

# Arabic text classification using machine learning and deep learning algorithms

Rawad Awad Alqahtani<sup>1</sup>, Hoda A. Abdelhafez<sup>1,2</sup>

<sup>1</sup>Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

<sup>2</sup>Faculty of Computer and Informatics, Suez Canal University, Ismailia, Egypt

## Article Info

### Article history:

Received Jan 29, 2025

Revised Aug 27, 2025

Accepted Oct 16, 2025

### Keywords:

Arabic text classification

Ensemble learning

Linguistic preprocessing

Machine learning

MARBERT

Stemming methods

## ABSTRACT

The classification of Arabic textual content presents considerable challenges due to the language's rich morphological structure and the wide variation among its dialects. This study aims to enhance classification accuracy by leveraging ensemble learning techniques and a deep bidirectional transformer-based model, specifically the multilingual autoregressive BERT (MARBERT). To address linguistic variability, advanced preprocessing techniques were employed, including Farasa, Tashaphyne, and Assem stemming methods. The Al Khaleej dataset served as the basis for supervised learning, providing a representative sample of Arabic text. Furthermore, term frequency-inverse document frequency (TF-IDF) with bigram and trigram feature extraction was utilized to effectively capture contextual semantics. Experimental results indicate that the proposed approach, particularly with the integration of MARBERT, achieves a peak classification accuracy of 98.59%, outperforming existing models. This research underscores the efficacy of combining ensemble learning with deep transformer-based models for Arabic text classification and highlights the critical role of robust preprocessing techniques in managing linguistic complexity and improving model performance.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Hoda A. Abdelhafez

Department of Information Technology, College of Computer and Information Sciences

Princess Nourah bint Abdulrahman University

P.O. Box 84428, Riyadh 11671, Saudi Arabia

Email: [hodaabdelhafez@gmail.com](mailto:hodaabdelhafez@gmail.com)

## 1. INTRODUCTION

Natural language processing (NLP) is a field within data science that focuses on the creation of software with the ability to comprehend, scrutinize, and decipher human speech. The objective of this technology is to improve the interface between humans and computers by enabling communication through writing and speech, hence boosting the computer's ability to understand. Arabic has seen a surge in interest in NLP because of considerable study undertaken in English and other languages. As a result, dedicated Arabic NLP research laboratories have been established to advance various applications, including text classification, spam detection, and sentiment analysis. Nevertheless, the progress of Arabic NLP tools encounters difficulties associated with the incorporation of Arabic characters and the elimination of vowel diacritics [1].

Moreover, Arabic dialects exhibit a wide range of diversity, such as Levantine, Maghrebi, Egyptian, and Arabian Gulf variations. Comprehending these differences is difficult due to morphological variability, orthographic inconsistencies, and linguistic complexity. Arabic texts on social networks often appear in both

modern standard Arabic (MSA) and dialectal forms, which can lead to different meanings for the same word. This complexity exemplifies the extensive linguistic variation characteristic of the Arabic language [2].

Text classification is a crucial and essential task in diverse NLP applications, such as sentiment analysis, subject labeling, question answering, and dialog act categorization. It entails the allocation of predetermined labels to textual content. Given the vast volume of available information, manually sorting and categorizing large text data is a laborious and time-consuming task. In addition, the precision of manual text classification can be readily affected by human variables, such as tiredness and proficiency. Using machine learning techniques to automate the text classification operation is advantageous as it leads to more dependable and less subjective outcomes. Furthermore, this can also improve the efficiency of retrieving information and reduce the issue of information overload by discovering the necessary information. Accurate text classification contributes significantly to the systematic organization of information, the extraction of actionable insights, and informed decision-making across various domains. Whether used for spam detection, topic categorization, or sentiment analysis, effective classification enhances both data comprehension and management [3].

Machine learning is a methodology that enables computers to acquire information and enhance their performance without depending on explicit programming. Machine learning has demonstrated significant advantages in intricate tasks such as NLP, obviating the need for specialist approaches. As a result, machine learning is widely used in areas such as automated NLP. Ensemble learning is an approach used in machine learning to enhance the accuracy of machine learning models [4]. Text categorization is a machine-learning process where a document is categorized into one or more predetermined categories based on its content. Texts can be composed of several genres, such as scientific articles, news reports, movie reviews, and ads. Genre refers to how a text was produced and edited, the linguistic style it employs, and the intended audience it targets [5].

Recent research used the bidirectional encoder representations from transformers (BERT) model, which integrates contextual word information. Transfer learning with embedding is a widely used and sophisticated deep learning technique that improves the effectiveness of various NLP applications. When it comes to categorizing Arabic text, both machine learning-based ensemble learning and bidirectional transformers have unique benefits. Ensemble learning uses the power of different models to increase accuracy and robustness by merging their predictions. Conversely, bidirectional transformers, such as BERT, are exceptionally effective in capturing contextual information and comprehending intricate linguistic patterns. The efficacy of each method in classifying Arabic text depends on several aspects, such as the particular target, characteristics of the dataset, and the computational resources available [6].

Furthermore, Arabic morphology is complicated, and words could have several root forms, which can impact the efficiency of categorization processes. Various stemming methods seek to mitigate this diversity by standardizing word forms to their base forms. An assessment of the influence of various stemming methods on categorization accuracy can offer valuable insights into the most efficient approach for managing Arabic text data.

This study investigates the integration of machine learning-based ensemble methods and deep bidirectional learning within the domain of NLP, with a particular focus on Arabic text. Despite growing interest, the influence of various stemming techniques on classification accuracy remains insufficiently examined. To address this gap, the study conducts a thorough evaluation of Arabic text classification approaches. The main objectives are: i) to compare the performance of traditional machine learning ensemble techniques with MARBERT, a deep bidirectional transformer model, on Arabic text datasets; ii) to evaluate the effect of different stemming methods—namely Assem, Farasa, and Tashaphyne—on classification performance; and iii) to propose a robust preprocessing framework that standardizes Arabic text and accommodate its complex morphological structure. The main contributions of this study are as follows:

- Conduct a comparative analysis of ensemble machine learning methods versus transformer-based models for Arabic text classification.
- Assess and benchmark various stemming techniques with respect to Arabic morphological characteristics and their impact on classification accuracy.
- Develop a comprehensive preprocessing pipeline tailored for Arabic NLP tasks.

The contributions of this study are guided by the following research questions:

- RQ1: what is the comparative effectiveness of ensemble learning and bidirectional transformer models in Arabic text classification?
- RQ2: how do various stemming methods affect classification performance across diverse Arabic datasets?

The remainder of this paper is structured as follows: section 2 reviews related work in the field. Section 3 outlines the methods used in this study, detailing the proposed approach and algorithms. Section 4 presents the implementation process and experimental results. Section 5 discusses the key findings and their implications. Finally, section 6 concludes the study and suggests potential directions for future research.

## 2. RETATED WORK

This section provides a comprehensive survey of previous studies related to the classification of news articles, focusing on the Arabic language. The review is presented in three subsections covering Arabic text classification without ensemble learning and Arabic text classification with ensemble learning techniques. It also includes Arabic text classification using deep bidirectional transformer learning.

### 2.1. Arabic text classification using machine learning without ensemble learning

Several studies have addressed the challenges of Arabic text classification without employing ensemble learning strategies. Muaad *et al.* [7] conducted a comparative study involving seven classification algorithms—multinomial naïve Bayes (MNB), Bernoulli naïve Bayes (BNB), stochastic gradient descent (SGD), logistic regression, support vector classifier (SVC), linear SVC, and convolutional neural networks (CNN)—to classify Arabic text using the Al-Khaleej dataset. The study utilized three feature extraction techniques: term frequency-inverse document frequency (TF-IDF), bag-of-words (BoW), and character-level representation. The authors emphasized that data augmentation for the Arabic language remains a significant challenge and noted that the choice of feature representation techniques plays a critical role in influencing the performance of text classification models. The experimental results showed that linear SVC outperformed the other models in terms of classification accuracy.

Elnagar *et al.* [8] conducted a comparison with several deep learning models based on CNN, recurrent neural networks (RNN), long short-term memory (LSTM), gated recurrent unit (GRU), hierarchical attention network (HAN), and proposed deep learning models for Arabic text classification in two datasets corpus: single-label Arabic news articles dataset (SANAD) and news articles dataset in Arabic (NADiA). The authors employed only one methodology for feature extraction, which was word2Vec embedded models. Highlighted that machine learning approaches employed in single-label classification differ from those used in multi-label classification, as the former require adaptation or issue transformation. Conventional learning methods required adjustment, but deep learning-based models required less modification. Diverse strategies were utilized to address adaptation challenges, one of which involved converting multi-label scenarios into multiple single-label instances. The experimental results showed that all models performed well on the SANAD dataset, with the attention-GRU model achieved the highest accuracy of 96.94% [8].

