress.com news platform. This dataset included a total of 46,924 tokens, which were manually annotated with keyword tags. Following this preliminary proof of concept, the findings can be extrapolated to a more extensive corpus. The IES results offer the possibility of selecting either Naïve Bayes or Logistic Regression as the algorithm of choice. Nevertheless, given the significant difference in accuracy and the negligible variation in ET, it is clear that Logistic Regression emerges as the most appropriate option to classify tokens into two distinct classes, namely keyword and non-keyword.

Thus, to generate the third list of keyword candidates, we utilize the logistic regression algorithm from the WEKA library. The corpus is categorized into two classes: "0" for non-keywords and "1" for keywords. Upon loading the data file, a "StringToWordVector" filter is applied to convert strings into N-grams, enabling a more nuanced analysis of the text. The model is then constructed using the logistic regression classifier. The last stage of SKET entails merging the three candidate lists into a unified list through a logical "and" operation, which detects tokens that exist in all three lists. To gauge the efficacy of SKET, the subsequent section outlines the experimental results for evaluating its performance.

Table 1. The outcomes of the comparison between the classifiers

Algorithm	Accuracy (%)	Execution time (s)	IES
Bayesian logistic regression	100	54.24	54,24
J48 decision tree	96.88	1637.55	1 690,75
Logistic regression	100	11.23	11,23
Naïve Bayes	89.3487	9.81	10,97
Random forest	99.95	213.12	213,11
Support vector machine	100	121.98	121,98

3. RESULTS AND DISCUSSION

The section starts with a detailed overview of the corpus utilized. Subsequently, an analysis of the experiment carried out and its findings is presented. This is then followed by a comparative analysis and an overarching discussion.

3.1. The CAKE corpus

Our aim is to evaluate our research by utilizing corpora from prior studies; however, unfortunately, those corpora are unavailable except for the ones listed below:

- Arabic keyphrase extraction corpus (AKEC) consisting of 160 Arabic documents from a variety of sources and their keyphrases collected using a large-scale crowdsourcing experiment;
- Arabic dataset for automatic keyphrase extraction (ADAKE) contains 400 documents (1,708,168 tokens) covering 18 topics. Ten Keyphrases per document are extracted a reader;
- Arabic Wikipedia Corpus (AWC) composed of a dump file of 100 Arabic Wikipedia pages. Keywords are obtained from the keyword meta-tag associated with each page.

Thus, assessing SKET through the use of AKEC, ADAKE, or AWC is inappropriate due to individual document limitation. In contrast, SKET is tailored to identify a singular list of keywords applicable to the entire corpus domain. As a result, we have created the Corpus for Arabic keywords extraction (CAKE) to tackle this issue. It consists of 2778 news articles, encompassing approximately 4 million tokens in four different categories: Art, Economy, Politics, and Sport. The corpus was sourced from a news website that was used before and keywords were assigned semi-automatically using lexicons from the specific domains outlined in Table 2. Figure 2 shows part of CAKE, highlighting how it is divided into domains. Each domain has a group of articles and a list of keywords. Every article includes a title and a text.

Table 3 displays a variety of statistics comparing the existing corpora with the developed one. The Arabic Wikipedia Corpus has a moderate number of documents, followed by AKEC, Arabic Dataset, and CAKE, totaling 2778 documents. The keyword statistics are of utmost importance for our research. AKEC, Arabic Wikipedia Corpus, and Arabic Dataset provide a set of keywords for each document, while CAKE offers a single list of keywords for all documents in each domain. CAKE will be accessible for free to researchers.

