frontiers | Research Topics

Emerging techniques in Arabic natural language processing

Edited by
Shadi Abudalfa, Motaz Saad and Samhaa El-Beltagy

Published in
Frontiers in Artificial Intelligence

ISSN 1664-8714
ISBN 978-2-8325-7175-0
DOI 10.3389/978-2-8325-7175-0

Generative AI statement
Any alternative text (Alt text) provided alongside figures in the articles in this ebook has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Emerging techniques in Arabic natural language processing

Topic editors

Shadi Abudalfa — King Fahd University of Petroleum and Minerals, Saudi Arabia

Motaz Saad — Islamic University of Gaza, Palestine

Samhaa El-Beltagy — New Giza University, Egypt

Citation

Abudalfa, S., Saad, M., El-Beltagy, S., eds. (2025). *Emerging techniques in Arabic natural language processing*. Lausanne: Frontiers Media SA.

doi: 10.3389/978-2-8325-7175-0

Table of contents

frontiers | Frontiers in Artificial Intelligence

TYPE Editorial
PUBLISHED 06 November 2025
DOI 10.3389/frai.2025.1715520

OPEN ACCESS

EDITED AND REVIEWED BY
Arkaitz Zubiaga,
Queen Mary University of London,
United Kingdom

*CORRESPONDENCE
Shadi Abudalfa
shadi_abudalfa@hotmail.com

RECEIVED 29 September 2025
ACCEPTED 17 October 2025
PUBLISHED 06 November 2025

CITATION
Abudalfa S, Saad M and El-Beltagy S (2025) Editorial: Emerging techniques in Arabic natural language processing.
Front. Artif. Intell. 8:1715520.
doi: 10.3389/frai.2025.1715520

Editorial: Emerging techniques in Arabic natural language processing

Shadi Abudalfa[1]*, Motaz Saad[2] and Samhaa El-Beltagy[3]

[1]SDAIA-KFUPM JRC for Artificial Intelligence, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, [2]Department of Data Science, Islamic University of Gaza, Gaza, Palestine, [3]School of Information Technology, Newgiza University, Giza, Egypt

KEYWORDS

Arabic NLP, detection, recognition, opinion mining, syntactic analyzers, machine translation, LLMS

Editorial on the Research Topic
Emerging techniques in Arabic natural language processing

Introduction

Arabic Natural Language Processing (NLP) is a rapidly growing field focusing on the unique computational and linguistic challenges of the Arabic language. Recent progress has been driven by deep learning approaches and the increasing use of large language models (LLMs), which have improved applications such as sentiment analysis, text processing, speech recognition, and machine translation (Haboussi et al., 2025; Abdu et al., 2025). Despite these advances, the field still faces critical obstacles, including a shortage of annotated datasets, insufficient tools for dialect handling, and the limited availability of Arabic-oriented LLMs (Mashaabi et al., 2024; Dahou et al., 2025; Abudalfa et al., 2024). This Research Topic presents studies covering various aspects of Arabic NLP, such as syntactic analysis, dialect identification, stance classification, and other tasks that contribute to practical real-world solutions.

Key contributions

The studies featured in this Research Topic highlight advancements in Arabic NLP and introduce innovative approaches within this field. The following subsections provide a concise overview of each paper included.

Syntactic analyzers

Syntactic analysis is a core task in NLP, particularly vital for morphologically rich languages like Arabic. Saadiyeh et al. compared a range of Arabic syntactic analyzers, from rule-based, statistical, and machine learning approaches to hybrid, neural, and transformer-based models, examining their strengths, weaknesses, and trade-offs. The complexity of Arabic morphology and syntax makes accurate parsing challenging, which they address through a detailed evaluation of algorithms and their reliance on high-quality annotated resources.

Machine translation

Algaraady and Mahyoob conducted a study comparing Arabic translations of Google Translate after post-editing by two professional translators and ChatGPT-4o, with three experts evaluating the final output. Quality was assessed through fluency, accuracy, coherence and efficiency, and a paired *t*-test analyzed the differences. Human post-editing generally yielded superior quality, while ChatGPT-4o stood out for speed and produced fluently flowing coherent translations.

In a related line of research, Beidas et al. examine the performance of GPT-3.5, GPT-4, and Bard (Gemini) on the QADI and MADAR datasets, whereas GPT-5 was tested solely on MADAR, which covers data from more than 15 countries. The evaluation relied on several metrics, including cosine similarity, the universal similarity encoder, sentence-BERT, TER, ROUGE, and BLEU. Two prompting strategies were applied: zero-shot and few-shot.

Opinion mining

Alkhathlan et al. presented ArabicStanceX, a large dataset for stance detection with 14,477 tweets covering 17 topics. Using the transformer-based MARBERTv2 model and a Multi-Topic Single Model approach, they achieved an F1 score of 0.74 for "favor" and "against" categories and 0.67 overall. Their results reveal strengths in stance classification but also difficulties with neutral labels and unseen topics. Additional zero-shot and few-shot learning tests show the model's flexibility in adapting to new subjects.

Jaber et al. explored the use of ensemble-based machine learning approaches for Arabic sentiment classification. A range of homogeneous ensemble models is developed and tested on two corpora: the balanced ArTwitter dataset and the highly skewed Syria_Tweets dataset. To address the imbalance problem, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. The experiments combine unigram features with pre-trained word embedding representations.

Arabic poetry

Mutawa and Alrumaih presented a deep learning technique for identifying the meter of Arabic poetry using a large annotated dataset. Text was encoded at the character level to classify full and half verses without removing diacritics, ensuring that essential linguistic features were preserved. Various neural network architectures, including LSTM, GRU, and Bi-LSTM, were explored. This framework demonstrates a robust approach to Arabic meter recognition and highlights the potential of AI in NLP.

Dialect detection

Saleh et al. presented a stacking-based technique to improve dialectal Arabic classification by combining two transformer models, Bert-Base-Arabertv02 and Dialectal-Arabic-XLM-R-Base. The technique involves two layers: the first generates class probabilities from the transformers, which are then used by a meta-learner in the second layer. This technique was benchmarked against individual models such as LSTM, GRU, CNN, and single transformers with various embeddings. Experimental results demonstrated that the combined model outperforms single-model methods by capturing a wider range of linguistic features, improving generalization across Arabic varieties.

Speech recognition

Al-Anzi and Thankaleela presented an Arabic speech recognition framework that begins by extracting Mel-frequency cepstral coefficients (MFCCs) from audio signals. These features are then grouped through K-means clustering, and the resulting clusters are classified using methods such as Decision Trees, Random Forests, K-Nearest Neighbors, and XGBoost. For demonstration purposes, both Euclidean Distance and Dynamic Time Warping (DTW) are employed. Additionally, the research highlights the effectiveness of Mozilla's DeepSpeech framework in handling Arabic speech recognition.

Cyberbullying detection

Allwaibed et al. reviewed 35 scholarly articles addressing the detection of cyberbullying in Arabic-language texts. From a methodological standpoint, traditional machine learning approaches that leverage Arabic-specific linguistic features continue to perform well on smaller datasets. However, more advanced deep learning models and transformer-based frameworks such as AraBERT achieve stronger results, especially when challenges like dialectal variation and orthographic inconsistencies are reduced.

Conclusion

The studies gathered in this Research Topic illustrate the diversity and dynamism of ongoing efforts in Arabic NLP. Collectively, these contributions showcase how deep learning and LLMs are driving progress in Arabic NLP, while also pointing to persistent obstacles such as dialectal differences, scarcity of annotated data, and specialized domain challenges. By introducing innovative approaches, releasing new datasets, and offering comparative assessments, the featured works not only push the field forward but also stress the importance of sustained collaboration, resource creation, and tool development to enhance Arabic NLP and extend its practical impact.

Author contributions

SA: Writing – original draft, Writing – review & editing. MS: Writing – review & editing. SE-B: Writing – review & editing.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that Gen AI was used in the creation of this manuscript. ChatGPT was used to rephrase some sentences to improve readability. The author(s) reviewed and take full responsibility for the content.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdu, F. J., Mughaus, R., Abudalfa, S., Ahmed, M., and Abdelali, A. (2025). An empirical evaluation of Arabic text formality transfer: a comparative study. *Lang. Resour. Eval.* 1–61. doi: 10.1007/s10579-025-09873-w

Abudalfa, S. I., Abdu, F. J., and Alowaifeer, M. M. (2024). Arabic text formality modification: a review and future research directions. *IEEE Access* 12, 185117–185148. doi: 10.1109/ACCESS.2024.3511661

Dahou, A., Dahou, A. H., Cheragui, M. A., Abdedaiem, A., Al-qaness, M. A., Abd Elaziz, M., et al. (2025). A survey on dialect Arabic processing and analysis: recent advances and future trends. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 24:84. doi: 10.1145/3747290

Haboussi, S., Oukas, N., Zerrouki, T., and Djettou, H. (2025). Arabic speech recognition using neural networks: concepts, literature review and challenges. *J. Umm Al-Qura Univ. Appl. Sci.* 1–23. doi: 10.1007/s43994-025-00213-w

Mashaabi, M., Al-Khalifa, S., and Al-Khalifa, H. (2024). A survey of large language models for Arabic language and its dialects. *arXiv* [Preprint]. arXiv:2410.20238. doi: 10.48550/arXiv.2410.20238

frontiers | Frontiers in Human Neuroscience

TYPE Original Research
PUBLISHED 11 February 2025
DOI 10.3389/fnhum.2025.1498297

OPEN ACCESS

EDITED BY
Shadi Abudalfa,
King Fahd University of Petroleum and Minerals, Saudi Arabia

REVIEWED BY
Arwa A. Al Shamsi,
Ministry of Education, United Arab Emirates
Roopesh Bharatwaj KR,
University College Dublin, Ireland

*CORRESPONDENCE
Hager Saleh
✉ hager.saleh.fci@gmail.com

RECEIVED 18 September 2024
ACCEPTED 24 January 2025
PUBLISHED 11 February 2025

CITATION
Saleh H, AlMohimeed A, Hassan R, Ibrahim MM, Alsamhi SH, Hassan MR and Mostafa S (2025) Advancing arabic dialect detection with hybrid stacked transformer models. *Front. Hum. Neurosci.* 19:1498297. doi: 10.3389/fnhum.2025.1498297

Advancing arabic dialect detection with hybrid stacked transformer models

Hager Saleh[1,2,3]*, Abdulaziz AlMohimeed[4], Rasha Hassan[5], Mandour M. Ibrahim[6], Saeed Hamood Alsamhi[7], Moatamad Refaat Hassan[5] and Sherif Mostafa[1]

[1]Faculty of Computers and Artificial Intelligence, Hurghada University, Hurghada, Egypt, [2]Insight SFI Research Centre for Data Analytics, School of Engineering, University of Galway, Galway, Ireland, [3]Atlantic Technological University, Letterkenny, Ireland, [4]Computer Science Department College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia, [5]Department of Computer Science, Faculty of Science, Aswan University, Aswan, Egypt, [6]Information Technology Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia, [7]Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul, Republic of Korea

The rapid expansion of dialectally unique Arabic material on social media and the internet highlights how important it is to categorize dialects accurately to maximize a variety of Natural Language Processing (NLP) applications. The improvement in classification performance highlights the wider variety of linguistic variables that the model can capture, providing a reliable solution for precise Arabic dialect recognition and improving the efficacy of NLP applications. Recent advances in deep learning (DL) models have shown promise in overcoming potential challenges in identifying Arabic dialects. In this paper, we propose a novel stacking model based on two transformer models, i.e., Bert-Base-Arabertv02 and Dialectal-Arabic-XLM-R-Base, to enhance the classification of dialectal Arabic. The proposed model consists of two levels, including base models and meta-learners. In the proposed model, Level 1 generates class probabilities from two transformer models for training and testing sets, which are then used in Level 2 to train and evaluate a meta-learner. The stacking model compares various models, including long-short-term memory (LSTM), gated recurrent units (GRU), convolutional neural network (CNN), and two transformer models using different word embedding. The results show that the stacking model combination of two models archives outperformance over single-model approaches due to capturing a broader range of linguistic features, which leads to better generalization across different forms of Arabic. The proposed model is evaluated based on the performance of IADD and Shami. For Shami, the Stacking-Transformer achieves the highest performance in all rates compared to other models with 89.73 accuracy, 89.596 precision, 89.73 recall, and 89.574 F1-score. For IADD, the Stacking-Transformer achieves the highest performance in all rates compared to other models with 93.062 accuracy, 93.368 precision, 93.062 recall, and 93.184 F1 score. The improvement in classification performance highlights the wider variety of linguistic variables that the model can capture, providing a reliable solution for precise Arabic dialect recognition and improving the efficacy of NLP applications.

KEYWORDS

Arabic dialects, Bert-Base-Arabertv02, Dialectal-Arabic-XLM-R-Base, transformer, Knowledge representation, NLP, deep learning, stacking model

1 Introduction

Dialects within a language are crucial as they represent the various cultural and regional variances within that language (Gregory and Carroll, 2018). As languages change and spread over different geographic areas, dialects naturally arise. Dialects may have their idiomatic phrases, distinct vocabulary, syntax, and pronunciation. Learning dialects has multiple benefits, including better communication, a greater understanding of culture, potential for employment, and increased interaction with media and literature (Zhang and Hansen, 2018). It makes it more straightforward to comprehend the variety within a language and makes it easier to build genuine connections with individuals from various geographical areas (Samih, 2017).

Given the large geographic area in which Arabic is spoken, dialects are essential for the Arabic language. Arabic dialects vary considerably from Modern Standard Arabic (MSA), the standard form for the language (Zaidan and Callison-Burch, 2014). Understanding the regional slang, customs, and traditions specific to each Arabic dialect is possible through understanding dialects. This improves comprehension of culture and makes handling social situations easier. Being fluent in a particular dialect pertinent to your line of work can help you get better employment and more significant support to Arabic-speaking communities (Alosaimi et al., 2024).

Gather a wide range of Arabic language samples across several dialects. The relevant dialect information needs to be labeled on the dataset. The data should be preprocessed by dividing it into training, validation, and test sets, tokenizing the text, and turning it into numerical representations (Haque et al., 2018). Learn a transformer model to identify dialects in Arabic. After the input text has been tokenized, the model should be able to predict the dialect label. Dialect identification requires contextual information captured by the transformer's self-attention mechanism (Lin et al., 2020). The labeled dataset is used to train the model employing optimization techniques (Chapelle et al., 2008).

Deep Learning (DL) and Machine Learning models (ML) have demonstrated promise in processing complicated linguistic data and dialects of Arabic. For example, Elaraby and Abdul-Mageed (2018) applied different ML models: SVM, RF, NB, and LR. Alzu'bi and Duwairi (2021) applied Recurrent Neural Networks (RNN) to support multiple classes of dialects. Alansari (2023) analyzed characteristics of dialects using CNN and RNN. Other authors proposed a hybrid model such as CNN-RNN (Abdelazim et al., 2022). These studies used classical DL models, which cannot capture the long-term dependencies over long sequences.