Muaad *et al.* [9] introduced a novel deep learning-based system called Arabic computer-aided recognition (ArCAR), designed specifically to classify Arabic text using character-level representation. The study addressed critical challenges encountered by traditional machine learning approached in classified Arabic text, attributed to the language's complex morphology and variation. These challenges include stemming, dialects, phonology, orthography, and morphology. The ArCAR system was built using a deep CNN to recognize Arabic text at the character level and underwent validation through five-fold cross-validation for document classification. The ArCAR system demonstrated it was proficient in accurately categorized Arabic text at the character level, achieved an impressive accuracy of 97.76%, an F-measure-score of 92.63%, a precision of 92.75%, and a recall of 92% accorded to the AlKhaleej-balanced dataset.

Muaad *et al.* [10] proposed an enhanced method for Arabic document classification, evaluating the same set of machine learning classifiers mentioned earlier, included MNB, BNB, SGD, logistic regression, SVC, Linear SVC, and CNN on the Al-Khaleej dataset. This study focused on optimizing the feature engineering process, employing BoW, TF-IDF, and character-level features. Experimental results revealed that the CNN model using character-level representations achieved the highest accuracy, reaching 98%, thereby demonstrating the effectiveness of deep learning architectures in handling the complexities of Arabic language.

### 2.2. Arabic text classification using machine learning with ensemble learning

Recent research has demonstrated the effectiveness of ensemble learning techniques in improving the performance of Arabic text classification models, particularly for news articles. Sabri *et al.* [11] applied ensemble learning strategies to the task of automatic Arabic news classification. The authors evaluated the performance of several base classifiers, including decision tree (DT), naïve Bayes (NB), k-nearest neighbors (KNN), and multilayer perceptron (MLP), in conjunction with ensemble techniques such as bagging, boosting, stacking, and voting. The models were tested on three widely recognized Arabic benchmark datasets: WATAN-2004, KHALEEJ-2004, and ANTCorpus. The study illustrated the advantages of ensemble learning in mitigating issues such as bias, overfitting, and noise, while enhancing model diversity, robustness, and scalability. Among the ensemble strategies, stacking and voting yielded the best results, with stacking achieving the highest accuracy of 95.20% on the ANTCorpus dataset. Additionally, the voting approach significantly improved classification accuracy, achieving 93.24% on the KHALEEJ-2004 dataset and 92.15% on the WATAN-2004 dataset.

Mohammed and Kora [12] explored the importance of ensemble learning and deep learning for enhanced text classification. They identified the selection of an optimal deep learning classifier as a key

challenge. The study proposed an ensemble approach to improve classification effectiveness across six datasets in Arabic and English, including the Arabic Twitter Corpus, AJGT, IMDB reviews, SemEval, COVID-19 fake news detection, and ArSarcasm datasets. Several deep learning models were implemented, including LSTM, GRU, CNN, GRU-CNN, LSTM-CNN, and bidirectional long short-term memory (BiLSTM), as well as ensemble techniques such as voting and stacking. The results showed that the ensemble technique significantly improved the classification precision of the initial deep models and outperformed the most advanced ensemble methods. Experimental results demonstrated that the ensemble approach significantly enhanced the classification precision and outperformed individual deep learning models in most cases. The Arabic corpus achieved the highest classification accuracy of 93.2% using the proposed ensemble technique.

Ahmad *et al.* [13] focused on fake news detection and evaluated the effectiveness of ensemble learning strategies such as bagging, boosting, and voting. The authors used the ISOT fake news dataset and compared the performance of multiple classifiers, including linear support vector machines (SVM), CNN, and BiLSTM networks. The ensemble-enhanced model achieved an accuracy of approximately 0.99 on the ISOT dataset and 0.96 on the DS3 dataset, indicating the effectiveness of ensemble methods in text classification for both factual and deceptive content.

Akhadam and Ayyad [14] contributed to the field of Arabic text classification by proposing a comprehensive processing pipeline that incorporates multiple preprocessing techniques aimed at improving classification accuracy. For feature extraction, the study utilized both BoW and TF-IDF methods. A range of machine learning and deep learning models were evaluated, including logistic regression, MNB, BNB, linear SVC, SGD, SVC, and CNN. The experiments were conducted using the Al-Khaleej dataset. Among the models tested, the CNN with word-level representation and stemming achieved the highest classification accuracy, reaching 97%.

### 2.3. Arabic text classification using deep bidirectional transformer learning

The transformer-based models have significantly enhanced the performance of Arabic text classification, particularly through the development of language models tailored to the linguistic and morphological characteristics of Arabic as outlined. Mageed *et al.* [15] introduced two Arabic-specific transformer-based models, ARBERT and MARBERT, developed to address the shortcomings of existing multilingual masked language models (MLMs) such as mBERT, XLM-R, and AraBERT in processing Arabic text. To evaluate the effectiveness of these models across diverse Arabic NLU tasks, the authors proposed a comprehensive benchmark, ArBench, specifically designed for multi-dialectal Arabic. ArBench comprises 41 datasets spanning five major NLU tasks: i) sentiment analysis (AJGT, AraNET, AraSenTi-Tweet, ArSarcasm, ArSAS, ArSenD-LEV, ASTD, ASTD-B, AWATIF, BBN, HARD, LABR, SAMAR, SemEval, SYTS datasets); ii) social meaning prediction (Arap-Tweet, AraDang, AraNET, Arap-Tweet, OSACT-B, FIRE2019, OSACT-A, and AraSarcasm datasets); iii) topic classification (Arabic news text, Khaleej, and OSAC datasets); iv) dialect identification (AOC, ArSarcasm, MADAR-TL, NADI, and QADI); and v) named entity recognition (ANERCorp, ACE-2003BN, ACE-2003BN, ACE-2004BN, and TW-NER dataset). Experimental results showed that ARBERT and MARBERT consistently outperformed the multilingual and earlier Arabic models across these tasks. The authors emphasized the importance of ArBench as a standardized evaluation framework for Arabic NLU and highlighted the significant contributions of ARBERT and MARBERT in advancing language modelling for Arabic.

Bahurmuz *et al.* [16] investigated the application of transformer-based deep learning models for Arabic rumor detection, employing a range of pre-trained models including AraBERT, MARBERT, ArElectra, ArBERT, and mBERT. The study utilized three Arabic-language datasets for model training and evaluation: a rumor vs. non-rumor tweets dataset, a general fake news detection dataset, and a COVID-19 misinformation dataset. Among the models evaluated, MARBERT demonstrated superior performance over AraBERT in terms of classification accuracy. To address the issue of dataset imbalance, the authors employed resampling techniques and conducted hyperparameter tuning to optimize model performance. The hyperparameters used for both AraBERT and MARBERT included an embedding size of 100, batch sizes of 40 for AraBERT and 32 for MARBERT, 8 training epochs, and a learning rate of  $5e-5$ . As a result of these optimizations, both AraBERT and MARBERT achieved a maximum classification accuracy of 97% on the evaluated datasets.

Nassif *et al.* [17] conducted a comprehensive study on Arabic fake news detection using deep contextualized embedding models. The authors developed and evaluated transformer-based classifiers using eight state-of-the-art Arabic-language pre-trained models: AraBert, QaribBert, ARBERT, MARBERT, Arabic-BERT, Arabert, GigaBERTv4, and XLM-Roberta. The study employed two datasets: the first was an original Arabic fake news dataset, collected via web scraping from Arabic Twitter posts; the second was an English-language fake news dataset sourced from Kaggle, which was translated into Arabic to expand the

training data. The pre-trained models were fine-tuned by adjusting parameters such as optimizer, learning rate, number of epochs, and dropout value. The study used ADAMW as the optimizer,  $1e^{-5}$  as the learning rate, the number of epochs from 1 to 100, and 0.23 as the dropout value. The Arabic-BERT and ARBERT models outperformed other models, achieving high accuracy ratings of 98%.

### 3. METHOD

The section explores pre-processing methods and procedures for textual data. It encompasses tasks such as cleansing and standardizing the text within the dataset, employing different stemming techniques, utilizing machine learning and deep learning algorithms, and employing evaluation methods. Figure 1 depicts the sequential phases that constitute the proposed solution.