Table 2. Lexicons used for tagging process

Lexicon	author	Year	Domain	URL
مسرد مصطلحات كرة القدم	سمير الشناوي	2007	Sport	https://www.noor-book.com
The Sports Dictionary	-	-	Sport	https://www.lexicool.com
Arabic sports & hobbies vocabulary	desert-sky	2007	Sport	https://arabic.desert-sky.net
قاموس المصطلحات التجارية والاقتصادية والمالية	منى جريج	2000	Economy	https://www.mktbty-eng.com
مصطلحات العملات و مصطلحات التداول	Arabic trader	-	Economy	https://www.arabictrader.com
Arabic media and the art vocabulary	desert-sky	2007	Art	https://arabic.desert-sky.net
مصطلحات فنية و تشكيلية	arch dictionary	2008	Art	https://archdic.blogspot.com
معجم المصطلحات السياسية	معهد البحرين للتنمية السياسية	2014	Politics	https://www.bipd.org
قاموس المصطلحات المدنية والسياسية	صقر الجبالي، أيمن يوسف، عمر رحال	2014	Politics	https://www.shams-pal.org
قاموس مصطلحات اقتصادية	محمد مجدي أحمد	2019	Economy	https://www.noor-book.com

```
<AKEC>
    ...
    <Category="Economy">
    ...
        <Article id="19">
            <Title>إنييستا و"البارصا" .. غيّاب لجودة اللّعب وطموح لاستئناف التألّق</Title>
            <Text> أقر أندريس إنييستا، لاعب وسط وقائد نادي "إف. سي. برشلونة" لكرة القدم، بأن هذا العام لم يكن جيدا بالنسبة للفريق؛ "لأننا لم نفز بالألقاب التي كنا نطمح إليها"، وفق تعبيره الذي شدد، أيضا، على أن "الجماهير يجب أن تعطي أهمية لتتويج الفريق، للعام الثالث على التوالي بلقب الكأس". ...
            </Text>
        </Article>
    ...
    <Keywords>فَرِيق, كُرَة, مُنْتَخب, بُطُولَة, نِهَايَة, رِيَاضِي, اتِّحَاد, فَوْز, بَطَل</Keywords>
    </Category>
    ...
</AKEC>
```

Figure 2. Extract of CAKE

Table 3. Corpora statistics

Corpus	Domains	Nbr of Documents	Avg nbr of tokens	Nbr of tokens	Nbr of keywords
CAKE	4	2,778	1,440	4,000,000	235
AKEC	9	160	757	121,074	9,949
Arabic Dataset	18	400	4,270	1,708,168	4,000
Arabic Wikipedia Corpus	-	100	2,230	2,123,182	7,606

3.2. Evaluation results

In the experiment, we adhere to SKET, leveraging a 6-core Windows computer equipped with an Intel(R) Core i7 2.6 GHz CPU and 16 GB memory. To evaluate the performance of our keyword detection approach, each token in the test set—comprising 20% of our corpus—is categorized into one of four classifications: true positive, false positive, true negative, or false negative. A token is designated as a true positive when it is both predicted and tagged as a keyword.

Table 4 presents the weighted average metrics for SKET, including Precision, Recall, and F1-score. Since F1-score exceeds 91% for all domains, this metrics evaluation provides a comprehensive overview of the SKET's effectiveness in accurately identifying keywords. Additionally, Table 4 includes two important time metrics: Tb, which indicates the time taken to build the model (measured in hours), and Tp, which represents the time taken to predict a token (measured in milliseconds).

Table 4. The obtained metrics' outcomes

Domain	Precision	Recall	F-score	Tb (hr)	Tp (ms)
Arts	0.945	0.941	0.938	1.61	7.066
Economy	0.930	0.920	0.918	1.77	7.042
Politics	0.951	0.948	0.946	1.76	7.091
Sport	0.984	0.983	0.983	1.65	7.006

3.3. Comparative analysis

The second experiment aimed to evaluate the performance of SKET through a comparative analysis with prior studies. Recent research [25]-[29] has indicated that transformer-based models, outperform traditional statistical, machine learning, and deep learning techniques in various tasks. To facilitate this comparison, we downloaded and implemented the Llama-2-7b, Llama-2-13b, Llama-2-70b, and Mistral-7B-v0.1 models.

However, our attempts to utilize these LLMs for keyword identification from the corpus encountered challenges due to limitations in their context size parameters. Specifically, Llama-2 models are constrained to a context size of 4,000 tokens, while the Mistral model can handle up to 8,000 tokens. This context size defines the maximum number of tokens that can be processed in a single input, including both the prompt and the response. Given that the CAKE corpus is relatively large, this limitation hindered the effectiveness of the LLMs in our comparative study, ultimately making the evaluation of SKET against these models less relevant.