Therefore, the transformer model has attention features that allow the model to focus on the most relevant parts of the input sequence, capturing long-range dependencies and complex relationships between words (Zhang et al., 2019; Hafiz et al., 2021). For example, Alghamdi et al. (2022) applied two transformer models, MARBERT and ARBERT, using two publicly available Arabic Online Commentary (ADC) (Elaraby and Abdul-Mageed, 2018). In our work, we use recent IADD datasets that were combined from datasets such as (ADC), Dialectal ARabic Tweets dataset (DART) (Alsarsour et al., 2018), the authors in Alghamdi et al. (2022) and Elaraby and Abdul-Mageed (2018) used AOC dataset is published at 2018, and is a subset of IADD, and do not apply stacking model. As a result, the novelty of this paper lies in the combination of transformer models and a meta-learner in a stacking framework designed for Arabic dialect classification. The proposed hybrid model greatly improves the state-of-the-art Arabic dialect detection, outperforms conventional methods, and captures a greater range of linguistic features.

1.1 Motivations and contributions

The motivation behind the paper is the increasing amount of dialectal Arabic information produced by social networks and the need to improve Natural language processing (NLP) functions such as knowledge representation and machine translation. NLP faces challenges due to the fast expansion of dialectal Arabic material on social networks. Substantial language disparities between Arabic dialects and Modern Standard Arabic (MSA) present serious challenges for current NLP models, while this rise provides a wealth of resources for linguistic and computational study. Critical NLP applications like knowledge representation, sentiment analysis, and machine translation are hampered by the models' frequent difficulties with accurate classification and generalization across languages. Classical DL models: CNN, GRU, and LSTM have demonstrated promise in processing complicated linguistic data. Still, these techniques cannot adequately capture the subtle and nuanced differences across Arabic dialects. Furthermore, a significant research vacuum restricts NLP models' wider usability and resilience in Arabic contexts due to the absence of customized solutions to handle these dialectal variances.

To address this gap, we propose a novel stacking model that combines a meta-learner with two transformer architectures: Bert-Base-Arabertv02 and Dialectal-Arabic-XLM-R-Base. By collecting a wider variety of linguistic variables, the proposed models improve dialect categorization, performance, and generalization across different Arabic dialects. Improved classification accuracy, useful applications in machine translation, sentiment analysis, conversational AI, and a strong framework that can be modified to operate with additional low-resource or linguistically challenging languages are some of the added values. The contributions improve the usability and effectiveness of NLP systems for Arabic-speaking regions. The proposed model delivers better performance across different Arabic dialects, increased generalization, and superior dialect classification by integrating various linguistic characteristics. The main contributions of this paper are summarized as follows:

- We introduce a novel stacking model that incorporates two transformer architectures, Bert-Base-Arabertv02 and Arabic-XLM-R-Base, as base models with combined Random Forest (RF) as a meta-learner to enhance classification. The proposed model performs more efficiently than the state-of-the-art models, including LSTM, GRU, CNN, and two transformer models.
- We evaluate the proposed model performance across two datasets to demonstrate the performance in classifying four

and five Arabic dialects. Stacking-Transformer has the highest performance in all rates compared to other models.

- The combination of Transformer in stack modeling with a meta-learner helps to capture more linguistic features, enhance generalization, and accurate dialect detection of Arabic.

1.2 Paper structure

The remainder of the paper is organized into sections. Section 2 presents related works on Arabic dialects. Section 3 outlines the primary steps for classifying Arabic dialects and introduces the proposed model. Section 4 presents the results and discussion, followed by the conclusion in Section 5.

2 Related work

This section presents different researcher have been applied DL, ML, and transformer models to classify Arabic dialects.

Lulu and Elnagar (2018) recognized dialects in Arabic using Four DL models CNN, LSTM, Bidirectional LSTM (Bi-LSTM), and Convolutional LSTM (CLSTM). The authors made use of the Arabic Online Commentary (AOC) dataset, which classifies Arabic into three main dialects: Gulf (including Iraqi), Levantine (LEV), and Egyptian (EGP). LSTM produced the most accurate results. Alsaleh and Larabi-Marie-Sainte (2021) utilized Genetic Algorithms (GA) to optimize the parameters of CNN for Arabic Text Classification. GA was employed to tackle the challenge of randomly initialized weights in CNN. The study utilized two extensive datasets that support text classification. Various pre-processing steps were applied: cleaning, normalization, tokenization, and stemming. The results were improved by 4% using GA with CNN. Alzu'bi and Duwairi (2021) applied RNN to support multiple classes of classification models for dialects. They utilized 110000 sentences from the MADAR corpus, including Maghreb, Levantine, Gulf, and Iraqi dialects. Cotterell and Callison-Burch (2014) proposed Arabic dialects dataset collected from newspaper websites and Twitter, including five Arabic dialects: Levantine, Gulf, Egyptian, Iraqi, and Maghrebi. They utilized unigram, bigram, and trigram models and SVM and NB algorithms. NB with trigram achieved the best accuracy. In addition, Kwaik et al. (2018) proposed the Shami corpus for four Arabic dialects in Palestine, Jordan, Lebanon, and Syria. They explored the effects of pre-processing dialectal Arabic using n-gram and NB models. Various pre-processing steps were applied: cleaning, normalization, tokenization, and stemming. The results showed that NB recorded the highest accuracy. Alansari (2023) captured the semantic and phonological characteristics of dialects using CNN, and RNN. The proposed model comprises six stages: preprocessing, feature engineering, neural networks, optimization techniques, and evaluation methods. Shatnawi et al. (2023) applied different DL models: CNN-BiLSTM, Pooling-BiGRU, and AraBERT with different pre-trained word embedding FastText, AraVec, and AraBERT using a mix of a Katherine dataset that covers the dialects of eight nations and a NADI dataset acquired via Twitter that includes the dialects of twenty-one countries. In addition, they applied various data augmentation to handle unbalanced data. The results showed that models with AraBERT achieved the height performance.

Other researchers have suggested hybrid models, and attention mechanisms and transformer models to classify Arabic dialects. For example, Abdelazim et al. (2022) proposed a hybrid model (CNN-RNN) to classify three dialects: Gulf, Egypt, and Levantine. CNN-RNN, compared with NB, SVM, and CNN, recorded the best accuracy. Alsuwaylimi (2024) proposed two hybrid models that combined BiLSTM with CAMeLBERT and the second model that combined the BiLSTM model with AlBERT. In addition, the conducted dataset includes 121289 collected from comments from various social media platforms and classified into four Arabic dialects (Egyptian, Jordanian, Gulf, and Yemeni). Two models compared with different ML and DL models. Experiment results showed that two hybrid models recorded the best performance. Elaraby and Abdul-Mageed (2018) applied various ML models: SVM, RF, NB, LR, and different DL models: LSTM, GRU, Bi-LSTM, Bi-GRU, and Attention-BiLSTM using various word embedding. Results showed that attention-based BiLSTM work well compared to other models. Alghamdi et al. (2022) applied two transformer models, MARBERT and ARBERT, using two publicly available Arabic-dialect classification datasets such as AOC. They explored results for binary, three, and multi-class dialect classification. The results showed that MARBERT achieved higher performance than ARBERT.

Table 1 compares different models used in research. It outlines the methods, advantages, limitations, and datasets referenced in the studies.

3 Methodology

Figure 1 shows the main steps of classifying Arabic dialects: Data collection, Data pre-processing, Classification models, feature representation methods, classification models, and evaluation models.

3.1 Datasets

Two benchmark Arabic dialect datasets are used for the experiment.

- Shami is a corpus of Levantine Arabic dialects (Kwaik et al., 2018) includes 66,245 rows with four dialect classes: Jordinian, Lebanees, Palestinian, and Syrian. The unbalanced dataset includes 37,758, 10,828, 10,642, and 7,017 rows for Syrian, Lebanese, Palestinian, and Jordanian, respectively.
- IADD is Arabic dialect identification (Zahir, 2022) is used and includes five dialects: Maghrebi (MGH), Levantine (LEV), Egypt (EGY), Iraq (IRQ), Gulf (GLF), and general. It was collected from tweets and Facebook.

TABLE 1 Comparison of existing work.

References	Method	Advantages	Limitations	Dataset
Lulu and Elnagar, 2018	LSTM	Proposing benchmark dataset	Applying the classical DL models Accuracy was lowest	AOC
Alsaleh and Larabi-Marie-Sainte, 2021	GA with CNN	Applying GA to optimize parameters of CNN	Applying the classical DL models Supporting text classification	Text classification
Alzu'bi and Duwairi, 2021	RNN	—	Applying single DL Using one dataset Obtaining the lowest accuracy	MADAR corpus
Cotterell and Callison-Burch, 2014	NB with Bi-gram	Proposing benchmark dataset	Applying ML models Using one dataset Obtaining the lowest accuracy	IADD
Kwaik et al., 2018	NB	Proposing benchmark dataset	Applying single model is NB Obtaining the lowest accuracy	Shami
Alansari, 2023	CNN and RNN	–	The results of the models have not been registered. Applying classical DL models	–
Shatnawi et al., 2023	AraBERT	Applying different wor-embedding Applying AraBERT Model	Obtaining the lowest accuracy	NADI
Abdelazim et al., 2022	RF	Proposing hybrid model	Applying classical DL models	Own
Elaraby and Abdul-Mageed, 2018	Attention BiLSTM	Proposing model based attention	Applying classical ML models. Using one dataset.	ADO
Alsuwaylimi, 2024	CAMeLBERT with BiLSTM	Proposing benchmark dataset Applying transformer models	No applying stacking models	ADO
Alghamdi et al., 2022	MARBERT	Applying transformer models	No applying stacking models	Own
Our work	Stacking-Transformer	Applying transformer to learn complex patterns in datasets.	–	IADD
	Stacking-Transformer	Applying generalization using stacking model based on two transformer models	–	Shami

3.2 Data pre-processing

Pre-processing the input data before starting to implement any model that processes text data is vital due to the various problems inherent, particularly in text data (Chai, 2023). Therefore, it is necessary to effectively rely on pre-processing the input text data to achieve a clear and accurate exploration of Arabic dialects based on stacked transformers. Data processing of the data aims to prepare and improve the quality of the input data to enhance the performance of the model. The four pillars of the pre-processing steps include Tokenization, data cleaning, stop word removal, and stemming (Kathuria et al., 2021). Carrying out these steps carefully will ultimately ensure that we obtain input data useful for accurately detecting the distinction between different Arabic dialects and obtaining a successful model in natural language processing tasks.

- Tokenization represents the first step in preparing textual data specifically, where the text is divided into smaller parts based on language-specific characteristics such as grammar and morphology (Khallaf, 2023). Tokenization comprises two types: word and sub-word Tokenization. In word tokenization, the result of this step is a set of separate words in addition to diacritics and linking marks. While Sub-word Tokenization is employed to handle out-of-vocabulary words and improve model robustness.
- Data Cleaning: The importance of this step lies in obtaining accurate data after removing irrelevant or confusing data that may hinder the performance of the model used. To accomplish this step, a normalization process must first be performed to convert different forms of the same word to its standard form, then deal with punctuation marks and special characters by removing or unifying them, especially those that do not affect the meaning (Berrimi, 2024). Also, deal with incorrect or incomplete data by neutralizing or removing them. After this step, we will ensure obtaining data of acceptable quality and consistency in its context, contributing to the model's success.
- Removing Stop Words enables the model to focus more on the main distinguishing features of dialects in the text. It thus improves the accuracy of the model in identifying and distinguishing them. Stop words represent a group of words that do not carry a critical or influential meaning in the context, and excluding them will positively reduce dimensions such as prepositions and articles (Khurana et al., 2023). These words are collected in a list to be excluded from the input data list.
- Stemming is a vital necessary process that reduces the expected complexity in the input data by converting words to their root form, which will allow better generalization when using

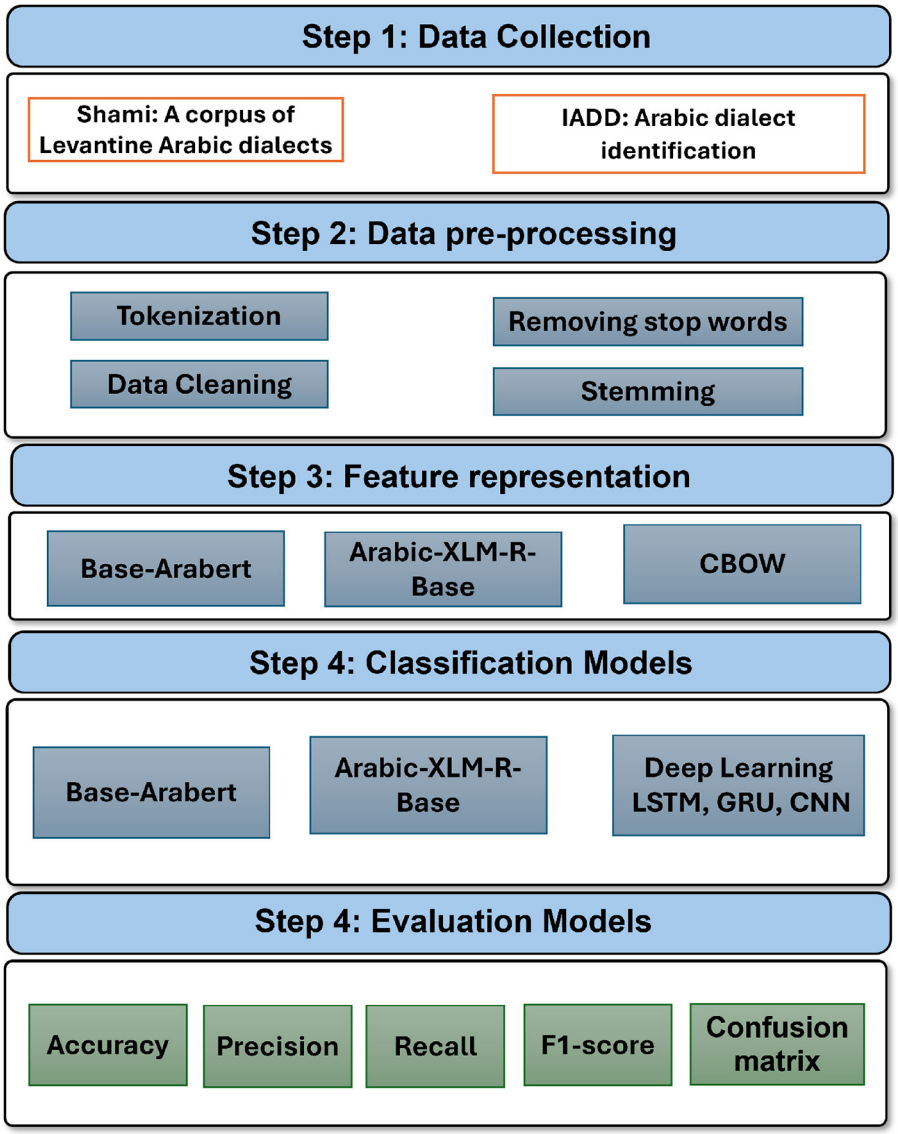

FIGURE 1
Arabic dialects classification framework.

the model to explore dialects (Farghaly and Shaalan, 2009). Many algorithms can be used during this step, some of which are designed specifically for the Arabic language due to its richness in morphology, which helps in grouping different morphological variants of a word. in this paper, stemming applies using Arabic-specific stemming algorithms to handle the morphological richness of Arabic. The algorithms are chosen carefully to prevent mistakes like confusing words with the same root but distinct meanings. In the context of Arabic dialects, this guarantees the results' validity and correctness.