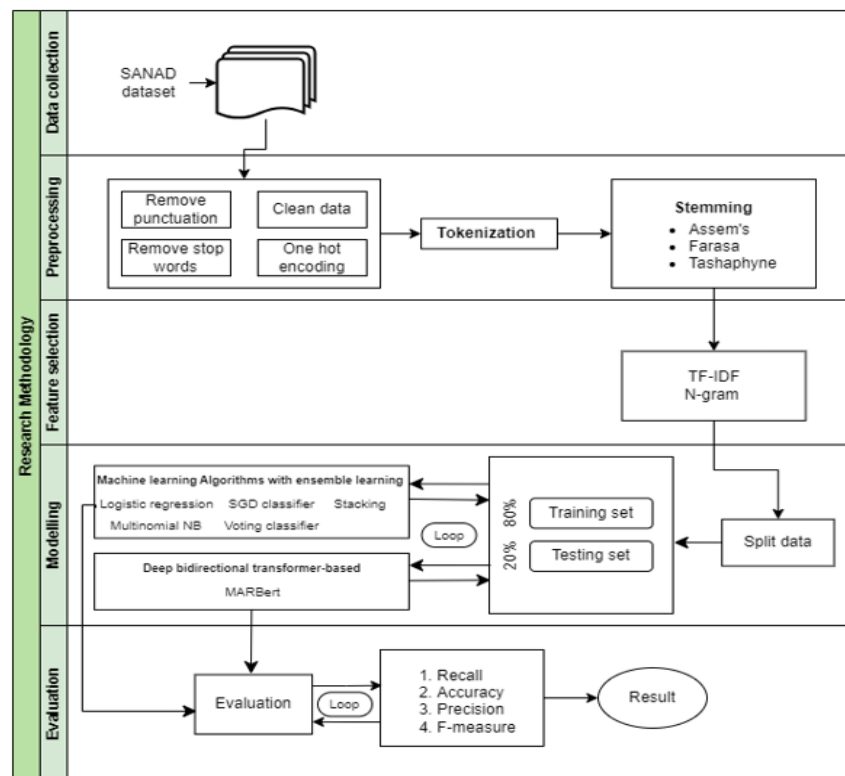


Figure 1. Five-phase methodological framework for the proposed solution

#### 3.1. Data collection

The SANAD data collection is specially designed for Arabic text categorization. The set is huge and classified, including several Arabic articles. These articles are commonly classified into several subjects such as politics, economy, sports, and culture. SANAD is employed by researchers and developers to train and assess machine learning algorithms for the categorization of Arabic text, specifically articles. SANAD has three collection sources: Al Arabia, Al Khaleej, and Akhbarona-Alanba. The Al Khaleej dataset, a constituent of the SANAD corpus, is derived from the Al Khaleej newspaper and serves as a critical resource for a wide range of NLP applications in Arabic. It is particularly utilized in tasks such as text classification, sentiment analysis, and language modeling, where high-quality Arabic textual data is essential. The dataset consists of over 45,500 articles published between 2008 and 2018, with approximately 6,500 documents per category. The corpus is systematically organized into seven categories: culture, technology, politics, medicine, sports, finance, and religion, thereby enabling robust evaluation across diverse content domains [18].

#### 3.2. Data preprocessing

The preprocessing phase includes Arabic text normalization and text cleaning. It also covers text encoding and tokenization. In addition, three distinct stemming methods are implemented.

### 3.2.1. Arabic text normalization, cleaning, and encoding

Text normalization is a fundamental step in NLP, aimed at converting textual data into a standardized and uniform format. In the context of Arabic, this process presents unique challenges due to the language's intricate word structure and morphology. Normalization is particularly vital in MSA, involving the substitution of specific characters with alternatives and the removal of certain elements, primarily frequent conjunctions [8]. Examples of potential actions for Arabic text normalization include:

- Replacing different forms of hamzated alif (أ, إ, ؤ) with alif bare (ا) without hamza.
- Replace the final letter of the word, alif maqsura (ي), with yaa (ي).
- Remove the first waaw (و) character, if there are three or more characters left.

Furthermore, Arabic text particularly from informal sources often includes non-standard symbols such as quotation marks, parentheses, asterisks, and punctuation. These characters are typically considered noise during preprocessing and are removed or replaced with whitespace. Additionally, repeated characters used for (e.g., “مبروووك”) are normalized to their base form (“مبروك”). This process of cleaning the text significantly reduces noise and helps restore words to their natural form, ultimately enhancing the performance and accuracy of text classification models [19]. For example, the phrase: دبي: «الخليج» تعاونت (دبي الخليج تعاونت دلسكو مع مجمع دبي الصناعي) after noise removal is transformed into: (دبي الخليج تعاونت دلسكو مع مجمع دبي الصناعي), demonstrating how the removal of punctuation and extraneous symbols preserves the semantic integrity of the text while reducing noise.

Stop word removal: stop words are frequently occurring words that hold little semantic value. They are typically removed during text preprocessing to minimize noise and improve the effectiveness of downstream NLP tasks. Common Arabic stop words include “في” (in), “إلى” (to), “لـ” (for), and “عن” (about). The process of stop word removal entails filtering out these terms from text, as they usually do not add significant meaning [7].

Categorical data is commonly found in real-world datasets, yet most machine learning algorithms require numerical input. Therefore, it's essential to convert categorical variables into a numerical representation. One of the most used techniques for this purpose is one-hot encoding, which is considered a foundational method due to its simplicity and effectiveness, particularly for nominal variables [20]. In the Al Khaleej dataset, the categorical labels include tech, culture, medical, finance, politics, religion, and sport, are transformed into numerical format through one-hot encoding.

### 3.2.2. Tokenization

Tokenization is a fundamental preprocessing step in NLP, as it transforms raw textual data into structured units tokens that can be represented numerically and processed by machine learning algorithms [21]. Traditional tokenization approaches often rely on whitespace delimiters or punctuation to segment text, but more advanced techniques such as sub-word tokenization and morphology-aware methods have demonstrated superior performance, particularly for morphologically rich languages like Arabic [22]. Arabic presents specific tokenization challenges due to its complex morphology, concatenative word structure, and the frequent absence of diacritic marks, which can lead to lexical ambiguity [23]. Effective tokenization methods for Arabic must account for these linguistic characteristics to enable accurate parsing and feature extraction. By segmenting text into linguistically meaningful units, tokenization enhances the ability of NLP systems to perform downstream tasks such as text classification and sentiment analysis, leading to improved model performance [24].

### 3.2.3. Stemming

In Arabic morphology, words frequently exhibit various prefixes, infixes, and suffixes. A prefix is an affix positioned before the stem of a word, while a suffix is appended to the end. Both prefixes and suffixes exert influence on the meaning of the word they modify, often creating new word categories or altering the existing ones [25]. An infix is an affix inserted within a word, differing from prefixes and suffixes that are added to the beginning or end. In Arabic, as with other Semitic languages, there's a structure comprising root letters and patterns. In this system, infixes, typically vowels or combinations of vowels and specific consonants, are inserted between the base consonants. This process yields different words or expresses distinct grammatical functions. Recent research affirms that Arabic morphology relies heavily on templatic root–pattern structures, where infixes—often comprising vowels and occasionally consonants—are systematically inserted into consonantal roots to derive various grammatical forms. This phenomenon is well-documented in computational morphology systems [26]. Stemming is used to reduce these words to their root form. Some common stemming algorithms for Arabic include:

Assem's stemmer, the Arabic light stemmer, is an algorithm designed to stem Arabic words, utilizing a snowball method to enhance search functionality in the Arabic language. Given Arabic's intricate structure of inflections, stemming poses challenges due to the language's propensity for alterations via

prefixes, infixes, and suffixes, potentially resulting in varied meanings [27]. Farasa is a comprehensive, full-stack Arabic NLP toolkit widely used in tasks such as search, machine translation, it integrates a range of high-performance modules including word segmentation, lemmatization, named entity recognition, part-of-speech tagging, diacritic recovery, and text classification [28]. The Farasa stemming algorithm plays a key role in reducing Arabic words to their root forms, thereby standardizing text and improving the efficiency of downstream applications such as information retrieval and linguistic analysis [29].

Tashaphyne stemmer is another Arabic tool designed for the segmentation and stemming of text, leveraging an affix-based finite state automaton to extract prefixes and suffixes from a defined set of affixes. This tool finds applications in tasks such as named entity identification, sentiment analysis, and text classification [30]. In the sample shown in Table 1, each stem represents a different word root, designed to reduce words to their base form in order to enhance the accuracy of text analysis. However, it is important to recognize that the choice of stemming algorithm can significantly impact the classification results.