Nevertheless, we compare SKET with some online LLMs using three randomly selected documents from CAKE containing fewer than 4000 tokens. As illustrated in Table 5, SKET displays robust precision and a high F1-score, positioning it as an exceptional model in terms of reliability for positive predictions. On the other hand, models like GPT-3.5 and Mixtral offer a more balanced performance, albeit with lower precision and F1-scores.

Table 5. The comparison results

Model	Precision	Recall	F1-score
GPT-3.5	0.6481	0.2734	0.3846
Gemini	0.5068	0.2891	0.3682
Llama2-70b	0.2000	0.0781	0.1124
Mixtral	0.3958	0.2969	0.3393
SKET	0.8889	0.7500	0.8156

3.4. Discussion

This study investigated the critical need for efficient tools capable of swiftly, automatically, and accurately identifying keywords within a collection of Arabic documents. While earlier studies have explored automatic keyword identification, they have not addressed the single document limitation. SKET, the proposed tool, consists of several components including preprocessing steps, frequency analysis, vector space modeling, and logistic regression, which collectively generate three lists of keyword candidates. These lists are subsequently integrated to form the final compilation of keywords.

The evaluation results for keyword detection on the CAKE corpus utilizing the proposed SKET tool indicate that the F1-score consistently exceeds 91% across all domains, reaching a maximum of 98.3% within the sports domain, thereby underscoring the significant effectiveness of our methodology. Additionally, SKET demonstrates remarkable efficiency, with model construction taking less than 1.77 hours and token prediction occurring in approximately 7 milliseconds. This assessment confirms both the effectiveness and computational efficiency of SKET.

Comparative analysis with existing LLMs reveals that SKET outperforms these models in terms of Precision and F1-score. While transformer-based models demonstrate potential across various applications, their context size constraints present significant challenges when applied to larger datasets. This constraint highlights the importance of utilizing a specialized tool such as SKET, which is specifically designed for efficient keyword identification within large Arabic text collections. By addressing these challenges, SKET provides a more reliable solution for researchers and practitioners working on keyword detection with large volumes of Arabic text data.

Our study demonstrates the efficacy of SKET for keyword detection in the Arabic language. Future studies may explore its application in various linguistic environments. By leveraging the foundational principles of our model, researchers and practitioners can effectively tailor the system to meet the specific needs of different languages, enhancing its utility in other applications.

4. CONCLUSION

The SKET employs a systematic approach to keyword extraction through distinct phases. It begins with a preprocessing stage to effectively address linguistic challenges. Following this, the frequency-based method identifies potential keywords based on their occurrence within the corpus, using TF as a guiding metric. To further enhance keyword relevance, SKET incorporates a vector space approach that considers the distribution of tokens across the entire corpus, thereby filtering out isolated terms. The machine learning phase introduces a classification algorithm, specifically logistic regression, which captures less common but significant keywords while integrating domain-specific knowledge to improve accuracy. Finally, SKET merges the three keyword lists using a logical “and” operation. The evaluation of SKET reveals its robust performance and efficiency in keyword detection within the CAKE corpus. With an impressive F1-score consistently exceeding 91%, and peaking at 98.3% in the sports domain, SKET demonstrates significant effectiveness in accurately identifying keywords across various contexts. The computational efficiency of the tool is further underscored by its model construction time of less than 1.77 hours and rapid token prediction at approximately 7 milliseconds. In comparative analyses, SKET outperformed several transformer-based models, such as Llama-2 and Mistral, particularly in terms of precision and F1-score.

The limitations faced by transformer models, particularly regarding context size constraints with larger datasets, further emphasize the necessity for specialized tools like SKET. Its design specifically caters to the challenges of processing extensive Arabic text corpora, making it a valuable asset for researchers and practitioners in the field. Moreover, the modular nature of SKET, encompassing preprocessing, frequency analysis, vector space modeling, and machine learning techniques, lays a strong foundation for potential adaptations beyond Arabic. This adaptability opens avenues for extending the tool's capabilities to other languages, thereby enhancing its applicability in diverse linguistic environments.

REFERENCES

[1] Ö. Aydın, F. B. Sakallı, and A. Ş. Mesut, "A review of approaches for keyphrase extraction," *Journal Fundamental Sciences and Applications*, vol. 26, no. 1, pp. 1-6, 2020.