3.3 Dataset splitting

Each dataset is split into a 75% training set and a 25% testing set. The split preserves enough data for objective assessment while guaranteeing reliable model training. Methods for feature representation are customized for the datasets.

3.4 Feature representation methods

While conventional DL models employed CBOW for word embeddings, transformer-based models like Bert-Base-Arabertv02 and Dialectal-Arabic-XLM-R-Base are utilized to generate high-quality contextual embeddings.

- Word2Vec is a widely used technique for learning word embeddings from large volumes of textual data (Karani, 2018). This approach generates embeddings by considering the context in which words appear, enabling the representation of words in a continuous vector space that captures semantic

relationships (Karani, 2018). Word2Vec effectively reduces the dimensionality of the word space while preserving meaningful relationships between words, offering a computationally efficient solution for processing language data (Dwivedi and Shrivastava, 2017). One variant of Word2Vec is the Continuous Bag-of-Words (CBOW) model (Sivakumar et al., 2020). CBOW predicts a target word based on its surrounding context words within a fixed-size window. The model is designed to maximize the probability of correctly predicting the target word, leveraging contextual information to enhance its learning capability (Melamud et al., 2016).

- Bidirectional Encoder Representations from Transformers (BERT) is the open-source transformer-based model that is renowned for its ability to model contextual relationships among words within a sentence through self-attention mechanisms (Vig, 2019). Thanks to this architecture, BERT excels at capturing contextual information and long-range dependencies (Wu et al., 2021). BERT profoundly comprehends linguistic subtleties by being pre-trained on vast volumes of unlabeled text data utilizing two unsupervised tasks. Namely, masked language modeling (MLM) and next sentence prediction (NSP) (Kryeziu and Shehu, 2022). In MLM, words from the input text are randomly masked. BERT is subsequently taught to predict these masked words through analysis of the surrounding context (Devlin et al., 2018). BERT can improve its skills on particular tasks by employing relatively more minor labeled datasets, even when pre-trained on massive quantities of data (Devlin et al., 2018). Bert-base-Arabic refers to the BERT model specially trained on the Arabic language, offering pre-trained representations that encapsulate both syntactic and semantic nuances of Arabic words (Chouikhi et al., 2021). This model accepts Arabic text as input and outputs contextualized word representations, which can be further refined using task-specific training data or directly utilized in downstream NLP tasks (Peters et al., 2019).
- Dialectal Arabic XLM-R Base represents a multilingual transformer model customized to comprehend and interpret several Arabic dialects (Khalifa et al., 2021). An expansion of the BERT architecture called the Cross-lingual Language Model (XLM-R) is intended to function with various languages, including dialects and languages with limited resources (Boudad et al., 2023). This transformer can cope with multiple Arabic dialects alongside other languages since it has been taught on many datasets. Conversational agents can be upgraded to more effectively comprehend and respond to dialectal Arabic more Base using the dialectal Arabic XLM-R Base (Joshi et al., 2024).

  By refining the translations between dialects and standard Arabic, it will be feasible to assess the thoughts and feelings expressed across dialects on social media or in reviews. Built on top of the XLM-R architecture, the Dialectal Arabic XLM-R Base architecture preserves the transformer architecture's scalability and efficacy while being tailored for the complex structure of dialectal Arabic. The model can figure out the word order in a sentence by mapping input tokens to dense vectors and then adding positional information to token embeddings (Qwaider and Abu Kwaik, 2022). Multi-head Self-Attention has been included to allow the model to concentrate on various segments of the input stream concurrently, thereby capturing contextual linkages. A feedforward network processes each attention output before applying it separately to each point. Improves training stability and convergence via normalizing the inputs to each layer (Berrimi, 2024).

3.5 Deep learning models

GRU, LSTM, and CNN are used for DL models.

- GRU is a recurrent architecture with update and reset gates intended to handle sequential data. The update gate controls how much past knowledge remains intact, whereas the Reset gate governs whether earlier data is forgotten (Dey and Salem, 2017). GRU has a hidden state that blends the current input and the prior hidden state, permitting information to flow through time. GRU is appropriate for tasks that need time series data and sequential information, such as language modeling and machine translation (Zargar, 2021). It is beneficial for determining context in textual data.
- LSTM is a more complicated recurrent architecture having forgotten, input, and output gates suitable for learning long-term dependencies (Okut, 2021). The forget gate regulates what information to exclude from the cell state, whereas the input gate determines what latest data to store in the cell state. The output gate determines which information to output based on the cell state (Okut, 2021). The cell state sustains long-term dependencies, allowing gradients to propagate throughout multiple time steps. LSTM can be utilized for text synthesis, machine translation, and speech recognition (Van Houdt et al., 2020). Also, it is competent at predicting potential outcomes using historical and time series data.
- CNN is a type of neural network that comprises convolutional and pooling layers, which help generate features from spatial data. CNN leverages convolution processes to extract features from input data, often images or sequences (Pinaya et al., 2020). It mitigates the spatial dimensions via down-sampling while maintaining the most significant features and then connects the pooled information to the output layer for classification or regression. CNN is frequently implemented for object detection and image segmentation. It also works for sentiment analysis and spam identification since it treats text data as a series (Bhuvaneshwari et al., 2021).

3.6 Proposed model

By integrating the strengths of various models, the stacking approach reflects a wide range of linguistic features, resulting in improved dialect detection. Figure 2 shows the central architecture's two levels. Level 1 provides the base models with

10.3389/fnhum.2025.1498297

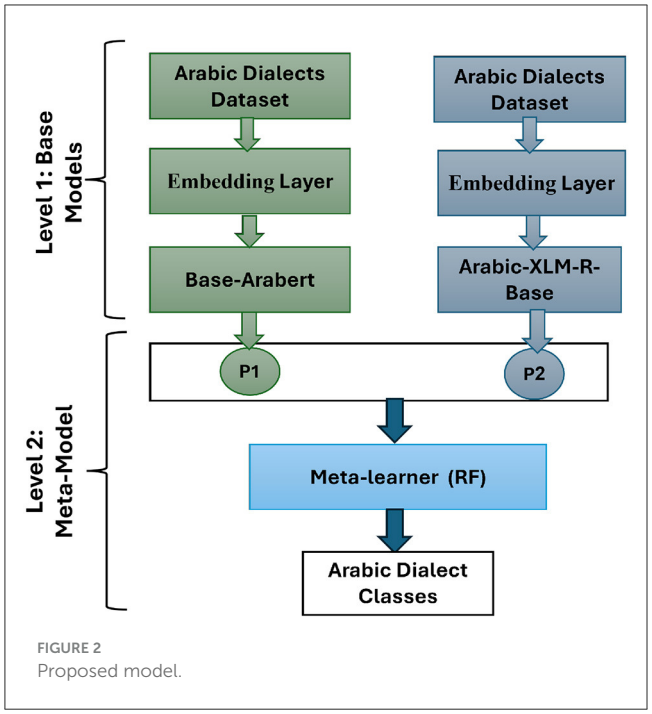

FIGURE 2
Proposed model.

TABLE 2 The number of rows in each dataset.

Datasets	Labels	Training set	Testing set	Total
Shami	Syrian	28,318	9,440	37,758
	Lebanees	8,121	2,707	10,828
	Palestinian	7,981	2,661	10,642
	Jordinian	5,263	1,754	7,017
	Total	49,683	16,562	66,245
IADD	LEV	65,605	21,864	87,469
	MGH	21,037	7,076	28,113
	GLF	5,011	1,671	6,682
	EGY	3,626	1,209	4,835
	general	1,873	625	2,498
	Total	97,152	32,445	129,597

the two transformers that produce class probabilities for training and testing datasets. The second level serves as a meta-learner, which is trained using Level 1's outputs, resulting in enhanced classification performance.

In Level 1, class probabilities are generated by the two transformer models for the training and testing sets and are stored in the stacking training and stacking testing datasets, respectively. In level 2, RF as a meta-learner is trained by stacking training and evaluated by stacking testing to get the final classification decision. RF is an ensemble technique that uses several decision trees during training and combines their outputs for more accurate and stable predictions (Feng et al., 2015).

TABLE 3 Setting of parameters.

Models	Parameters	Specifications
LSTM	Number of nodes	200
	Dropout	0.2
	Activation function	Relu
	Optimizer	Adam
	Loss function	CrossEntropyLoss
GRU	Number of nodes	200
	Dropout	0.2
	Activation function	Relu
	Optimizer	Adam
	Loss function	CrossEntropyLoss
CNN	Filter size	3x3
	Kernel size	4
	Dropout	0.2
	Optimizer	Adam
	Loss function	CrossEntropyLoss
Bert-Base-Arabertv02	Number of transformer layers	12
	Hidden Size	768 dimensions
	Attention Heads	12 per layer
	Optimizer	Adam
	Loss function	CrossEntropyLoss
	Dropout rate	0.1
Dialectal-Arabic-XLM-R-Base	Number of transformer layers	12
	Hidden Size	768 dimensions
	Attention Heads	12
	Optimizer	Adam
	Loss function	CrossEntropyLoss

3.7 Models evaluation

The F1-score, Accuracy, Precision, and Recall metrics are used to assess the models. Where TN indicates the aggregate amount of accurate negative predictions, FP is the total number of false positive estimations, while FN stands for the overall number of false negative predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$F1 - score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (4)$$

TABLE 4 Proposed model performance in Shami dataset.

Approaches	Models	Classes	Precision	Recall	F1-score
DL models	GRU	Jordinian	60.84	55.53	58.06
		Lebanees	77.05	77.39	77.22
		Palestinian	69.22	73.28	71.19
		Syrian	91.28	91.13	91.21
	LSTM	Jordinian	62.25	50.40	55.70
		Lebanees	73.45	75.03	74.23
		Palestinian	72.37	65.54	68.78
		Syrian	87.59	92.48	89.97
	CNN	Jordinian	62.25	50.40	55.70
		Lebanees	73.45	75.03	74.23
		Palestinian	72.37	65.54	68.78
		Syrian	87.59	92.48	89.97
The transformer model	Base-Arabert	Jordinian	80.16	61.52	69.61
		Lebanees	84.64	79.61	82.05
		Palestinian	77.64	82.60	80.04
		Syrian	92.07	95.96	93.98
	Arabic-XLM-R-Base	Jordinian	79.77	60.03	68.51
		Lebanees	84.49	79.09	81.70
		Palestinian	77.34	82.60	79.88
		Syrian	91.82	95.96	93.85
The proposed model	Stacking-Transformer	Jordinian	80.16	61.52	69.61
		Lebanees	84.64	79.61	82.05
		Palestinian	77.64	82.60	80.04
		Syrian	92.07	95.96	93.98

4 Results and discussion

We applied different experiments using various models and two datasets to prove that the Stacking-Transformer model achieved the best performance compared to other models.

4.1 Experimental setup

The experiment was conducted on a laptop with an Intel Core i7 10750H and 16GB memory. The execution environment for the training and validation of the networks was set to a single GPU: Nvidia GeForce GTX 1650 with 4GB VRAM. The models were evaluated by two datasets: Shami with four classes (Jordinian, Lebanees, Palestinian, and Syrian) and IADD with five classes (EGY, GLF, LEV, MGH, and general). Base-Arabert and Dialectal-Arabic-XLM-R-Base are used as feature representations for transformer models, and CBOW is used for DL models. The datasets are split into 75% training set and 25% testing set and the number of rows in each dataset is shown in Table 2. The setting of parameters of models are presented in Table 3.

4.2 Results

Two subsections present the results of Shami and IADD based on precision, recall, F1-score in each class, and confusion matrices. Furthermore, the average accuracy, precision, recall, and F1-score of each dataset is presented.

4.2.1 Proposed model performance in Shami dataset

The results of models based on precision, recall, and F1-score for different classes: Jordinian, Lebanees, Palestinian, and Syrian as shown in Table 4. We can see that GRU, LSTM, and CNN score the lowest in performance compared to transformer models because CNN models focus on local feature extraction but fail to capture complex, long-term relationships. GRU and LSTM handle sequential data, and they have limits to capturing long-term dependencies, especially with large datasets. Transformer-based models leverage self-attention mechanisms to learn both local and global patterns in parallel dynamically, and capture long-term dependencies.

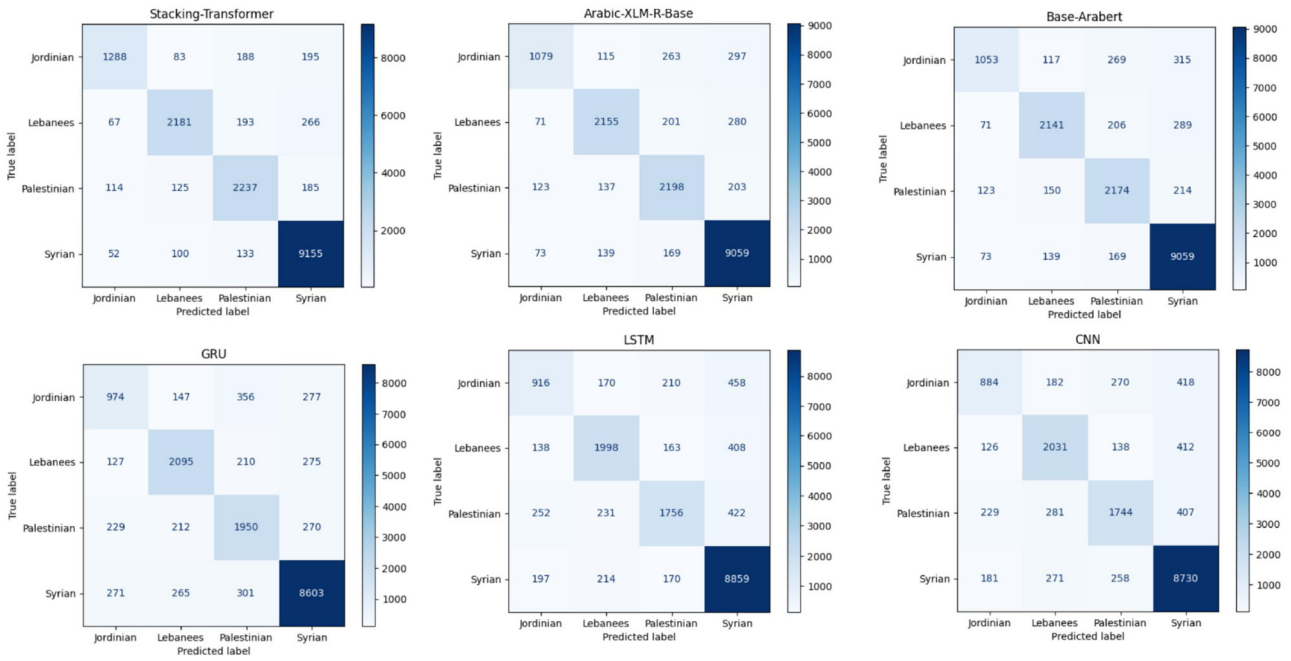

FIGURE 3
Confusion matrices of models for Shami.