Table 1. Examples of word stems generated by three different Arabic stemmers

Arabic stemmer	Arabic text and extracted root	English translation
Sample article	دبي: «الخليج» تعاونت «دلسكو» مع «مجمع دبي الصناعي» لافتتاح أول عيادة طبية متكاملة بهدف تلبية الاحتياجات الطبية لأكثر من 42 ألف عامل يتوزعون على ثلاث قرى عمالية ضمن المجمع. وتعاقدت العيادة مع كبريات شركات التأمين في الدولة لتوفير أعلى مستويات الدعم والرعاية الصحية لقوى العمل في المجمع	Dubai: Al Khaleej: Dulsco collaborated with Dubai Industrial complex to open the first fully integrated medical clinic, aiming to meet the healthcare needs of more than 42,000 workers living across three labor villages within the complex. The clinic has signed agreements with major insurance companies in the country to provide the highest standards of support and healthcare for the workforce in the complex.
Assem stemmer	دب خليج تعاون دلسك مجمع دب صناع لافتتاح اول عياد طبي متكامل هدف تلب احتياح طبيه لاكثر الف عامل يتوزع عل قر عمال ضمن مجمع تعاقد عياده كبريا شركا تام دوله لتوفير اعل مستويا دعم والرعا صحيه لقو عمل مجمع	Dub Gulf collaborated with Dulsco with Manufacturers Complex to open the first integrated medical clinic aiming to meet the medical needs of more than a thousand workers distributed over labor villages within a complex contracted clinic major company in the country to provide the highest level of support and healthcare for the work complex.
Farasa stemmer	دبي خليج تعاون دلسكو مجمع دبي صناعي افتاح اول عياد طبي متكامل هدف لبي احتياح طبي اكثر الف عامل توزع علي قري عامل ضمن مجمع تعاقد عياده اكبر شركة تام دولة توفير اعلي مستوى دعم رعايه صحيه قوي عمل مجمع	Dubai Gulf collaborated with Dulsco Dubai industrial complex opening the first fully integrated medical clinic aimed to meet the medical needs of more than a thousand workers distributed across the labor villages of workers within the complex clinic contacted the largest company in the country providing the highest level of healthcare support strong work complex.
Tashaphyne stemmer	دب خليج تعاون دلسك مجمع دب صناع افتاح ول عياد طب متكامل هدف لب احتياح طبيه اكثر لف عامل توزع علي قر عمال ضمن مجمع تعاقد عياده بر شرك تام دوله وفير عل مستو دعم رعايه صحيه قو عمل مجمع	Dub Gulf cooperation Dulsco Dub complex manufacturers opening an integrated medical clinic goal to meet medical needs of more than a thousand workers distributed across labor villages within a complex contracting clinic with a complete state-of-the-art company, providing a level of support for health care, a strong work complex.

The Assem stemmer works by removing the characters 'ya' ('ي'), tah marbuta, and maftoha ('ة', 'ا') at the end of words. Additionally, it eliminated 'ال' and 'ب' at the beginning of words, along with any associated numerals and pronouns. In contrast, the Tashaphyne stemmer is quite similar to Assem, with minor distinctions such as the exclusion of the initial alef ('ا') from words. However, the Farasa stemmer exhibited a notable difference compared to the other stemmers. Most words reverted to their stems without significant alterations.

Assem and Tashaphyne stemmers employ aggressive approaches that effectively simplify text but may lose important semantic or morphological details. In contrast, Farasa preserves the word integrity, making it ideal for tasks requiring semantic richness, such as sentiment analysis or dialect studies. However, Farasa's less aggressive nature may not be suitable for noisy data, where Assem and Tashaphyne perform better. Each stemmer demonstrates unique strengths tailored to different NLP tasks.

### 3.3. Feature extraction

TF-IDF is a widely used method in NLP and information retrieval that quantifies the importance of a term within a specific document relative to its occurrence across a larger collection of documents. It combines two key metrics: term frequency, which measures how often a word appears in a document, and inverse document frequency, which accounts for how rare the word is across the entire corpus. The TF-IDF

weight helps distinguish significant terms that are characteristic of a document while reducing the impact of frequently occurring but less meaningful words [31].

N-grams play a critical role in text analysis by capturing contextual relationships and syntactic structures between words. Bigrams, which represent sequences of two consecutive words, offer richer contextual information than isolated terms, enhancing the performance of many NLP tasks. Trigrams, involving three-word sequences, provide deeper insights into linguistic patterns and structural dependencies within text. Recent research by Shannaq [32] demonstrates that optimizing n-gram length, including but not limited to bigrams and trigrams, significantly improves classification, accuracy, and generalizability, particularly within Arabic-language corpora.

### 3.4. Modelling in text classification

This section presents machine learning techniques. It also introduces ensemble learning methods. Furthermore, it covers the MARABERT deep bidirectional transformer model.

#### 3.4.1. Machine learning techniques

Machine learning classifiers are fundamental to text classification, providing an automated and effective means of detecting patterns within textual data. Broadly, machine learning can be categorized into three types: supervised learning, unsupervised learning, and reinforcement learning. This research focuses on the supervised learning approach due to its ability to leverage labeled data in building accurate predictive models. Specifically, this study employs logistic regression, MNB, and SGD classifiers, each chosen for its distinct strengths and proven effectiveness in handling text classification tasks.

Logistic regression is a widely adopted statistical technique commonly used for classification and predictive analytics. Its main objective is to predict categorical outcomes instead of continuous variables. By evaluating the likelihood of success versus failure, it transforms odds into probabilities that fall within the range of 0 to 1, as in (1). The simplicity and interpretability of logistic regression make it an excellent baseline model in text classification [33], [34].

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}} \quad (1)$$

The MNB classifier has been extensively used and evaluated in the context of Arabic text classification. MNB is a widely recognized supervised learning algorithm. It is a probabilistic method that uses Bayes' theorem to determine the likelihood of each tag in a sample as in (2). It assumes that all features are conditionally independent, meaning that the presence or absence of one feature is assumed to have no influence on others [35].

$$p(A|B) = \frac{p(A) * p(B|A)}{p(B)} \quad (2)$$

The MNB variant is particularly well-suited for categorical and text data; it is extensively utilized in NLP tasks due to its efficiency and effectiveness. The algorithm operates based on Bayes' theorem, computing the likelihood for each potential tag and selecting the one with the greatest probability as the output. The simplicity of NB, along with its robust efficacy in high-dimensional textual data, prompted its application in this research [36].

A SGD classifier demonstrates strong performance in Arabic text classification tasks, underscoring its effectiveness and dependability as a machine learning method for processing Arabic text. This is especially valuable given the language's intricate morphology and diverse dialects. As an optimization technique, SGD works by iteratively updating model parameters to minimize a cost function. It does so by computing the gradient of the loss function using randomly selected data points [14]. The function calculates the gradient of the loss function with respect to the model's parameters, using either a single training instance or a small batch of samples. The classifier seeks to minimize a predefined loss function —such as hinge loss for linear SVM or log loss for logistic regression. The learning rate parameter controls the magnitude of the increments during parameter updates and has a substantial influence on both the rate at which the optimization process converges, and its stability as in (3). The size of the minibatch used in each iteration also plays a critical role in the algorithm's overall performance. To reduce overfitting and improve generalization to unseen data, SGD classifiers incorporate regularization techniques. They are highly scalable and well-suited for handling high-dimensional data and large-scale datasets [37].

$$W^{(t+1)} = W - \alpha \nabla f_i(W^{(t)}) \quad (3)$$



### 3.4.2. Ensemble learning

Ensemble learning is a technique in machine learning that improves forecast accuracy by aggregating predictions from many models. Attempts to reduce mistakes, improve performance, and boost overall prediction resilience, ensemble learning often results in superior outcomes across various machine learning tasks [38]. A voting classifier, or (majority rules) is regularly used for classification problems. It operates by aggregating the predictions of multiple base models, typically through majority voting or by taking the average of their outputs, to produce a final decision. Each model provides an estimation. An estimate from each model counts as one 'vote'. The most common 'vote' is picked to represent the merged model; this method leverages the strengths of individual models to create a more balanced and accurate overall prediction [13].

Stacking is a powerful ensemble learning approach in machine learning that combines the predictions of many base models to get a final prediction with higher performance. The process comprises training many base models on the same training dataset and then feeding their predictions into a more advanced model, also referred to as a meta-model to create the final prediction. The main concept behind stacking is to incorporate the predictions of many base models to get better predictive performance than using a single model [39].

### 3.4.3. Deep bidirectional transformer learning

The BERT is a deep learning model that employs bidirectional self-attention to evaluate all words in a phrase at the same time, taking into consideration both the left and right contexts. This characteristic makes it a notable language model based on deep learning. BERT can be tested via pre-training applying a MLM and next-sentence prediction (NSP). This involves randomly masking tokens in an input sequence and predicting the masked words based on the surrounding context. MLM assists BERT in comprehending the contextual meaning inside a sentence, while NSP aids BERT in capturing the correlation or association between pairs of phrases. Consequently, by training both techniques simultaneously, BERT acquires a wide-ranging and thorough comprehension of language, encompassing both intricate aspects inside phrases and the coherence between sentences [8]. The motivation for using BERT in this research stems from its ability to handle complex aspects of language, including the relationships between sentences. MARBERT is a pre-trained transformer-based model specifically designed for the Arabic language. The MARBERT model utilizes both dialectal Arabic and MSA as inputs for evaluating semantic sentence similarity [18]. Its tailored design for the complexities of the Arabic language prompted its use in this study to improve text classification efficacy [40]. This study selects models to encompass a variety of methodologies, ranging from traditional machine learning classifiers to advanced deep learning techniques, thereby optimizing performance for Arabic text classification.