[2] S. Lazemi, H. Ebrahimpour-Komleh, and N. Noroozi, "PAKE: a supervised approach for Persian automatic keyword extraction using statistical features," *SN Applied Sciences*, vol. 1, pp. 1-4, 2019, doi: 10.1007/s42452-019-1627-5.

[3] M. M. A. Hadidi, M. Alzghool, and H. Muaidi, "Keyword extraction from arabic text using the page rank algorithm," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 3495–3504, 2019, doi: 10.35940/ijitee.L2614.1081219.

[4] P. Sharma and M. Chen, "Text summarization and keyword extraction," in *2023 14th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, Koriyama, Japan, 2023, pp. 369-372, doi: 10.1109/IIAI-AAI59060.2023.00078.

[5] F. Horasan, "Keyword extraction for search engine optimization using latent semantic analysis," *Politeknik Dergisi*, vol. 24, no. 2, pp. 473-479, 2021, doi: 10.2339/politeknik.684377.

[6] Z. Yang, H. Yu, J. Tang, and H. Liu, "Toward keyword extraction in constrained information retrieval in vehicle social network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4285-4294, 2019, doi: 10.1109/TVT.2019.2906799.

[7] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "YAKE! Keyword extraction from single documents using multiple local features," *Information Sciences*, vol. 509, pp. 257-289, 2020, doi: 10.1016/j.ins.2019.09.013.

[8] W. A. Etaiwi, A. A. Awajan, and D. Suleiman, "Keywords extraction from Arabic documents using centrality measures," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2019: IEEE, pp. 237-241, doi: 10.1109/SNAMS.2019.8931808.

[9] Y. W. Yu and H. G. Kim, "Interactive morphological analysis to improve accuracy of keyword extraction based on cohesion scoring," *Journal of the Korea Society of Computer and Information*, vol. 25, no. 12, pp. 145-153, 2020, doi: 10.9708/JKSCI.2020.25.12.145.

[10] X. Liu, J. Tan, J. Fan, K. Tan, J. Hu, and S. Dong, "A Syntax-enhanced model based on category keywords for biomedical relation extraction," *Journal of Biomedical Informatics*, vol. 132, p. 104135, 2022, doi: 10.1016/j.jbi.2022.104135.

[11] B. Armouty and S. Tedmori, "Automated keyword extraction using support vector machine from Arabic news documents," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 2019, doi: 10.1109/JEEIT.2019.8717420.

[12] H. Hashim, E.-S. Atlam, A. R. Alzighaibi, and M. Almaliki, "An implementation method for Arabic keyword tendency using decision tree," *International Journal of Computer Applications in Technology*, vol. 63, no. 1-2, pp. 55-63, 2020, doi: 10.1504/IJCAT.2020.107911.

[13] N. A. Abdullah and N. T. Jaboory, "Arabic keywords extraction using conventional neural network," *Iraqi Journal of Science*, pp. 283-293, 2022, doi: 10.24996/ijs.2022.63.1.28.

[14] D. Suleiman, A. A. Awajan, and W. A. Etaiwi, "Arabic text keywords extraction using word2vec," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, 2019: IEEE, pp. 1-7, doi: 10.1109/ICTCS.2019.8923034.

[15] R. Alharbi and H. Al-Muhtasab, "Arabic keyphrase extraction: enhancing deep learning models with pre-trained contextual embedding and external features," in *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, 2022, doi: 10.18653/v1/2022.wanlp-1.30.

[16] P. C. Chatterjee, M. Bordoloi, and S. K. Biswas, "Keyword extraction using Graph based supervised term weighting," in *2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*, 2019, doi: 10.1109/IESPC.2019.8902431.

[17] S. Duari and V. Bhatnagar, "sCAKE: semantic connectivity aware keyword extraction," *Information Sciences*, vol. 477, pp. 100-117, 2019, doi: 10.1016/j.ins.2018.10.034.

[18] N. Zhou, W. Shi, R. Liang, and N. Zhong, "Textrank keyword extraction algorithm using word vector clustering based on rough data-deduction," *Computational Intelligence and Neuroscience*, vol. 2022, 2022, doi: 10.1155/2022/5649994.