The following summarizes the results models with Jordinian record the lowest rates compared to other classes. Models with Syrian class record the highest rate. GRU with Syrian has the highest precision, recall, and F1-score at 91.28, 91.13, and 91.21, respectively. LSTM with Syrian records 91.13 recall higher than GRU. GRU with Lebanees class has the second-highest performance compared to CNN and LSTM with 77.05 precision and 77.22 with F1-score. CNN and LSTM with Lebanees and Palestinian have the same approximate results. Base-Arabert and Arabic-XLM-R-Base with Syrian class record the same recall at 95.96. Both record the same precision, recall, and F1-score at 84.49, 79.09, and 81.70, respectively with Lebanees class. Stacking-Transformer records the highest performance in all classes compared to other models. The best precision, recall, and F1-score are achieved by Stacking-Transformer with Syrian, at 92.07, 95.96, and 93.98, respectively.

Figure 3 comprises six confusion matrices, each of which shows how various models performed in a classification exercise aimed at classifying data into one of four groups: Syrian, Palestinian, Lebanese, or Jordanian. Four groups are created from the models: Syrian, Palestinian, Lebanese, and Jordanian. Darker colors indicate higher counts. The color intensity in each confusion matrix reflects the number of samples sorted into each class. Classifying the Syrian category appears to be generally easier across all models, but the Palestinian and Jordanian categories are more difficult.

4.2.2 Proposed model performance in IADD dataset

Table 5 presents the precision, recall, and F1-score for different classes: EGY, GLF, LEV, MGH, and general for each model. The best precision, recall, and F1-score are achieved by GRU and LSTM with LEV, at 93.19, 93.01, and 93.10, respectively. GRU and LSTM general EGY record the same approximate results. In comparison to CNN and LSTM, GRU with MGH class has the second-highest precision (90.67) and F1-score (89.51). Of all the models based on each class, CNN yields the lowest results. Base-Arabert with GLF records precision, recall, and F1-score at 73.43, 62.18, and 67.34, respectively, compared to DL models. Arabic-XLM-R-Base with LEV and MGH classes records the same precision at 94. The stacking Transformer records the highest performance in all classes compared to other models. The best precision, recall, and F1-score are achieved by Stacking-Transformer with LEV, at 95.90, 95.6, and 95.76, respectively. Also, it has significant performance in the general class compared to other models. Figure 4 comprises six confusion matrices, each of which shows how various models performed in a classification exercise aimed at classifying data into one of five groups: EGY, GLF, LEV, MGH, and general. Darker colors indicate higher counts. The color intensity in each confusion matrix reflects the number of samples sorted into each class.

4.2.3 Discussion

Transformer models have achieved state-of-the-art performance across various tasks compared to traditional DL models for several key reasons the self-attention mechanism in transformers allows them to consider all parts of the input sequence simultaneously. This enables the model to capture long-range dependencies more effectively than traditional recurrent, which are typically limited by sequential processing or fixed-size filters. Figure 5 shows the average accuracy, precision, recall, and F1-score of DL models, transformer models, and the proposed model (Stacking-Transformer) for classifying Syrian, Lebanees, Palestinian, Jordinian. From the table, transformer

TABLE 5 Performance of proposed model in Shami dataset.

Approches	Models		Precision	Recall	F1-score
DL models	GRU	EGY	67.16	56.82	61.56
		GLF	63.49	59.01	61.17
		LEV	93.19	93.01	93.10
		MGH	88.37	90.67	89.51
		general	17.89	22.56	19.96
	LSTM	EGY	66.60	55.42	60.50
		GLF	60.16	58.29	59.21
		LEV	93.16	93.01	93.08
		MGH	87.93	89.36	88.64
		general	17.25	22.08	19.37
	CNN	EGY	66.30	54.51	59.83
		GLF	59.32	58.29	58.80
		LEV	93.11	92.67	92.89
		MGH	87.50	89.36	88.42
		general	16.20	21.28	18.40
The transformer model	Base-Arabert	EGY	71.24	68.24	69.71
		GLF	73.43	62.18	67.34
		LEV	94.07	95.56	94.81
		MGH	94.17	91.72	92.93
		general	23.61	29.12	26.07
	Arabic-XLM-R-Base	EGY	74.71	78.91	76.75
		GLF	75.80	66.37	70.77
		LEV	94.64	95.62	95.13
		MGH	94.44	91.72	93.06
		general	27.59	32.80	29.97
The proposed model	Stacking-Transformer	EGY	80.41	91.65	85.66
		GLF	81.75	80.67	81.20
		LEV	95.90	95.62	95.76
		MGH	94.87	91.72	93.27
		general	43.94	54.56	48.68

models record the best performance compared to deep learning models and improve results by improving results above 5%. The transformer models have the attention that can capture long-range dependencies more effectively than DL models. Arabic-XLM-R-Base has the highest performance compared to Base-Arabert, LSTM, GRU, and CNN with accuracy = 87.495, precision = 87.278, recall = 87.495, and F1-score = 87.209. CNN has the worst all measures with 80.842 of accuracy and 80.363 of F1-score. Stacking-Transformer has the highest performance in all rates with 89.73 of accuracy and 89.574 of f1-score.

Figure 6 shows the average accuracy, precision, recall, and F1-score of DL models, transformer models, and the proposed model (Stacking-Transformer) for classifying EGY, GLF, LEV, MGH, and general. From the table, transformer models record the best performance compared to DL models and improve results by improving results above 2%. Arabic-XLM-R-Base has the highest performance compared to Base-Arabert, LSTM, GRU, and CNN with accuracy = 91.432, precision = 91.595, recall = 91.432, and f1-score = 91.485. CNN has the worst of all measures with 87.382 of accuracy and 87.492 of F1-score. Stacking-Transformer has the highest performance in all rates with 93.062 of accuracy and 93.184 of f1-score, and improve performance by 2 compared to Arabic-XLM-R-Base.

4.3 Comparison of the proposed model with existing work

Table 6 compares our work with the state-of-the-art based on dataset and results. The proposed model, Stacking-Transformer,

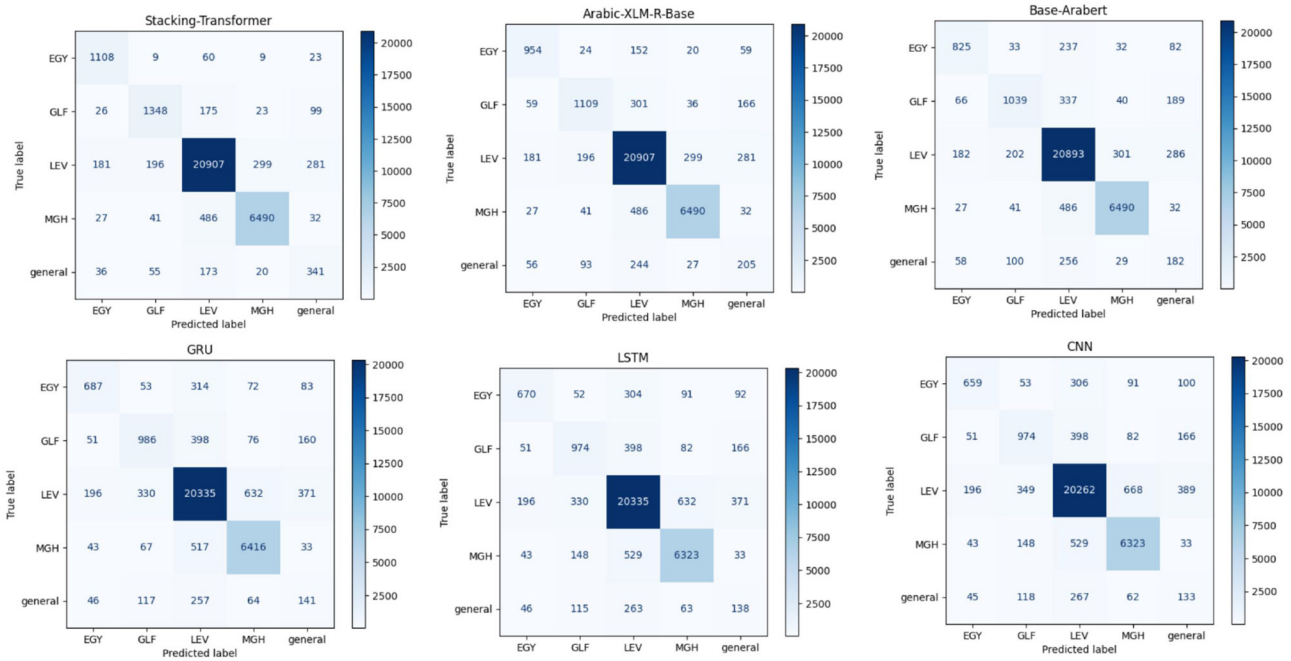

FIGURE 4
Confusion matrices of models for IADD.

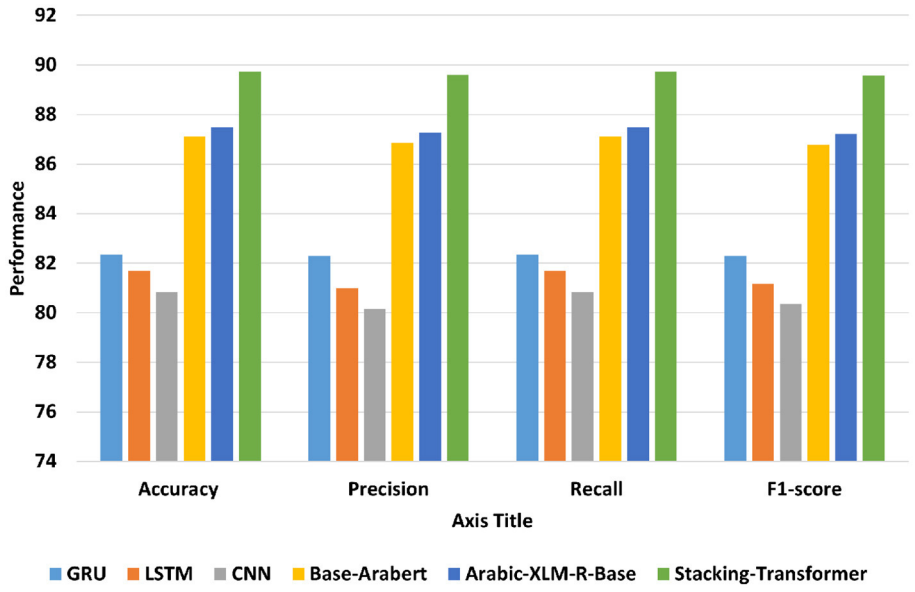

FIGURE 5
Average accuracy, precision, recall, and F1-score of models for Shami.

is based on two transformer models as the baseline and an RF as the meta-learner. It achieves the highest accuracy due to the advantages of the attention mechanism in the transformer, which extracts long dependencies between text, and the generalization capability of stacking models. For IADD, Stacking-Transformer recorded the highest accuracy at 93.062 compared to NB with Bi-gram, which was recorded at 70 in Cotterell and Callison-Burch (2014). For Shami, the Stacking-Transformer recorded the highest accuracy at 89.73 compared to NB in Kwaik et al. (2018). For ADO as a subset of Shami, LSTM was used in Lulu and Elnagar (2018) and recorded 71.4 accuracy. In Elaraby and Abdul-Mageed (2018), Attention BiLSTM recorded 87.81 of accuracy. CAMeLBERT with BiLSTM was recorded at 87.

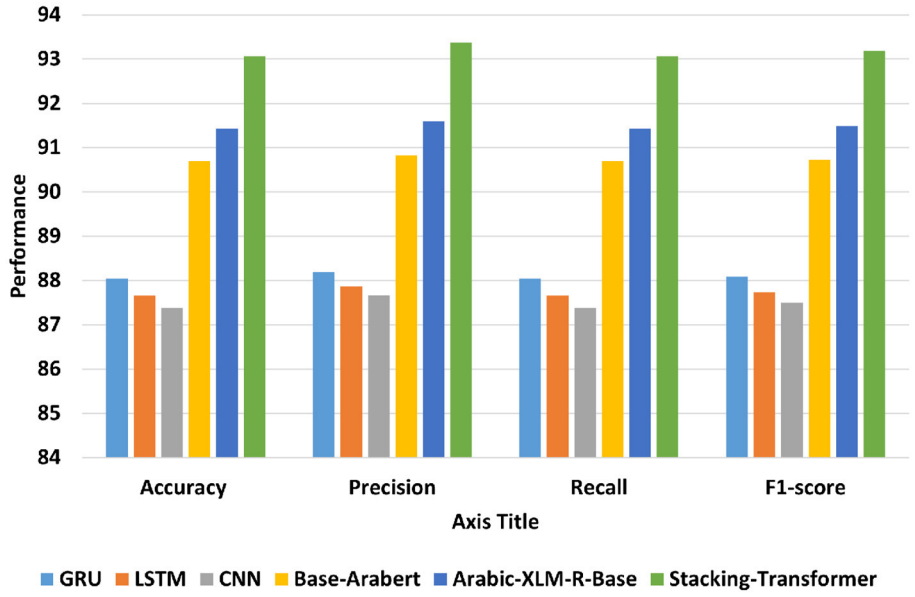

FIGURE 6
Average accuracy, precision, recall, and F1-score of models for IADD.

TABLE 6 Comparison with existing work and the proposed models based on models and performance.