## 3.5. Evaluation

Multiple metrics are being used for evaluating machine learning using ensemble learning and deep learning models. The confusion matrix is the most widely employed criterion. Categorization model evaluation is a technique used for assessing the effectiveness of categorization models. This foundational tool serves as the basis for calculating other crucial performance metrics.

Accuracy is a commonly employed measure in multi-class classification, which may be directly derived from the confusion matrix as in (4). Accuracy is the quantification of correctly predicting values in their entirety [41]. It's effective when the classes in a dataset are evenly balanced, as it provides a general measure of the model's correctness. However, for imbalanced datasets, accuracy alone can be misleading, making metrics like precision and recall more valuable for understanding the model's performance [42].

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (4)$$

Precision measures how confidently a model can classify an instance as positive, focusing on the proportion of true positives among all predicted positives as in (5) [41]. It's crucial in situations where the cost of false positives is high. Emphasizing the accuracy of positive predictions helps prevent the model from incorrectly classifying negative instances as positive, which can have significant consequences in many real-world applications [43].

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Recall gauges the model's capacity to locate every positive unit in the dataset and assesses its forecast accuracy for the positive class as in (6) [42]. It's crucial in scenarios where missing positive instances (false negatives) can have serious consequences. Therefore, recall provides valuable insight into the model's ability to capture all relevant positive cases [44].

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

The F1-score evaluates a classification model by combining precision and recall into a single metric, calculated as their harmonic mean as in (7) [43]. It offers a single performance metric that considers both the accuracy of positive predictions and the model's ability to identify all positive cases. This is especially useful in text classification and other domains where both false positives and false negatives have significant consequences [40].

$$F1 - Score = \left( \frac{2}{precision^{-1} + recall^{-1}} \right) = 2 \times \left( \frac{precision \cdot recall}{precision + recall} \right) \quad (7)$$

## 4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

### 4.1. Implementation

As outlined earlier in the method section, the study followed five main phases. The first phase involved selecting and understanding the dataset—the Al Khaleej dataset, which is part of the SANAD corpus. In the second phase, the raw data was pre-processed to prepare it for classification. The third phase focused on feature selection, followed by the application of classification models in the fourth phase. Finally, the fifth phase involved evaluating the performance of these models as shown in Figure 1.

The experiments were conducted using the Google Colab and Kaggle platforms. A custom `extract_features` function, combined with `TfidfVectorizer`, was used to analyze textual data. This method evaluated term importance within a corpus using TF-IDF, with `sublinear_tf` applying a logarithmic scale (TF) with  $1 + \log(TF)$  to better represent term significance. To enhance feature extraction and capture captured subtle linguistic patterns, n-gram representations—specifically bigrams and trigrams—were utilized. A maximum of 10,000 features was selected to strike a balance between computational efficiency and model accuracy. The dataset was split into 80% for training and 20% for testing, following standard practice.

Model training for logistic regression and MNB focused on hyperparameter optimization using randomized search with cross-validation. This approach evaluated multiple configurations across 100 iterations, employing a 3-fold cross-validation strategy to ensure robust and reliable parameter tuning. In the SGD classifier, a squared hinge loss function and L2 regularization were implemented. The model was configured with several parameters, including  $\alpha = 0.0001$  to control regularization strength, an `l1_ratio` of 0.15 to define the ElasticNet mixing ratio, a maximum of 1000 iterations, and a tolerance of 0.001 to specify when the model should halt. Data were shuffled before each iteration, utilizing all available processors for concurrent tasks, setting a fixed seed value of 1 for repeatability, an 'optimal' technique was applied to adjust the learned rate, and the `etal` parameter was set to 0.0. The model also used a `power_t` value of 0.5 for inverse scaling and did not implement early stopping.

Voting and stacking ensemble techniques were implemented to enhance prediction performance by leveraging a combination of multiple models—logistic regression, MNB, and SGD classifier. They converted input data to NumPy arrays for compatibility, leveraged parallel processing to maximize computational efficiency, and provided detailed output during training. While the voting classifier aggregated predictions through a robust voting mechanism, the stacking classifier built a meta-model to learn from base model outputs, both effectively enhancing classification performance.

The MARBERT model (UBC-NLP/MARBERT) was implemented using the Transformers library for tokenization and sequence classification. TF-IDF vectorization was also applied to the text data for additional processing. The Al Khaleej dataset was processed using a Transformers tokenizer, with Lambda functions applying three stemmers—Farasa, Assem, and Tashaphyne—to generate token IDs and encode the text. Tokens and input IDs were stored for use with a pre-trained BERT model. The dataset was then split into features and target variables (80/20 split with random state 42). Sequence padding was performed using the Keras Preprocessing library's `"pad_sequences"` function, setting the maximum sequence length `"MAX_LEN"` of 128, commonly used for BERT models. Attention masks were created to distinguish between actual tokens and padding, enabling BERT to focus on relevant input. The `"train_test_split"` function splits the data into 90% training and 10% validation sets; replication of the split is ensured by specifying a random state of 42. The input data, comprising labeled inputs and attention masks for both training and testing, was transformed into PyTorch tensors to facilitate efficient processing and smooth integration with neural networks. Training utilized a batch size of 16 to enhance computation on both graphics processing units (GPUs) and central processing units (CPUs). The optimizer was then configured using `"optim.AdamW"` with a learning rate of 0.00001. Training was performed over 11 epochs with performance tracked throughout, followed by evaluation on the validation set using the `"flat_accuracy"` function to measure accuracy. During testing, the data underwent similar preprocessing—including tokenization, padding, and

attention mask creation. Predictions were made on the test set using the trained model, and results were decoded back to original labels using a fitted label encoder to assess final model accuracy.

## 4.2. Experiment results and discussion

The study conducted three distinct stemming experiments for both machine learning and ensemble learning, as well as MARBERT—advanced deep learning method. A comprehensive set of evaluation metrics was applied to thoroughly evaluate the effectiveness of each stemming method. The results were meticulously analyzed and discussed individually for each evaluation criterion.

### 4.2.1. Accuracy

Table 2 presents the accuracy results of six classification models—logistic regression, MNB, SGD, voting, stacking, and MARBEENRT—evaluated using three different stemming techniques: Assem, Farasa, and Tashaphyne. It highlighted that the MARBERT classifier with the Assem stemmer achieved the highest accuracy at 98.38% and outperformed both Farasa and Tashaphyne. However, Farasa stemmer coupled with MARBERT also demonstrated notable accuracy at 98.29%, surpassing Tashaphyne's 98.06%. Among the machine learning-based ensemble methods, the stacking technique applied to the Farasa stemmer exhibited a slightly superior accuracy of 97.89% compared to Assem (97.69%) and Tashaphyne (97.81%). Conversely, MNB yields the lowest accuracy across all stemmers, with scores of 94.34%, 94.84%, and 94.51% for Assem, Farasa, and Tashaphyne, respectively.

Figure 2 illustrates minor performance variations based on the stemming methods used, with MARBERT consistently achieving accuracy scores above 98% across the models. This indicates that the models were well-optimized for their respective tasks, and the stemming methods effectively preprocess the text data. Notably, Farasa and Tashaphyne stemming demonstrate relatively higher and closely aligned accuracy scores across all models. However, the Assem stemmer produces lower accuracy scores compared to Farasa and Tashaphyne across various classifiers, included MNB, logistic regression, SGD, voting, and stacking. Interestingly, MARBERT paired with Assem stemming demonstrated significantly higher scores compared to the other stemmers.