[19] X. Mao, S. Huang, R. Li, and L. Shen, "Automatic keywords extraction based on co-occurrence and semantic relationships between words," *Ieee Access*, vol. 8, pp. 117528-117538, 2020, doi: 10.1109/ACCESS.2020.3004628.

[20] W. X. Zhao *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023, doi: 10.48550/arXiv.2303.18223.

[21] K. Shaalan, S. Siddiqui, M. Alkhatib, and A. A. Monem, "Challenges in Arabic natural language processing," in *Computational Linguistics, Speech and Image Processing for Arabic Language*: World Scientific, 2019, pp. 59-83, doi: 10.1142/9789813229396_0003.

[22] J. Rossi and E. Kanoulas, "Query generation for patent retrieval with keyword extraction based on syntactic features," *Frontiers in Artificial Intelligence and Applications*, p. 210, 2018, doi: 10.3233/978-1-61499-935-5-210.

[23] K. Bouzoubaa *et al.*, "A description and demonstration of SAFAR framework," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2021, pp. 127-134, doi: 10.18653/v1/2021.eacl-demos.16.

[24] D. Namly, K. Bouzoubaa, A. E. Jihad, and S. L. Aouragh, "Improving Arabic lemmatization through a lemmas database and a machine-learning technique," *Recent Advances in NLP: The Case of Arabic Language*, pp. 81-100, 2020, doi: 10.1007/978-3-030-34614-0_5.

[25] E. C. Garrido-Merchan, R. Gozalo-Brizuela, and S. Gonzalez-Carvajal, "Comparing BERT against traditional machine learning models in text classification," *Journal of Computational and Cognitive Engineering*, vol. 2, no. 4, pp. 352-356, 2023, doi: 10.47852/bonviewJCCE3202838.
[26] N. Navér, "The past, present or future?: A comparative NLP study of Naive Bayes, LSTM and BERT for classifying Swedish sentences based on their tense," 2021.
[27] Z. A. Guven, "Comparison of BERT models and machine learning methods for sentiment analysis on Turkish tweets," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 2021: IEEE, pp. 98-101, doi: 10.1109/UBMK52708.2021.9559014.
[28] K. Dhola and M. Saradva, "A comparative evaluation of traditional machine learning and deep learning classification techniques for sentiment analysis," in *2021 11th International Conference On Cloud Computing, Data Science and Engineering (Confluence)*, 2021: IEEE, pp. 932-936, doi: 10.1109/Confluence51648.2021.9377070.
[29] G. Bourahouat, M. Abourezq, and N. Daoudi, "Leveraging moroccan arabic sentiment analysis using arabert and qarib," in *The Proceedings of the International Conference on Smart City Applications*, 2022: Springer, pp. 299-310, doi: 10.1007/978-3-031-26852-6_29.

BIOGRAPHIES OF AUTHORS

Driss Namly is Assistant Professor of Computer Science at Mohammed V University in Rabat, Morocco. Prof NAMLY received his PhD degree in Computer Science from Mohammed V University in 2020. His research interests include artificial intelligence, especially natural language processing. He can be contacted at email: d.namly@um5r.ac.ma.

Karim Bouzoubaa is a Full Professor of computer science in the Department of Computer Science, Mohammadia School of Engineers, Mohammed V University in Rabat. Prof/Dr. Karim Bouzoubaa received his PhD degree in 1998 from Laval University. His research interests include artificial intelligence, data science, natural language processing and computational linguistics. He can be contacted at email: karim.bouzoubaa@emi.ac.ma.

Ridouane Tachicart Prof. Tachicart Ridouane has been an Assistant Professor of Computer Science at Chouaib Doukkali University since November 2021, where he teaches both Bachelor's and Master's programs. He coordinates course materials, syllabus, and curriculum with the department chair, and creates lesson plans that align with established curriculum and university policies. Prof. Ridouane evaluates students through written and oral examination methods. He earned his PhD in Computer Science from Mohammed V University in Rabat in 2021. His research interests include natural language processing, machine translation, morphological analysis, text preprocessing, and language resources design and building. He can be contacted at email: tachicart.r@ucd.ac.ma.