References	Methods	Results	Datasets
Lulu and Elnagar, 2018	LSTM	71.4	AOC
Cotterell and Callison-Burch, 2014	NB with Bi-gram	87.00	IADD
Kwaik et al., 2018	NB	70	Shami
Elaraby and Abdul-Mageed, 2018	Attention BiLSTM	87.81	ADO
Alsuwaylimi, 2024	CAMeLBERT with BiLSTM	87	ADO
Our work	Stacking-Transformer	93.062	IADD
	Stacking-Transformer	89.73	Shami

4.4 Implication and challenges

The proposed investigation has important ramifications for expanding NLP applications and improving Arabic dialect identification. The paper shows improved accuracy, precision, and recall in dialect classification via a hybrid stacking model that incorporates the advantages of transformer designs such as Dialectal-Arabic-XLM-R-Base and Bert-Base-Arabertv02. Given the increasing amount of dialectal material on social media and other platforms, the development fills a significant gap in NLP for managing the linguistic variety of Arabic. The model's cross-dialect generalization establishes a new standard for datasets like Shami and IADD, providing a solid basis for further study and advancement. Additionally, the study has practical applications, such as enhancing conversational AI, sentiment analysis, and machine translation systems to better interpret a variety of complex language inputs.

The paper points out several challenges, including substantial differences in syntax, vocabulary, and semantics between regional dialects and Modern Standard Arabic (MSA) pose a difficult obstacle for models to overcome, especially when generalizing across underrepresented dialects; data imbalance, as seen in the Shami dataset, makes this problem worse and restricts the performance of models on less represented classes, like Jordanian dialects; and the computational demands of training and fine-tuning stacked transformer models demand a significant amount of resources, which may limit accessibility for researchers with limited financial resources. Challenges with scalability and practical implementation also exist, especially for real-time applications that may encounter resource constraints and latency, such as chatbots and virtual assistants. Tokenization, stemming, and stop-word deletion are examples of preprocessing processes that increase complexity since they might not adequately capture the subtle differences present in dialectal Arabic. Even if the model produces state-of-the-art results on certain datasets, there is still a need for more research in generalizing Arabic dialects or languages with equally complex linguistic patterns.

5 Conclusion

In this paper, we introduced a unique stacking model that combines two potent transformer models, Bert-Base-Arabertv02 and Dialectal-Arabic-XLM-R-Base, with a meta-learner to improve the categorization of Arabic dialects. The model formed involved

two levels: base models and meta-learners. Within level one, the two transformer models yield class probabilities for the training and testing sets, which are retained in stacking training and stacking testing, respectively. Level 2 meta-learners with machine learning models are trained and tested using stacking. The stacking model has been contrasted against multiple models, including LSTM, GRU, CNN, and two transfer models with distinct word embedding. Models were assessed on two benchmark datasets to classify four and five dialects of Arabic, featuring various evaluation matrices, including accuracy, precision, recall, F1-score, and confusion matrix. The results proved that the stacking model outperformed single-model techniques. The proposed model addressed a wider spectrum of linguistic traits, allowing for more accurate generalization across different varieties of Arabic. Shami dataset testing reveals that the Stacking-Transformer outperforms all other models in accuracy, precision, recall, and f1-score, with 89.73, 89.596, and 89.574, respectively. For IADD, Stacking-Transformer outperforms other models in all rates, with 93.062 accuracy, 93.368 precision, 93.062 recall, and 93.184 F1-score. In the future, we will concentrate on developing this method to handle other dialects and investigating whether it can be used in other low-resource languages with comparable linguistic complexity.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Ethics statement

Ethical approval was not required for the study involving human data in accordance with the local legislation and institutional requirements. Written informed consent was not required, for either participation in the study or for the publication of potentially/indirectly identifying information, in accordance with the local legislation and institutional requirements. The social media data was accessed and analyzed in accordance with the platform's terms of use and all relevant institutional/national regulations.

Author contributions

HS: Data curation, Methodology, Writing – original draft, Writing – review & editing. AA: Data curation, Investigation, Writing – original draft, Writing – review & editing. RH: Methodology, Visualization, Writing – original draft, Writing – review & editing. MI: Methodology, Validation, Writing – original draft, Writing – review & editing. SA: Methodology, Writing – original draft, Writing – review & editing. MH: Methodology, Supervision, Writing – original draft, Writing – review & editing. SM: Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abdelazim, M., Hussein, W., and Badr, N. (2022). Automatic dialect identification of spoken Arabic speech using deep neural networks. *Int. J. Intell. Comput. Inf. Sci.* 22, 25–34. doi: 10.21608/ijicis.2022.152368.1207

Alansari, I. S. (2023). Artificial intelligence model to detect and classify Arabic dialects. *J. Softw. Eng. Applic.* 16, 287–300. doi: 10.4236/jsea.2023.167015

Alghamdi, A., Alshutayri, A., and Alharbi, B. (2022). "Deep bidirectional transformers for Arabic dialect identification," in *Proceedings of the 6th International Conference on Future Networks Distributed Systems*, 265–272. doi: 10.1145/3584202.3584243

Alosaimi, W., Saleh, H., Hamzah, A. A., El-Rashidy, N., Alharb, A., Elaraby, A., et al. (2024). Arabbert-LSTM: improving Arabic sentiment analysis based on transformer model and long short-term memory. *Front. Artif. Intell.* 7. doi: 10.3389/frai.2024.1408845

Alsaleh, D., and Larabi-Marie-Sainte, S. (2021). Arabic text classification using convolutional neural network and genetic algorithms. *IEEE Access* 9, 91670–91685. doi: 10.1109/ACCESS.2021.3091376

Alsarsour, I., Mohamed, E., Suwaileh, R., and Elsayed, T. (2018). "Dart: a large dataset of dialectal Arabic tweets," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Alsuwaylimi, A. A. (2024). Arabic dialect identification in social media: a hybrid model with transformer models and BILSTM. *Heliyon* 10:e36280. doi: 10.1016/j.heliyon.2024.e36280

Alzu'bi, D., and Duwairi, R. (2021). Detecting regional Arabic dialect based on recurrent neural network," in *2021 12th International Conference on Information and Communication Systems (ICICS)* (IEEE), 90–93. doi: 10.1109/ICICS52457.2021.9464605

Berrimi, M. (2024). *Deep models for understanding and generating textual Arabic data*. PhD thesis.

Bhuvaneshwari, P., Rao, A. N., and Robinson, Y. H. (2021). Spam review detection using self attention based CNN and Bi-directional LSTM. *Multimed. Tools Appl.* 80, 18107–18124. doi: 10.1007/s11042-021-10602-y

Boudad, N., Faizi, R., and Oulad Haj Thami, R. (2023). Multilingual, monolingual and mono-dialectal transfer learning for Moroccan Arabic sentiment classification. *Soc. Netw. Anal. Mining* 14:3. doi: 10.1007/s13278-023-01159-9

Chai, C. P. (2023). Comparison of text preprocessing methods. *Nat. Lang. Eng.* 29, 509–553. doi: 10.1017/S1351324922000213

Chapelle, O., Sindhwani, V., and Keerthi, S. S. (2008). Optimization techniques for semi-supervised support vector machines. *J. Mach. Learn. Res.* 9, 203–233. doi: 10.1145/1390681.1390688

Chouikhi, H., Chniter, H., and Jarray, F. (2021). "Arabic sentiment analysis using Bert model," in *Advances in Computational Collective Intelligence: 13th International Conference, ICCCI 2021, Kallithea, Rhodes, Greece, September 29-October 1, 2021, Proceedings 13* (Springer), 621–632. doi: 10.1007/978-3-030-88113-9_50

Cotterell, R., and Callison-Burch, C. (2014). "A multi-dialect, multi-genre corpus of informal written Arabic," in *LREC*, 241–245.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dey, R., and Salem, F. M. (2017). "Gate-variants of gated recurrent unit (GRU) neural networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)* (IEEE), 1597–1600. doi: 10.1109/MWSCAS.2017.8053243

Dwivedi, V. P., and Shrivastava, M. (2017). "Beyond word2vec: embedding words and phrases in same vector space," in *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, 205–211.

Elaraby, M., and Abdul-Mageed, M. (2018). "Deep models for Arabic dialect identification on benchmarked data," in *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 263–274.

Farghaly, A., and Shaalan, K. (2009). Arabic natural language processing: challenges and solutions. *ACM Trans. Asian Lang. Inf. Proc.* 8, 1–22. doi: 10.1145/1644879.1644881

Feng, Z., Mo, L., and Li, M. (2015). "A random forest-based ensemble method for activity recognition," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (IEEE), 5074–5077. doi: 10.1109/EMBC.2015.7319532

Gregory, M., and Carroll, S. (2018). *Language and Situation: Language Varieties and Their Social Contexts*. London: Routledge. doi: 10.4324/9780429436185

Hafiz, A. M., Parah, S. A., and Bhat, R. U. A. (2021). Attention mechanisms and deep learning for machine vision: a survey of the state of the art. *arXiv preprint arXiv:2106.07550*. doi: 10.21203/rs.3.rs-510910/v1

Haque, T. U., Saber, N. N., and Shah, F. M. (2018). "Sentiment analysis on large scale amazon product reviews," in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)* (IEEE), 1–6. doi: 10.1109/ICIRD.2018.8376299

Joshi, A., Dabre, R., Kanojia, D., Li, Z., Zhan, H., Haffari, G., et al. (2024). Natural language processing for dialects of a language: a survey. *arXiv preprint arXiv:2401.05632*.

Karani, D. (2018). "Introduction to word embedding and word2vec," in *Towards Data Science*, 1.

Kathuria, A., Gupta, A., and Singla, R. (2021). "A review of tools and techniques for preprocessing of textual data," in *Computational Methods and Data Engineering: Proceedings of ICMDE 2020*, 407–422. doi: 10.1007/978-981-15-6876-3_31

Khalifa, M., Abdul-Mageed, M., and Shaalan, K. (2021). Self-training pre-trained language models for zero-and few-shot multi-dialectal Arabic sequence labeling. *arXiv preprint arXiv:2101.04758*.

Khallaf, N. A. A. (2023). *An automatic Modern Standard Arabic text simplification system: a corpus-based approach*. PhD thesis, University of Leeds.

Khurana, D., Koli, A., Khatter, K., and Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimed. Tools Appl.* 82, 3713–3744. doi: 10.1007/s11042-022-13428-4

Kryeziu, L., and Shehu, V. (2022). "A survey of using unsupervised learning techniques in building masked language models for low resource languages," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)* (IEEE), 1–6. doi: 10.1109/MECO55406.2022.9797081

Kwaik, K. A., Saad, M., Chatzikyriakidis, S., and Dobnik, S. (2018). "Shami: a corpus of levantine Arabic dialects," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Lin, W., Madhavi, M., Das, R. K., and Li, H. (2020). "Transformer-based Arabic dialect identification," in *2020 International Conference on Asian Language Processing (IALP)* (IEEE), 192–196. doi: 10.1109/IALP51396.2020.9310504

Lulu, L., and Elnagar, A. (2018). Automatic Arabic dialect classification using deep learning models. *Procedia Comput. Sci.* 142, 262–269. doi: 10.1016/j.procs.2018.10.489

Melamud, O., Goldberger, J., and Dagan, I. (2016). "context2vec: learning generic context embedding with bidirectional LSTM," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 51–61. doi: 10.18653/v1/K16-1006

Okut, H. (2021). "Deep learning for subtyping and prediction of diseases: long-short term memory," in *Deep Learning Applications*. doi: 10.5772/intechopen.96180

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? Adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*.

Pinaya, W. H. L., Vieira, S., Garcia-Dias, R., and Mechelli, A. (2020). "Convolutional neural networks," in *Machine Learning* (Elsevier), 173–191. doi: 10.1016/B978-0-12-815739-8.00010-9

Qwaider, C., and Abu Kwaik, K. (2022). *Resources and applications for dialectal Arabic: the case of Levantine*. Doctoral thesis.

Samih, Y. (2017). *Dialectal Arabic processing Using Deep Learning*. PhD thesis, Dissertation, Düsseldorf, Heinrich-Heine-Universität 2017.

Shatnawi, M. Q., Yasin, M. B., and Huq, A. A. (2023). "Building a framework for identifying Arabic dialects using deep learning techniques," in *ACM Transactions on Asian and Low-Resource Language Information Processing*. doi: 10.1145/3630632

Sivakumar, S., Videla, L. S., Kumar, T. R., Nagaraj, J., Itnal, S., and Haritha, D. (2020). "Review on word2vec word embedding neural net," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)* (IEEE), 282–290. doi: 10.1109/ICOSEC49089.2020.9215319

Van Houdt, G., Mosquera, C., and Nápoles, G. (2020). A review on the long short-term memory model. *Artif. Intell. Rev.* 53, 5929–5955. doi: 10.1007/s10462-020-09838-1

Vig, J. (2019). Visualizing attention in transformer-based language representation models. *arXiv preprint arXiv:1904.02679*.

Wu, Z., Jain, P., Wright, M., Mirhoseini, A., Gonzalez, J. E., and Stoica, I. (2021). "Representing long-range context for graph neural networks with global attention," in *Advances in Neural Information Processing Systems*, 13266–13279.

Zahir, J. (2022). Iadd: an integrated Arabic dialect identification dataset. *Data Brief* 40:107777. doi: 10.1016/j.dib.2021.107777

Zaidan, O. F., and Callison-Burch, C. (2014). Arabic dialect identification. *Comput. Ling.* 40, 171–202. doi: 10.1162/COLI_a_00169

Zargar, S. (2021). *Introduction to Sequence Learning Models: RNN, LSTM, GRU*. Department of Mechanical and Aerospace Engineering, North Carolina State University.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019). "Self-attention generative adversarial networks," in *International Conference on Machine Learning* (PMLR), 7354–7363.