Table 2. Accuracy results of six classifiers with different stemming techniques

Stemming/Classifiers	Assem	Farasa	Tashaphyne
Logistic regression	0.9648	0.9678	0.9692
MNB	0.9434	0.9484	0.9451
SGD	0.9565	0.9596	0.9592
Voting	0.9735	0.9767	0.9763
Stacking	0.9769	0.9789	0.9781

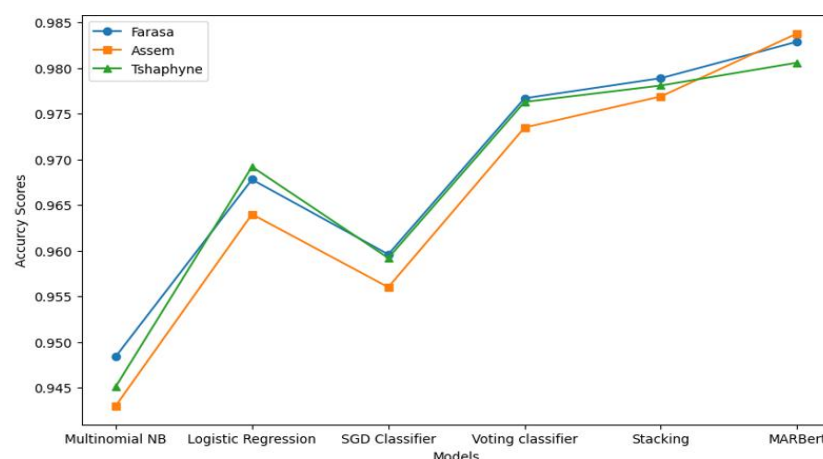


Figure 1. Accuracy results for six classifiers with different stemming techniques

### 4.2.2. Precision

Table 3 presents the precision scores of six classification models evaluated across three different stemming methods. The models include traditional classifiers, ensemble techniques, and the MARBERT model. Among them, MARBERT achieved the highest precision, reaching 98.61% and outperformed logistic regression, MNB, SGD, voting, and stacking. Farasa and Tashaphyne stemming methods also lightly improved results. Moreover, stacking for all three stemming methods demonstrated superior precision scores

in Assem (97.59%), Farasa (97.99%), and Tashaphyne (97.99%), respectively. In contrast, the MNB model consistently recorded the lowest precision scores across all three stemming methods, with values of Assem (96.02%), Farasa (96.41%), and Tashaphyne (96.02%).

Figure 3 illustrates a general upward trend in precision scores across all stemming methods, with the MARBERT model consistently achieved the highest precision scores. This highlights the effectiveness of the MARBERT model in precision-oriented tasks. Tashaphyne and Farasa stemming methods consistently exhibited higher precision scores across all models, while Assem consistently showed lower values. However, Assem achieved the highest precision percentage when used with the MARBERT model.

Table 3. Precision results for six classification models using three stemming methods

Stemming/classifiers	Assem	Farasa	Tashaphyne
Logistic regression	0.9739	0.9739	0.9759
MNB	0.9602	0.9641	0.9602
SGD	0.9661	0.9681	0.9701
Voting	0.9741	0.9780	0.9760
Stacking	0.9759	0.9799	0.9799
MARBERT	0.9861	0.9821	0.9800

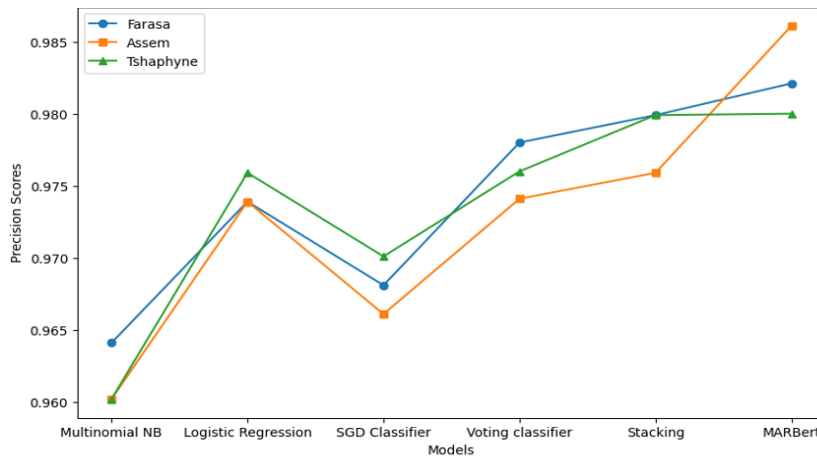


Figure 2. Comparison of precision results for six classification models using three stemming methods

#### 4.2.3. Recall

Table 4 presents the recall scores for logistic regression, MNB, SGD, voting, stacking, and MARBERT classifiers evaluated using three stemming methods. MARBERT classifier achieved the highest recall score of 98.58%, outperforming Farasa (98.39%) and Tashaphyne (97.98%). The Farasa stemmer showed similar recall scores across logistic regression (97.59%), voting (97.99%), and stacking (97.79%), but MNB and SGD classifiers had the lowest recall scores at 95.77% and 96.78%, respectively. The Tashaphyne method showed a significant increase in recall scores across logistic regression, voting, and stacking, while MNB and SGD showed the lowest recall scores at 95.56% and 96.78%, respectively.

Figure 4 illustrates the variation in recall scores across different stemming methods and models. The multinomial model exhibited relatively consistent performance across all stemming methods. However, the logistic regression model demonstrated a significant difference in performance, particularly between Tashaphyne and the other two stemming methods. The MARBERT model consistently achieved high recall scores across all three stemming methods, with Farasa and Tashaphyne showed a very similar performance leveled.

Table 4. Recall scores of six classification models evaluated using three stemming methods

Stemming/Classifiers	Assem	Farasa	Tashaphyne
Logistic regression	0.9700	0.9759	0.9759
MNB	0.9536	0.9577	0.9556
SGD	0.9659	0.9678	0.9678
Voting	0.9718	0.9799	0.9778
Stacking	0.9759	0.9779	0.9779
MARBERT	0.9858	0.9839	0.9798

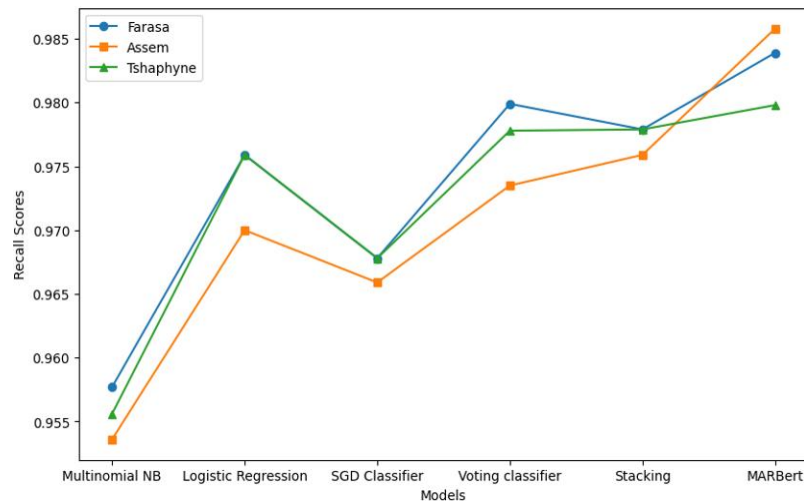


Figure 3. Recall performance of six models evaluated with three stemming methods

#### 4.2.4. F1-measure

Table 5 presents the F1-scores of logistic regression, MNB, SGD, voting, stacking, and MARBERT classifiers evaluated using three stemming techniques. Using the Assem stemmer, the MARBERT classifier attained the best F-measure result at 98.59%, surpassed Farasa (98.39%) and Tashaphyne (97.99%) stemmers. The Farasa and Tashaphyne models showed slightly better results in stacking, with Farasa scored (97.80%) and Tashaphyne scored (97.79%), while the MNB model showed lower F-measured scores across all stemming approaches, indicating lesser effectiveness. Figure 5 demonstrates the fluctuation of F-measured scores across different models, with Farasa and Tashaphyne models show consistent performance across all classifiers. However, the MARBERT model showed differences between the Farasa and Tashaphyne stemming methods, and the Assem Stemmer showed significant variations across different classifiers.