Zhang, Q., and Hansen, J. H. (2018). Language/dialect recognition based on unsupervised deep learning. *IEEE/ACM Trans. Audio, Speech Lang. Proc.* 26, 873–882. doi: 10.1109/TASLP.2018.2797420

frontiers | Frontiers in Artificial Intelligence

TYPE Methods
PUBLISHED 14 February 2025
DOI 10.3389/frai.2025.1523336

OPEN ACCESS

EDITED BY
Houda Bouamor,
Carnegie Mellon University, United States

REVIEWED BY
Miodrag Zivkovic,
Singidunum University, Serbia
Arwa A. Al Shamsi,
Ministry of Education, United Arab Emirates

*CORRESPONDENCE
A. M. Mutawa
✉ dr.mutawa@ku.edu.kw

RECEIVED 05 November 2024
ACCEPTED 28 January 2025
PUBLISHED 14 February 2025

CITATION
Mutawa AM and Alrumaih A (2025) Determining the meter of classical Arabic poetry using deep learning: a performance analysis.
Front. Artif. Intell. 8:1523336.
doi: 10.3389/frai.2025.1523336

Determining the meter of classical Arabic poetry using deep learning: a performance analysis

A. M. Mutawa[1,2]* and Ayshah Alrumaih[1]

[1]Department of Computer Engineering, College of Engineering and Petroleum, Kuwait University, Safat, Kuwait, [2]Department of Computer Sciences, University of Hamburg, Hamburg, Germany

The metrical structure of classical Arabic poetry, deeply rooted in its rich literary heritage, is governed by 16 distinct meters, making its analysis both a linguistic and computational challenge. In this study, a deep learning-based approach was developed to accurately determine the meter of Arabic poetry using TensorFlow and a large dataset. Character-level encoding was employed to convert text into integers, enabling the classification of both full-verse and half-verse data. In particular, the data were evaluated without removing diacritics, preserving critical linguistic features. A train–test–split method with a 70–15–15 division was utilized, with 15% of the total dataset reserved as unseen test data for evaluation across all models. Multiple deep learning architectures, including long short-term memory (LSTM), gated recurrent units (GRU), and bidirectional long short-term memory (Bi-LSTM), were tested. Among these, the bidirectional long short-term memory model achieved the highest accuracy, with 97.53% for full-verse and 95.23% for half-verse data. This study introduces an effective framework for Arabic meter classification, contributing significantly to the application of artificial intelligence in natural language processing and text analytics.

KEYWORDS

Arabic poetry, Arabic meters, Bi-LSTM, deep learning, machine learning, natural language processing

1 Introduction

Arabic prosody (Arud) has been studied for many years in morphology and phonetics. The study of meters in poetry enables us to determine whether the poetry is sound or broken (Jones, 2011). Some of the terminology used most frequently in Arabic prosody are as follows: a single line of the poetry comprises two verses, each half-verse called a "bayt." The first verse is "sadder," and the second is "ajuz." Classical Arabic poetry, defined by units called meters, was analyzed by the famous lexicographer and grammarian Al-Khalil ibn Ahmad al-Farahidi in the eighth century (Alnagdawi et al., 2013). The meter is based on the syllables in a word and consists of two parts: short and long syllables. The 16 meters are Tawil, Basiit, Madid, Wafir, Kamil, Hazaj, Rajaz, Ramal, Munsarih, Khafif, Muqtadab, Mujtath, Mudari', Sarii', Mutaqarib, and Mutadarik. The ode may consist of 120 lines, split into two half-lines characterized by their meters, repeated for the whole verse. Al-Farahidi represented some feet provided in a rhythmic to make it easy to remember the meter (fa'uulun, mafaa'iilun).

Poetry is a way of communication and interaction and an essential aspect of any language and literature. Communities, nations, and societies have expressed themselves through poetry for ages (Lavzheh, 2009). Poetry is hard to understand as it has a specific pattern and underlying meanings in its words and phrases, making it different from prose. It is necessary to understand the structure to understand the poetry completely. Bahar is the meters in Arud science. Arud science helps divide Arabic poems into 16 meters, making them easy to

understand without referring to the context (Alnagdawi et al., 2013). Classical Arabic poetry can be recognized and understood using various methods and tools. Arud is the rule and regulations of poems used in many languages (Abuata and Al-Omari, 2018). Poetry is different from prose, mainly because of its form and structure. Poetry consists of tone, metrical forms, rhythm, imagery, and symbolism. In Arabic poetry, each line ends with a specific tone. The field that studies rhyme and rhythm is called prosody and is complex due to many overlapping rules (Khalaf et al., 2009).

There are two vowels in modern and classical Arabic: long and short. The long vowels are explicitly written, and short vowels are also called diacritic. Various attempts have been carried out to implement Arabic text. A proposal was made to use Arabic diacritics or 'harakat' for text hiding for security purposes (Ahmadoh and Gutub, 2015). The diacritics in Arabic are split into three parts as shown in Table 1. The majorty of studies in this field use a deep learning method to diacritize the Arabic text before loading it into the model (Abandah et al., 2022; Abandah et al., 2020; Kharsa et al., 2024).

Artificial intelligence (AI) has become exponentially more practical and significant over the last few years. The AI-enabled state-of-the-art technologies have expanded substantially and shown effective results in almost every industry, such as security (Wu et al., 2020), surveillance, health (Davenport and Kalakota, 2019), automobiles (Manoharan, 2019), fitness tracking (Fietkiewicz and Ilhan, 2020), and smart homes (Gochoo et al., 2021). In general, AI and machine learning (ML) are correlated. They are primarily used to develop intelligent systems (Das et al., 2015). Deep learning (DL) is a type of ML that allows computers to learn from data representation with more neural levels. Convolutional neural networks (CNN) have revolutionized image, video, and audio processing, and recurrent neural networks (RNN) have gained insight into text and speech sequential data (LeCun et al., 2015). The design of any deep learning model must consider the choice of algorithm. Most sequential applications follow the RNN model (Iqbal and Qureshi, 2022), and it has the context of previous input but not the future context of the speech or text data. Bidirectional recurrent neural networks (Bi-RNN) extract the context of data in both forward and backward directions (Schuster and Paliwal, 1997).

The proposed research offers substantial contributions to text analytics and natural language processing (NLP), particularly focusing on the complex issue of classifying Arabic poetry meters. This study employed Arabic text without removing diacritics from the poetry dataset. The 14 meters of the Arabic poem were considered. Two meters were removed because of very little data compared to other meters. The RNN models such as long short-term memory (LSTM), gated recurrent units (GRU), and Bi-RNN models, such as bidirectional LSTM (Bi-LSTM), are used to implement the proposed study. Despite the long history of Arabic poetry, automated techniques for meter classification have not received much attention. The proposed study utilized a large dataset and advanced neural network models. The main contribution of the study is defined as follows:

- Development of a DL framework utilizing TensorFlow for the categorization of Arabic poetry meter. The framework is specifically designed to categorize Arabic poetry meters, a field that presents linguistic and structural difficulties because of the complexity and variety of the Arabic language.
- Employing character-level encoding to transform text into integers for efficient categorization. This encoding enables the model to discern complex language patterns and nuanced differences at the character level, facilitating more efficient classification.
- To strengthen the robustness and usefulness of the classification methodology, the study employed both full-verse and half-verse types of Arabic poetry. This analysis allows the model to accurately identify poetry of diverse lengths and structural complexities, offering a thorough comprehension of Arabic poetic traditions.
- The research conducts an extensive assessment of several DL architectures, including LSTM, GRU, and Bi-LSTM, to determine the most efficient model for Arabic meter categorization. The Bi-LSTM model exhibited exceptional performance, attaining the greatest classification accuracy and highlighting its proficiency in managing the sequential and contextual intricacies of Arabic poetry.
- The findings of the study highlight the efficacy of DL techniques in tackling the complex nature of Arabic poetry meter classification. The research utilizes neural architectures and encoding methodologies to provide useful insights into the adaptation of existing NLP methods for the linguistically rich and morphologically complicated Arabic language.

The remaining section of this paper is organized into five sections. Section 2 explains the literature review, including Arabic meter and DL models. Section 3 describes the methodology used and the model algorithm. Section 4 presents the results in detail, with a discussion in section 5. Section 6 describes the conclusion with future study.

TABLE 1 Arabic diacritic types.

Diacritic	Types	Example
Harakat	"fatha" "dahmmah" "kasrah" "sukon"	شَرب الطفْلُ الحليبَ
Tanween	Tanween fateh, tanween dham and tanween kasr	بارداً, باردٌ, باردٍ
Dhawabet	Shad, mad	الشَّمس, آية

2 Literature review

Alnagdawi et al. (2013) used another tool for language recognition to find the meter of Arabic poems. This tool works in three steps: first, it converts poetry into Arud form. The second step is the segmentation of the Arud form. In this phase, the Arud state is divided into sounds, such as short sounds, vowel or long sounds, and consonants. The sound string was sent to the final stage at the end of the second step, and the poetry meter was detected. It is compared with grammar to check its validity. If the grammar is valid, the verse belongs to 16 meters. The meter patterns match the poem's words, identifying the meter's name.

A considerable body of literature is on recognizing Arabic poetry using deep learning algorithms. Baïna and Moutassaref (2020) developed an algorithm that accurately identifies the meter of the poem and outputs the 'Arud' writing in addition to the meter. The algorithm follows five phases. First, it adds diacritics to the verse. This

step is significant as it might impede moving to the next step. Second, it transforms the diacritics into 'Arud' writing. Third, it utilizes binary representation to convert the 'Arud' writing, where 1 represents a 'haraka' and 0 illustrates a 'sukon.' Fourth, the algorithm identifies the meter based on the binary representation. The fifth and final step includes detecting the errors and ensuring the meter matches the poem.

Furthermore, Albaddawi and Abandah (2021) proposed a narrow, deep neural network with significantly high accuracy. The proposed network consists of an embedding layer at its input, five Bi-LSTM layers, a concentration layer, and an output layer with softmax activation. Similarly, Abandah et al. (2020) suggested improving the recognition of diacritics via a specific neural network. This strategy tries to enhance readability and recognition accuracy. Moreover, identifying the meter of an Arabic poem may be a long and complicated process that involves a few steps (Al-shaibani et al., 2020). A study by Ahmed et al. (2019) utilized ML algorithms to identify and classify Arabic texts. The study supports linear vector classification and naïve Bayes classification, which showed the highest precision. Many studies have been conducted on analyzing Arabic poetry. Formulating one system or technique to identify meters in Arabic poetry is challenging. A study on identifying Arabic poetic meter (Saleh and Elshafei, 2012) suggested a method that produces coded Al-Khalili transcriptions of Arabic.

Abuata and Al-Omari (2018) electronically analyzed the Arud meter of Arabic poetry. They introduced an algorithm to determine the meter of Arud for any Arabic poetry. The algorithm works on well-defined rules applied only to the first part of the poem verse. Moreover, some of the most outstanding works in Arabic poetry are the computerization of Arabic poetry meters (Khalaf et al., 2009). It focuses on computerizing El-Katib's method for analyzing Arabic poetry. The linguist El-Katib proposed a study in which poetry is converted into binary bits and given decimal codes. This system was helpful for educational purposes. Many students and teachers use it to understand prosody. The computerized and systematic analysis of prosody also minimizes the chance of error.

Attempts have been made to develop algorithms that recognize modern Arabic poetry meters (Abandah et al., 2022; Abandah et al., 2020; Al-shaibani et al., 2020). For instance, an algorithm has been introduced to identify standard features of classical Arabic poems (Zeyada et al., 2020). These features include rhyme, rhythm, punctuation, and text alignment. This algorithm can only recognize whether the Arabic piece is poetic or non-poetic but cannot acknowledge its meter. Furthermore, an algorithm has been developed to detect the Arabic meter of certain poetry and convert the verse into 'Arud' writing (Al-Talabani, 2020). It classifies Arabic poetry using meters or 'Bahr' and investigates methods of detecting Arabic poems in rhythm, rhyme, and meter. It utilizes time and non-time series representation of the Mel-frequency cepstral coefficients (MFCC) and linear predictive cepstral coefficients (LPCC) features to recognize automated 'Arud' meters. Arabic 'Arud' meters seem to possess a time-series nature; however, the non-time series representation performs better.

Another detection method includes a comparison that has been conducted between modern and classical Arabic poetry (Almuhareb et al., 2015). The results reveal that contemporary Arabic poetry lacks more distinctive features than classical poetry. For instance, modern Arabic poetry is characterized by partial meter, the uneven lining of verses, word repetition, usage of punctuation, and irregular rhyming. At the same time, classical Arabic poetry is characterized by a regular rhyme, a single meter, even lining of verses, and self-contained lines. Similarly, Berkani et al. (2020) notes that extracting the meter of the poem using automatic meter detection methods requires challenging data collection and processing efforts. Syllable segmentation and similarity checks are performed. This method has further proven the high accuracy of meter detection. Finally, creating detecting algorithms may considerably improve the efficiency and accuracy of Arabic poetry identification methods.

The LSTM model is one of the most widely used RNN systems for vanishing gradients (Hochreiter and Schmidhuber, 1997). In addition, these networks have several advantages compared to conventional RNN systems, including the ability to sustain prolonged interrelationships and exhibit a stochastic nature when dealing with time-series input data. With RNN or LSTM, the uniform weight is retained across all layers, limiting the number of parameters the network must learn. The LSTM model had more parameters, which made it slower.

Later, GRUs were proposed as a better alternative to LSTMs and have gained significant recognition (Cho et al., 2014). In addition, GRUs have been recognized to be effective in numerous applications using sequential or time-series input (Dey and Salem, 2017). For instance, they have been incorporated in diverse areas such as speech synthesis, NLP, and signal processing. Furthermore, LSTM, RNNs, and GRUs have been exhibited to operate better in long-sequence applications. In GRUs, gating network signaling plays a significant role as it controls how inputs and memory are used to update current activations. Each gate has weights that are adapted and modified in the learning phase. However, these systems enable effective learning in RNNs, increasing parameterization. It leads to a simpler RNN model with a higher computational cost. The LSTM and GRU differ because the former utilizes three novel gate networks, whereas the latter uses only 2.

The Bi-LSTM neural network comprises LSTM units that operate in both directions to exploit contextual information from the past and future (Liang and Zhang, 2016). In addition, with Bi-LSTM, long-term dependencies can be learned without maintaining redundant background information. Thus, it has projected significant performance for sequential modeling issues and is generally used for text classification (Huang et al., 2015; Al-Smadi, 2024). Bi-LSTM networks transmit forward and reverse phases in both directions, unlike LSTM networks, which communicate only in one direction.

Many NLP sequences-to-sequence methods use LSTM, GRU, Bi-LSTM, and Bi-GRU deep learning models (Liang and Zhang, 2016; Wazery et al., 2022; Yin et al., 2017; Huang et al., 2015). In recent years, ML has become a formidable method for text analysis, exhibiting adaptability across several applications. Diverse ML methodologies have been effectively utilized in tasks such as dialect detection, spam detection, poetry classification, text classification, and sentiment analysis (Ahmed et al., 2019; El Rifai et al., 2022; Chen et al., 2022; Abdulghani and Abdullah, 2022; Alqasemi et al., 2021; Zivkovic et al., 2021), demonstrating their proficiency in managing intricate textual data.

An important use of ML is sentiment categorization, employed for the identification of insider threats. Recent studies by Mladenovic et al. (2024) have illustrated that sentiment analysis can be augmented through optimized classifiers, thereby enhancing the precision of

threat detection in organizational contexts. In spam email screening, NLP combined with ML has shown success (Bacanin et al., 2022). It explains how swarm intelligence can maximize conventional ML techniques, thereby improving user experience and spam detection accuracy. Another study by Kozakijevic et al. (2024) examined the incorporation of sentiment analysis in e-commerce, highlighting its significance in assessing seller reputation and influencing consumer choices. They attained a maximum accuracy of 88% by integrating transformer embeddings with an efficient extreme gradient boost model, refined via a modified firefly approach.