Table 5. F1-scores of six classification models evaluated with three stemming techniques

Stemming/Classifiers	Assem	Farasa	Tashaphyne
Logistic regression	0.9719	0.9739	0.9740
MNB	0.9559	0.9579	0.9579
SGD	0.9660	0.9680	0.9680
Voting	0.9740	0.9759	0.9759
Stacking	0.9759	0.9780	0.9779
MARBERT	0.9859	0.9839	0.9799

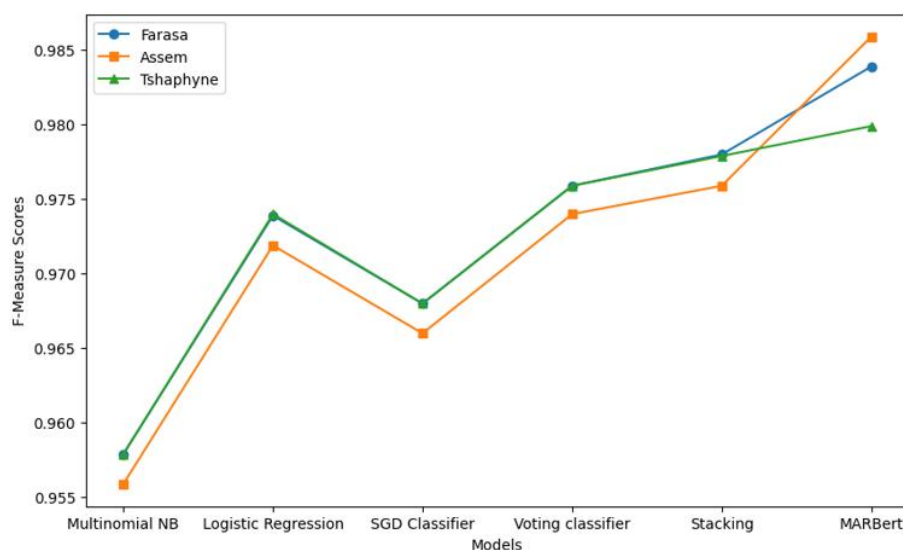


Figure 4. F1-score comparison of six classification models using three stemming techniques

## 5. DISCUSSING THE RESEARCH FINDINGS

The choice of stemming technique exerted a significant effect on the evaluation metrics of models, with Farasa, Assem, and Tashaphyne demonstrated varied effectiveness across models and emphasized the importance of preprocessed steps like stemming in NLP tasks. The MARBERT model consistently scored high in all evaluation metrics for all stemming approaches, demonstrated robustness and efficacy in handled diverse text preprocessed methods, and suggested that more sophisticated models can derive meaningful insight from data.

Farasa and Tashaphyne generally demonstrate competitive or superior stemming performance for most models. However, their application to the more complex MARBERT architecture results in a slight decline in performance. This insight can inform the selection of preprocessing techniques during model deployment, based on the specific model in use. MARBERT consistently delivers superior performance across all evaluation metrics.

Comparison with similar studies, this study employed the Al Khaleej dataset of Arabic news articles to evaluate the effectiveness of a text classification methodology. By integrating TF-IDF with bigram and trigram models, along with various stemming techniques the approach achieved a notable classification accuracy of 98.39% and an F-measure of 98.59%. The results of the proposed solution demonstrated a significant improvement compared to the previous study in [9]. That study used various machine-learned models included MNB, BNB, SGD, logistic regression, SVC, and linear SVC, were utilized. The best-performed algorithm for the Al Khaleej dataset attained an accuracy of 97% used logistic regression, while SGD achieved a similar accuracy of 97%.

Another study in [10] reported that a CNN model achieved an accuracy of 97.76%, an F-measure of 92.63%, a precision of 92.75%, and a recall of 92.75%. Notably, the study in [14] achieved a slightly higher accuracy of 98% using a CNN model as well. While both studies demonstrate comparable accuracy levels, these results highlight the effectiveness of CNN in Arabic text classification tasks. However, it was essential to consider that this research study incorporated advanced pre-processed techniques, machine learning-based ensemble learning, and MARBERT. This comprehensive methodology achieved competitive results while leveraged a diverse range of techniques. This comparison underscored the versatility and efficacy of different methods in achieved high accuracy in Arabic text classification and provided valuable insight for future research and development in the field. Table 6 presents a comparison between the proposed model and those used in other studies.

This research explores the integration of machine learning-based ensemble learning with deep bidirectional learning within the field of NLP, focusing specifically on Arabic text classification. Unlike prior studies, it places particular emphasis on the comparative evaluation of preprocessing and stemming methods, which are crucial for standardizing word variants and enhancing NLP accuracy. The study addresses a critical gap by systematically analyzing how preprocessing choices impact model performance and tailoring stemming techniques for Arabic texts.

Broader impact and ethical considerations: the increasing deployment of Arabic text classification systems raises ethical and societal concerns. Including potential bias in data and model outcomes due to the language's diverse dialects and regional variations. To combat this, it's crucial to use diverse and inclusive datasets that accurately represent the spectrum of Arabic dialects.

Table 6. Comparative performance evaluation of the proposed text classification solution and prior studies

Comparative study	Dataset	Model	Accuracy	Precision	Recall	F-measure
[9]	Khaleej dataset and various datasets	MNB, BNB, SGD, logistic regression, SVC, linear SVC	Logistic regression and SGD 97%			
[10]	Al Khaleej	Deep convolutional neural network (DCNN)	97.76%	92.75%	92.75%	92.63%
[14]	Al Khaleej	MNB, BNB, SGD, logistic regression, SVC, linear SVC, and CNN	CNN 98%			
Proposed solution	Al Khaleej	Logistic regression, MNB, SGD, voting, stacking, MARBERT	MARBERT 98.39%	MARBERT 98.61%	MARBERT 98.58%	MARBERT 98.59%

## 6. CONCLUSION

Classification stands as a fundamental and indispensable facet within the domain of machine learning. With the ongoing surge in textual and document datasets, there arises a crucial necessity to underscore the development and dissemination of supervised machine learning algorithms, particularly those customized for text classification tasks. The study investigates the classification of Arabic text articles sourced from the Al Khaleej dataset, employing preprocessing and feature extraction techniques. It explores a range of text representation methods, including Three stemming algorithms Farasa, Tashaphyne, and Assem, and feature extraction TF-IDF technique to prepare the text for analysis. Through this exploration, the study uncovers variations in performance among machine learning-based ensemble learning and deep learning methods. The study shows that stacking classify, Bigram, and Trigram TF-IDF features consistently yield high performance, with MARBERT being the most effective model in deep learning. The feature representation approach significantly influences text classification performance, allowing classifiers to capture contextual cues, handle linguistic intricacies, reduce computational complexity, support generalization, and enhance interpretability. This contributes to a broader understanding of effective methodologies for Arabic text classification. The research study faced limitations due to the inability to efficiently access powerful computers with high GPU and RAM devices for the study. For future work, the study recommends several directions to improve large language models (LLMs) for Arabic text. These include exploring multilabel text classification, utilizing multiple deep learning algorithms for NLP in Arabic, and conducting domain-specific studies, such as healthcare and education, to provide practical insights. Additionally, efforts could focus on developing specialized lexicons and leveraging extensive Arabic datasets. These approaches could significantly contribute to advancing LLMs for Arabic text and dialects.

## FUNDING INFORMATION

Authors state there is no funding involved.

## AUTHOR CONTRIBUTIONS STATEMENT

This work was solely conducted by the author. This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration. The contributions based on the CRediT taxonomy are as follows:

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rawad Awad Alqahtani	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓			
Hoda A. Abdelhafez	✓	✓				✓	✓			✓	✓	✓	✓	

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

## INFORMED CONSENT

The study did not involve human participation and therefore informed consent was not required.

## ETHICAL APPROVAL

No humans or animals were involved in this study.

## DATA AVAILABILITY

The data that support the findings of this study are openly available in Mendeley Data at <https://data.mendeley.com/datasets/57zpx667y9/1>, reference [18].