3 Materials and methods

The methodology of the study is shown in Figure 1. The key phases of the study include fetching the dataset, preprocessing and splitting the data, and developing and applying the DL models. The results were evaluated using a combination of accuracy, precision, recall, and the F1 score.

3.1 Dataset and preprocessing

The dataset contains 1,862,046 verses with 22 meters (Yousef et al., 2019). The data are in a well-structured format. The central 16 meters consist of a data size of 1,647,854. Two meters with fewer verses are avoided when classifying the meters. After eliminating the empty cells, the total number of verses in the 14 meters of data, which include both right and left verses, is 1,646,771. The count of each meter label with a full-verse is depicted in Figure 2. The minimum count is for the Mutadarik meter, 4,507 verses, and the maximum is for the Tawil meter, 398,239 verses. To address data scarcity for certain meters and improve the robustness of the models, half-verse data were doubled during training by treating the left and right verses of each meter as independent samples.

The dataset underwent a thorough cleaning process to enhance its quality and suitability for deep learning. Non-Arabic characters, symbols, and other irrelevant text artifacts were systematically removed. This step ensured that only meaningful linguistic content remained, aligning the dataset with the methodological requirements. The preprocessing methodology closely follows the approach described in Al-shaibani et al. (2020) including the construction of a character-level vocabulary. The character-level encoding uses the index value for each cleaned text and implements DL models. Parameter tuning was conducted for each deep learning model to optimize performance, with attention to hyperparameters such as learning rate, batch size, and sequence length. The data are split into 70% training and 15% validation; the remaining 15% are set as unseen data for testing.

3.2 Deep learning models

This study uses the deep neural network (DNN) architecture. The two main architectures of DNN are RNN and CNN (Yin et al., 2017). LSTM, GRU, and Bi-LSTM are models under RNN (Sherstinsky, 2020). The base model for LSTM consists of four layers. The first layer of the sequential model is the input layer with the size of the padded sequence, which is then given to the embedding layer with the output dimension kept as 64. The embedding layer will learn how to map the characters to vectors. The output from the embedding layer is fed into the LSTM layer with units 256, recurrent, and the activation function is set as the default. The LSTM layer is added accordingly to increase the hidden layers. At this moment, the return sequence parameter should be set as 'True.' The GRU model is like the LSTM model. In both models, sentence processing is only in one direction.

The LSTM layer is depicted in Figure 3. It allows the model to store the information for future access and has a hidden state: short-term memory. There are three gates for LSTM such as input (i_t), output (O_t), and forget gate (f_t). A time step is indicated by the subscript '*t*.' The LSTM has three inputs: an input vector at the current time stamp (X_t), a cell or memory state vector (C_{t-1}), and a hidden state vector at the previous time stamp (h_{t-1}). The symbol '×' denotes the element-wise product or the Hadamard product. $\widetilde{C_t}$ is the cell state activation vector or the candidate memory vector (Harrou et al., 2021).

As a first step, what information the cell state should discard should be determined. It is accomplished by the sigmoid activation

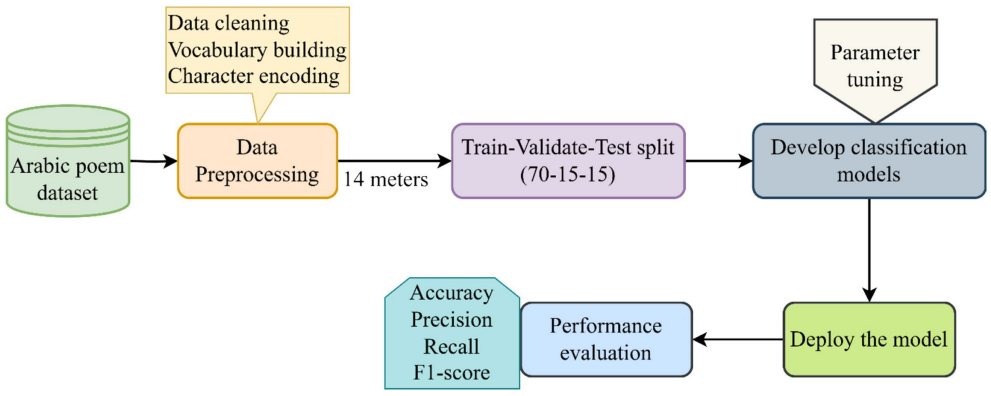

FIGURE 1
Overview of the research methodology.

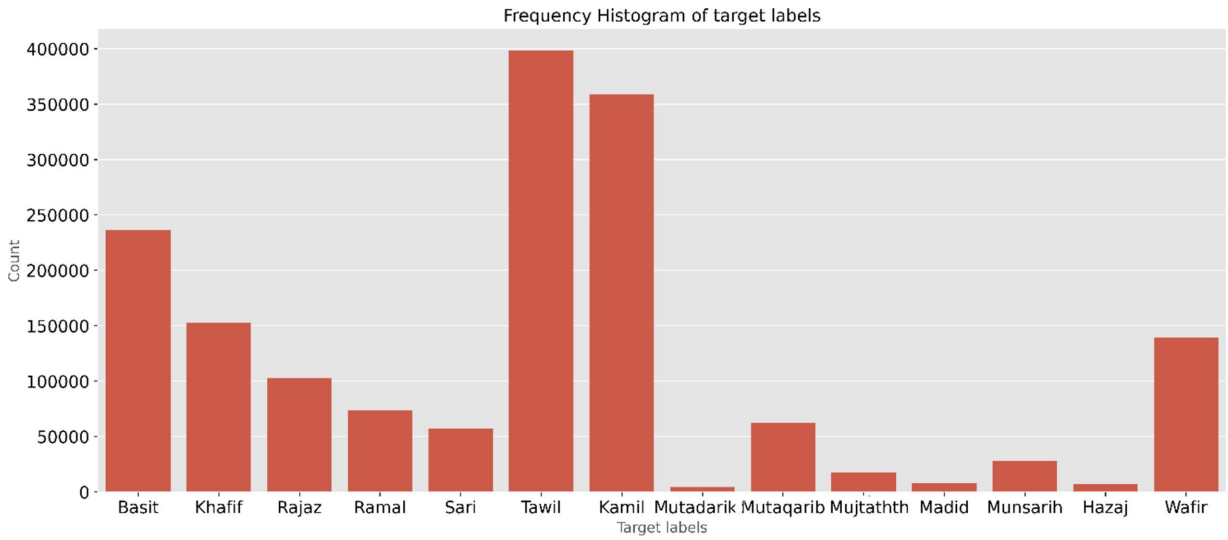

FIGURE 2
Full-verse count of the 14 meters in the dataset.

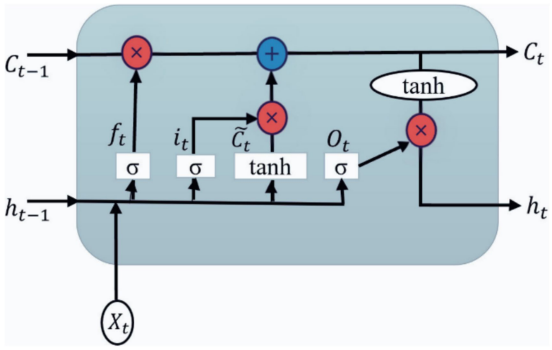

FIGURE 3
Internal architecture of the LSTM layer.

function (σ) in the forget gate and applies the sigmoid function to the current input vector X_t and the past hidden state vector h_{t-1} as shown in Equation 1. Input activations activate memory cells through input gates.

$$f_t = \sigma\left(w_f X_t + u_f h_{t-1} + b_f\right) \tag{1}$$

where f_t = forget gate, w_f and u_f are the weight matrices of the forget gate, X_t is the actual input, b_f is the bias vector, h_{t-1} is the hidden state output from the previous time stamp, and σ is the sigmoid activation function. The result from Equation 1 is in the range of 0 and 1. The element-wise product of C_{t-1} and f_t decides what information to retain and forget.

The second step is to update the memory cell with an input gate as shown in Equation 2. The sigmoid function indicates two values: if it is 1, the actual data are unchanged, and if it is 0, it will be dropped. A tanh function is applied to the selected input values, which indicates a range from −1 to +1. It creates a new vector of values, a candidate memory cell (Equation 3).

$$i_t = \sigma\left(w_i X_t + u_i h_{t-1} + b_i\right) \tag{2}$$

where i_t = input gate, w_i and u_i are the weight matrices of the input gate, b_i is the bias vector, X_t is the actual input, h_{t-1} is the hidden state output from the previous time stamp, and σ is the activation function.

$$\widetilde{C_t} = \sigma\left(w_c X_t + u_c h_{t-1} + b_c\right) \tag{3}$$

where $\widetilde{C_t}$ = candidate memory cell, w_c and u_c are the weight matrices, b_c is the bias vector, X_t is the actual input, h_{t-1} is the hidden state output from the previous time stamp, and σ is the activation function.

The following step involves updating and converting the previous cell state C_{t-1} to the new C_t. Equation 4 is defined as:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t} \tag{4}$$

where f_t = forget gate calculated from Equation 1, C_{t-1} is the memory state vector of the previous time stamp, i_t = input gate calculated from Equation 2, and $\widetilde{C_t}$ is the candidate memory cell from Equation 3.

The final stage is to decide what portion of the output will be selected. It is done in two steps. First, the sigmoid function is performed with the input to determine the quantity of cell state to transmit as the output (Equation 5). The tanh operation is then applied to the new cell state C_t, and the sigmoid result is multiplied by the result (Equation 6). Thus, the outcome is based only on the selected portions.

$$O_t = \sigma\left(w_o X_t + u_o h_{t-1} + b_o\right) \tag{5}$$

where O_t = output gate, w_o and u_o are the weight matrices of the output gate, b_o is the bias vector, X_t is the actual input, h_{t-1} is the hidden state output from the previous time stamp, and σ is the activation function.

$$h_t = \tanh(C_t) \cdot O_t \tag{6}$$

where O_t = output gate calculated from Equation 5, and new cell state C_t calculated from Equation 4.

The GRU layer is illustrated in Figure 4. A reset gate and an update gate are two gates. However, the GRU requires fewer parameters to train than the LSTM model, which runs faster. The reset gate (R_t) regulates the amount of the initial state that needs to be remembered. Similarly, an update gate (Z_t) enables us to assess how much the new form replicates the previous one. As each hidden unit reads/generates a sequence, these two gates control how much of it is remembered or forgotten (Harrou et al., 2021).

The reset gate performs similar functions to the forgotten gate of LSTM (Equation 7). It manages the short-term memory of the network. A decision is made regarding what information should be forgotten.

$$R_t = \sigma(w_r X_t + u_r h_{t-1} + b_r) \tag{7}$$

where R_t = reset gate, w_r and u_r are the weight matrices of the reset gate, b_r is the bias vector, X_t is the actual input, and h_{t-1} is the hidden state output from the previous time stamp.

The update gate manages the long-term memory of the network. It accomplishes a similar task as the forget and input gates of an LSTM. It determines what data should be removed and what new data should be added (Equation 8).

$$Z_t = \sigma(w_z X_t + u_z h_{t-1} + b_z) \tag{8}$$

where Z_t = update gate, w_z and u_z are the weight matrices of the update gate, b_z is the bias vector, X_t is the actual input, and h_{t-1} is the hidden state output from the previous time stamp.

The hidden state ($\widetilde{h_t}$) of the candidate is also called an intermediate memory unit, which combines the previously hidden state vector in the reset gate with the input vector (Equation 9).

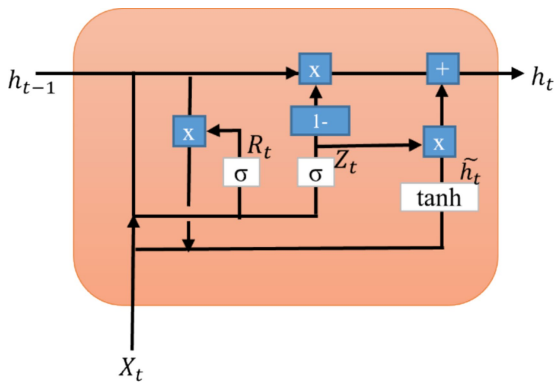

FIGURE 4
Internal structure of GRU layer.

$$\widetilde{h_t} = \tanh(w_h X_t + u_h(R_t \cdot h_{t-1}) + b_h) \tag{9}$$

where $\widetilde{h_t}$ = candidate hidden state vector, w_h and u_h are the weight matrices, b_h is the bias vector, R_t = reset gate calculated from Equation 7, X_t is the actual input, and h_{t-1} is the hidden state output from the previous time stamp.

The final hidden state is determined based on the update gate and candidate hidden state. The update gate is multiplied elementwise and summed with the candidate vector (Equation 10).

$$h_t = (1 - Z_t) \cdot h_{t-1} + \widetilde{h_t} \cdot Z_t \tag{10}$$

where h_t is the hidden state output, Z_t = update gate calculated from Equation 8, h_{t-1} is the hidden state output from the previous time stamp, and $\widetilde{h_t}$ = candidate hidden state vector calculated from Equation 9.

The Bi-LSTM model processes the sequence in both directions of a text. One hidden layer is in the forward movement, and the other is backward. These LSTM layers are concatenated for the final output of the Bi-LSTM layer. Hence, unit 256 is doubled in this model. The return sequence parameter of LSTM is set to 'True' if two or more layers need to be added. The dropout parameter in the Bi-LSTM layer is set to 0.2, which helps prevent the training model from overfitting. The hidden layers are tuned from 1 to 3 in all three models. A better iteration of LSTM is the Bi-LSTM layer, which processes the sequence in forwarding and backward directions, as shown in Figure 5. The Bi-LSTM can understand the context better than the LSTM and GRU models (Li et al., 2020), as it processes input sequences in both forward and backward directions. This architecture builds upon the traditional LSTM model, enhancing its ability to capture dependencies in sequential data. In the Bi-LSTM framework, X_t and X_{t+1} are the input vectors at time frame t.

While calculating the forward output sequence ($\overrightarrow{h_t}$), the positive sequence is used, and when calculating the backward output sequence, ($\overleftarrow{h_t}$), the reverse inputs are used. The output vector, y_t, is obtained by combining the forward and backward output sequences (Equation 11).

$$y_t = \int(\overrightarrow{h_t}, \overleftarrow{h_t}) \tag{11}$$

where $\overrightarrow{h_t}$ is the forward output sequence and $\overleftarrow{h_t}$ is the backward output sequence. The symbol '$\int$' can have different operations, such as summation, multiplication, concatenation, and average function. The default function in TensorFlow is concatenation.