## REFERENCES




- [1] S. Pais, J. Cordeiro, and M. L. Jamil, "NLP-based platform as a service: a brief review," *Journal of Big Data*, vol. 9, no. 54, pp. 1–26, Dec. 2022, doi:10.1186/s40537-022-00603-5.
- [2] L. H. Baniata and S. Kang, "Transformer text classification model for Arabic dialects that utilizes inductive transfer," *Mathematics*, vol. 11, no. 24, pp. 1–20, Dec. 2023, doi:10.3390/math11244960.
- [3] Q. Li *et al.*, "A survey on text classification: from traditional to deep learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 2, 2022, doi:10.1145/3495162.
- [4] J. Agarwal, S. Christa, H. A. Pai, M. A. Kumar, and M. S. G. Prasad, "Machine learning application for news text classification," in *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2023, pp. 463–466, doi:10.1109/Confluence56041.2023.10048856.
- [5] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on text classification algorithms: from text to predictions," *Information*, vol. 13, no. 2, Feb. 2022, doi:10.3390/info13020083.
- [6] A. S. Alammary, "BERT models for Arabic text classification: a systematic review," *Applied Sciences*, vol. 12, no. 11, 2022, doi:10.3390/app12115720.
- [7] A. Y. Muaad *et al.*, "Arabic document classification: performance investigation of preprocessing and representation techniques," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–16, Apr. 2022, doi:10.1155/2022/3720358.
- [8] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic text classification using deep learning models," *Information Processing & Management*, vol. 57, no. 1, 2020, doi: 10.1016/j.ipm.2019.102121.
- [9] A. Y. Muaad, M. A. Al-Antari, S. Lee, and H. J. Davanagere, "A novel deep learning ArCAR system for Arabic text recognition with character-level representation," in *Computer Sciences and Mathematics Forum*, vol. 2, no. 1, 2022, doi: 10.3390/IOCA2021-10903.
- [10] A. Y. Muaad *et al.*, "An effective approach for Arabic document classification using machine learning," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 267–271, 2022, doi: 10.1016/j.gltp.2022.03.003.
- [11] T. Sabri, M. Kissi, S. Bahassine, and O. El Beggar, "Analytics of ensemble learning-based methods for Arabic text classification," in *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, IEEE, Nov. 2023, pp. 1–6, doi: 10.1109/SITA60746.2023.10373728.
- [12] A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 10, pp. 8825–8837, Nov. 2022, doi: 10.1016/j.jksuci.2021.11.001.
- [13] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, pp. 1–11, Oct. 2020, doi: 10.1155/2020/8885861.
- [14] I. Akhadam and H. Ayyad, "Enhancing Arabic text classification: a comparative study of machine learning and deep learning approaches," in *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, Marrakech, Morocco, 2024, pp. 1–6, doi: 10.1109/ISIVC61350.2024.10577929.
- [15] M. A.-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: deep bidirectional transformers for Arabic," *59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, doi: 10.18653/v1/2021.acl-long.551.
- [16] N. O. Bahurmuz, G. A. Amoudi, F. A. Baothman, A. T. Jamal, H. S. Alghamdi and A. M. Alhothali, "Arabic rumor detection using contextual deep bidirectional language modeling," *IEEE Access*, vol. 10, pp. 114907–114918, 2022, doi: 10.1109/ACCESS.2022.3217522.
- [17] A. B. Nassif, A. Elnagar, O. Elgendy, and Y. Afadar, "Arabic fake news detection based on deep contextualized embedding models," *Neural Computing and Applications*, vol. 34, pp. 16019–16032, 2022, doi: 10.1007/s00521-022-07206-4.
- [18] O. Einea, A. Elnagar, and R. Al Debsi, "SANAD: single-label Arabic news articles dataset for automatic text categorization," *Data in Brief*, vol. 25, Aug. 2019, doi: 10.1016/j.dib.2019.104076.
- [19] M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, and A. Al-Sumari, "Preprocessing Arabic text on social media: cleaning and normalization algorithms," *Heliyon*, vol. 7, no. 2, 2021, doi: 10.1016/j.heliyon.2021.e06191.
- [20] J. A. Samuels, *One-hot encoding and two-hot encoding: an introduction*, Imperial College London, Jan. 2024, doi: 10.13140/RG.2.2.21459.76327.
- [21] M. T. Alrefaie, N. E. Morsy, and N. Samir, "Exploring tokenization strategies and vocabulary sizes for enhanced Arabic language models," *arXiv:2403.11130*, 2024.
- [22] E. Asgari, Y. El Khair, and M. A. S. Javaheri, "MorphBPE: a morpho-aware tokenizer bridging linguistic complexity for efficient LLM training across morphologies," *arXiv:2502.00894*, 2025.
- [23] M. M. Elmallah *et al.*, "Arabic diacritization using morphologically informed character-level model," in *2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024.
- [24] F. Qarah and T. Alsanoosy, "A comprehensive analysis of various tokenizers for Arabic large language models," *Applied Sciences*, vol. 14, no. 13, 2024, doi: 10.3390/app14135696.
- [25] Z. Alyafeai, M. S. Al-Shaibani, M. Ghaleb, and I. Ahmad, "Evaluating various tokenizers for Arabic text classification," *Neural Processing Letters*, vol. 55, no. 3, pp. 2911–2933, 2023, doi: 10.1007/s11063-022-10990-8.
- [26] M. Alkhairy, A. Jafri, and D. Smith, "Finite state machine pattern-root Arabic morphological generator, analyzer and diacritizer," in *Twelfth Language Resources and Evaluation Conference (LREC)*, Marseille, France, 2020, pp. 3834–3841.
- [27] M. O. Alhawarat, H. Abdeljaber, and A. Hilal, "Effect of stemming on text similarity for Arabic language at sentence level," *PeerJ Computer Science*, vol. 7, May 2021, doi: 10.7717/peerj-cs.530.
- [28] Qatar Computing Research Institute, "Farasa: state of the art full stack Arabic NLP toolkit," *farasa.qcri.org*, 2025. [Online]. Available: <https://farasa.qcri.org/>.
- [29] A. Alrayzah, F. Alsolami, and M. Saleh, "Challenges and opportunities for Arabic question-answering systems: Current techniques and future directions," *PeerJ Computer Science*, vol. 9, 2023, doi: 10.7717/peerj-cs.1633.
- [30] T. Zerrouki, "Tashaphyne: a Python package for Arabic light stemming," *Journal of Open Source Software*, vol. 9, no. 93, Jan. 2024, doi: 10.21105/joss.06063.
- [31] A. A. Mohammed and T. A. Rashid, "Document retrieval using term frequency inverse sentence frequency weighting scheme," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 3, pp. 1478–1485, 2023, doi: 10.11591/ijeecs.v31.i3.pp1478-1485.
- [32] B. Shannaq, "Optimizing n-gram lengths for cross-linguistic text classification: a comparative analysis of English and Arabic morphosyntactic structures," *International Journal of Advanced Applied Sciences*, vol. 12, no. 4, pp. 136–145, 2025.
- [33] A. H. Ababneh, "Investigating the relevance of Arabic text classification datasets based on supervised learning," *Journal of Electronic Science and Technology*, vol. 20, no. 2, pp. 187–208, 2022, doi: 10.1016/j.jnlest.2022.100160.






- [34] C. Starbuck, "Logistic regression," in *The Fundamentals of People Analytics*, Cham, Switzerland: Springer, 2023, pp. 223–238, doi: 10.1007/978-3-031-28674-2\_12.
- [35] M. F. Ibrahim, M. A. Alhakeem, and N. A. Fadhil, "Evaluation of naïve Bayes classification in Arabic short text classification," *Al-Mustansiriyah Journal of Science*, vol. 32, no. 4, pp. 42–50, Nov. 2021, doi: 10.23851/mjs.v32i4.994.
- [36] M. Masadeh *et al.*, "Investigating the impact of preprocessing techniques and representation models on Arabic text classification using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 1, pp. 1115–1123, 2024.
- [37] S. Diab, "Optimizing stochastic gradient descent in text classification based on fine-tuning hyper-parameters approach: a case study on automatic classification of global terrorist attacks," *International Journal of Computer Science and Information Security*, vol. 16, no. 12, pp. 155–160, Feb. 2019.
- [38] B. Arkok and A. M. Zeki, "Classification of Quranic topics using ensemble learning," in *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, IEEE, Jun. 2021, pp. 244–248, doi: 10.1109/ICCCE50029.2021.9467178.
- [39] P. Proscura and A. Zaytsev, "Effective training-time stacking for ensembling of deep neural networks," *2022 5th International Conference on Artificial Intelligence and Pattern Recognition*, 2022, pp. 78–82, doi: 10.1145/3573942.3573954.
- [40] L. B. Cesar, M.-A. M.-Callejo, and C.-I. Cira, "BERT (bidirectional encoder representations from transformers) for missing data imputation in solar irradiance time series," *Engineering Proceedings*, vol. 39, no. 1, doi: 10.3390/engproc2023039026.
- [41] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," *arXiv:2008.05756*, 2020.
- [42] F. Pistorius, D. Grimm, F. Erdosi, and E. Sax, "Evaluation matrix for smart machine-learning algorithm choice," *2020 1st International Conference on Big Data Analytics and Practices (IBDAP)*, Bangkok, Thailand, Sep. 2020, pp. 1–6, doi: 10.1109/IBDAP50342.2020.9245610.
- [43] Z. Karimi, "Confusion matrix," *Research Gate*, 2021. [Online]. Available: <https://www.researchgate.net/publication/355096788>.
- [44] M. F. Amin, "Confusion matrix in three-class classification problems: a step-by-step tutorial," *Journal of Engineering Research*, vol. 7, no. 1, pp. 1–10, Mar. 2023, doi: 10.21608/erjeng.2023.296718.

## BIOGRAPHIES OF AUTHORS



**Rawad Awad Alqahtani**    earned a master's degree in data science from Princess Noura bint Abdulrahman University, specializing in the Department of Information Technology. Her research interests encompass artificial intelligence, machine learning, deep learning, natural language processing, and data engineering. She can be contacted at email: rawadawad93@outlook.sa.



**Hoda A. Abdelhafez**    obtained her Ph.D. in Information Technology from Alexandria University and is currently an Associate Professor in the Department of Information Technology at Princess Nourah bint Abdulrahman University. Her research interests include big data, data science, natural language processing, data warehousing, decision support, data mining, e-learning, machine learning, and deep learning. She has authored three book chapters, two of which were published in the Encyclopedia of Business Analytics and Optimization, and the third in the Encyclopedia of Information Science and Technology. She can be contacted at email: hodaabdelhafez@gmail.com.