The optimizer used for the compilation is adaptive moment estimation (Adam). This memory-light optimization algorithm works well with large datasets (Kingma and Jimmy, 2014). As the method label-encoder provides a sparse array of targets, the loss function uses a sparse-categorical cross-entropy.

3.2.1 Hyperparameter tuning

The tuned parameters are the hidden layer and learning rate for the above models. The hidden layers are tuned from 1 to 3 in all three DL models. EarlyStopping is used in the callback application

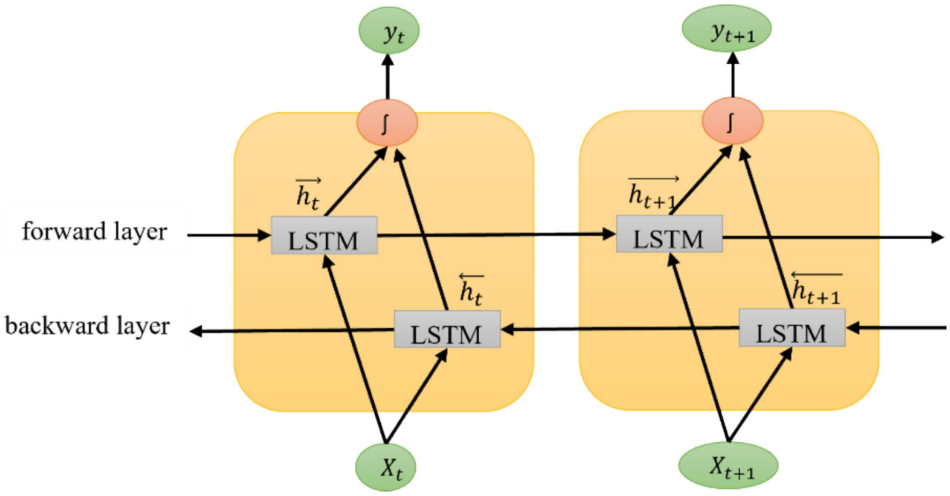

FIGURE 5
Bi-LSTM model architecture with two consecutive time frames.

programming interface (API) of the TensorFlow model to stop overfitting the models. In this, the parameter 'patience' is set to 6, so the training will terminate if the validation loss function does not decrease after six epochs. Another function used is ReduceLROnPlateau. The monitoring parameter of this function is set to validity loss, patience is 3, and the minimum learning rate is 1.0*10−6. It indicates that if the loss value does not change after two epochs, the learning rate value decreases by 0.1. Thus, the new rate for the next epoch will be 0.1 times the previous rate. The most accurate model is chosen based on the accuracy of the validation set, and it is then applied to the test set.

3.3 Evaluation metrics

Accuracy, precision, recall, and f1-score are the metrics used to assess the classification model on the test data. For each technique, the confusion matrix is also considered. Accuracy might not be a complete metric for unbalanced data (Sturm, 2013). Therefore, precision, recall, and F1-score are also used (Grandini et al., 2020; Tharwat, 2020). The precision determines how many predicted samples are relevant (Equation 12). Recall computes how many relevant samples are predicted (Equation 13). Calculating the harmonic mean of recall and precision yields an F1-score (Equation 14). Precision is also called a positive predictive rate (PPR), and recall is known as sensitivity. Accuracy is the total sample count that was successfully predicted (Equation 15). Four performance measures are calculated using the following formulas.

$$Precision = \frac{TruePose}{(TruePose + FalsePose)} \quad (12)$$

$$Recall = \frac{TruePose}{(TruePose + FalseNega)} \quad (13)$$

$$F1 - Score = \frac{2TruePose}{(2TruePose + FalsePose + FalseNega)} \quad (14)$$

$$Accuracy = \frac{TruePose + TrueNega}{TruePose + TrueNega + FalsePose + FalseNega} \quad (15)$$

where TruePose is a true positive, TrueNega is a true negative, FalsePose is a false positive, and FalseNega is a false negative. When the model correctly predicted the positive label, the result was considered TruePose. Similarly, if the model predicts a negative label correctly, the outcome is TrueNega. On the other hand, FalsePose is calculated based on the incorrectly predicted positive label, and FalseNega is based on the incorrectly predicted negative label.

4 Results

Neural networks formed the foundation of the classification models of the study, with DL techniques preferred due to the substantial volume of data involved. The experiments were conducted on a system running 64-bit Windows 10, equipped with an Intel® Core™ i7-4770K CPU at 3.50 GHz, 16 GB of RAM, and an NVIDIA GeForce GTX 1080 Ti GPU. The development environment utilized Python 3.9 and incorporated libraries such as TensorFlow 2.7 for implementing the DL models, Scikit-learn 1.0 for data preprocessing and evaluation, and PyArabic 0.6.14 for handling Arabic text processing (Abadi et al., 2016). This computational setup enabled efficient training and testing of the models, contributing to the high accuracy achieved in classifying the meters of classical Arabic poetry. The diacritics are not removed for both the full-verse and half-verse data.

4.1 Training and testing using full-verse data

The full-verse data are split according to 70% for training, 15% for validation, and 15% for testing. The validation accuracy

according to the hidden layers is tabulated in Table 2 for the full-verse data. In addition, the number of parameters the model uses for training is specified (in millions). The trainable parameter also increases; hence, the time taken to complete the execution also increases. The training epochs are set to 60 for all the models. Callback applications such as EarlyStopping and ReduceLROnPlateau evaluate whether the model overfits. The validation loss is the parameter to check in the ReduceLROnPlateau function. If the loss value is found stable for three epochs, then the learning parameter is increased. For the EarlyStopping function, the program stops where it finds the loss value increases from the previous value or is stable for approximately six epochs. The training epochs in Table 2 show the number of epochs each model took without overfitting the data. The LSTM, GRU, and Bi-LSTM models perform better at three layers. Moreover, compared to the three models, the Bi-LSTM shows an accuracy of 97.53%.

The training and validation loss and accuracy of the Bi-LSTM with three layers are depicted in Figure 6. The training loss indicates how well a DL model fits the training set. Validation loss measures the performance of the validation set. Accuracy increases as the loss value decreases.

The confusion matrix of the Bi-LSTM three-layer model is shown in Figure 7. The model was tested with the remaining 15% of unseen data. All the labels show good model fitting, and there was no overfitting or underfitting problem with the model performance.

The complete details of the model performance are shown in Table 3. The precision, recall, accuracy, and f1-score of each meter or label are evaluated. The basit and tawil meters show the highest accuracy of 99%. The low performance is demonstrated by the hazaj meter with 80% accuracy.

4.2 Training and testing using half-verse

The study also implemented the model based on the half-verse data without removing diacritics. The half-verse data count is double the number of full-verse data, and the data are split into 70% training, 15% validation, and 15% testing. The hidden layers are tuned from one to three as shown in Table 4. Increasing the layers increases the parameters to train the model. In addition, the time to complete the training increases according to hidden layers. Even though the Bi-LSTM model exists in 31 epochs, it took approximately 11 h to complete the execution.

The best model is Bi-LSTM, with 95.23% accuracy. The training and validation accuracy and loss values are shown in Figure 8. Both the loss and accuracy are inversely proportional to each other. The model exits from the iteration if the loss value is stable for six epochs.

The confusion matrix and the complete details of the target meters results are shown in Figure 9 and Table 5, respectively.

The model shows better performance as seen in Table 5. The highest class accuracy is demonstrated by the basit and tawil meters with 98% accuracy. The lowest performance is shown by the hazaj meter, which has 74% accuracy.

5 Discussion

The Bi-LSTM model predicts the data better when compared with LSTM and GRU. This model's sequence learning is in both directions, from left to right and right to left. GRU trains faster than LSTM, with fewer training parameters than LSTM (Atassi and El Azami, 2022). Few studies have been done on Arabic poetry, including the diacritization of the text data. The study by Abandah et al. (2022) showed a Bi-LSTM model with automatic diacritization. The results show a 42% improvement in the error rate of diacritization. The study by Alqasemi et al. (2021) was based on machine learning algorithms and a diacritic text. An accuracy of 96.34% was achieved using support vector machines (SVM). Another study by Al-shathry et al. (2024) employed a balanced dataset by randomly choosing 1,000 poem verses for each meter. Their study achieved 98.6% accuracy, but 90% precision, recall, and f1-score value with the Bi-GRU model.

The proposed study can be compared with the studies by Abandah et al. (2020) and Al-shaibani et al. (2020). With five hidden layers, Al-shaibani et al. (2020) reached an accuracy of 94.32% with the bi-directional GRU (Bi-GRU) model and 14 target meters. The model also attains 88.8% accuracy for half-verse data. With four hidden layers, the Bi-LSTM model by Abandah et al. (2020) achieved an accuracy of 97% without removing diacritics and 97.27% with removed diacritics. They use 16 meters as target classes. The study carried out by Yousef et al. (2019) used seven hidden layers for the Bi-LSTM model and achieved an accuracy of 96.38%. In the proposed

TABLE 2 The results of increasing the layers of each model on the test accuracy of full-verse data.

Models	Hidden layers	Parameters (in millions)	Accuracy	Training Epochs	Training time (in hours)
LSTM	1	0.34	0.9720	28	89.95
	2	0.86	0.9733	26	148.17
	3	1.38	0.9737	35	286.15
GRU	1	0.26	0.9710	28	166.93
	2	0.65	0.9723	37	212.63
	3	1.05	0.9726	60	455.93
Bi-LSTM	1	0.67	0.9698	19	110.02
	2	2.24	0.9744	26	249.97
	3	3.82	0.9753	25	442.50

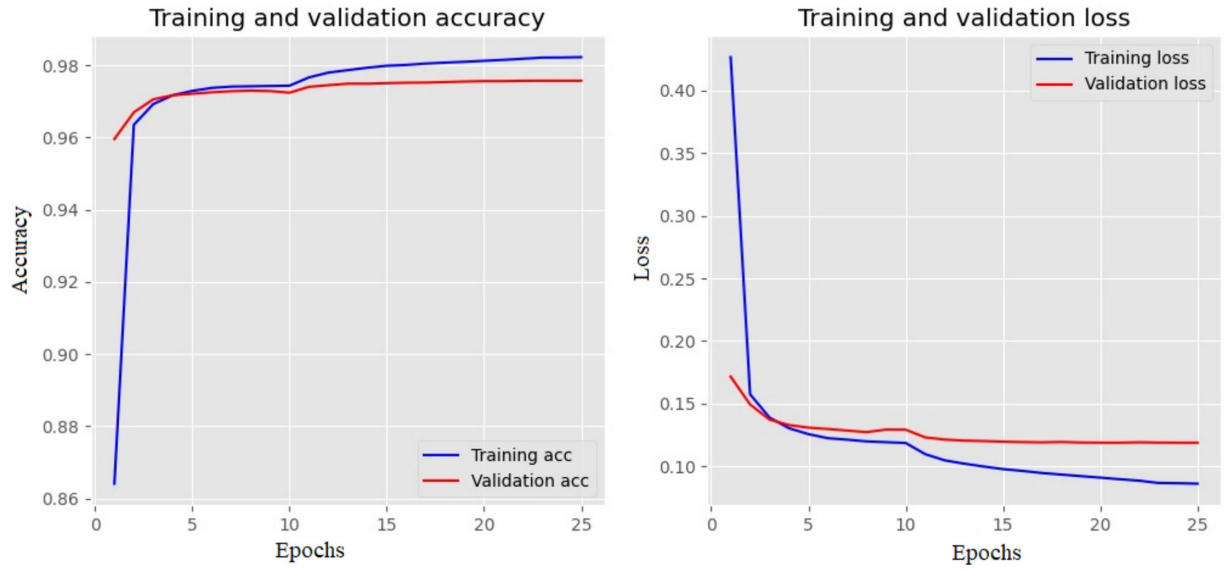

FIGURE 6
Training and validation plot of Bi-LSTM with three layers. The left side shows the accuracy, and the right shows the loss values for each epoch.

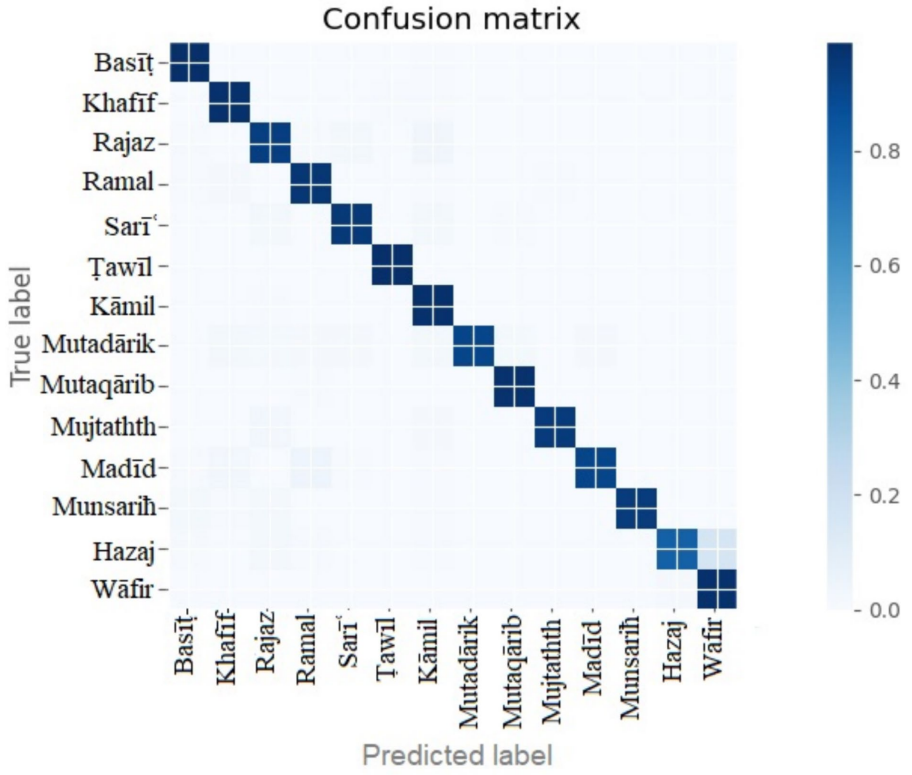

FIGURE 7
Confusion matrix of three hidden layers of the three-layer Bi-LSTM model.

research, the number of verses is much higher than in the study done by Al-shaibani et al. (2020). In addition, the number of hidden layers is less than in all three studies. The comparison of Arabic meter studies is mentioned in Table 6.

The studies (Abandah et al., 2020; Yousef et al., 2019) employed the identical dataset as the proposed study, although it documented varying verse counts. This suggests that although the dataset is uniform, discrepancies in verse counts may influence model efficacy.

The models employed in the compared research, Bi-LSTM with four and seven layers, attained competitive accuracy rates; nevertheless, the proposed Bi-LSTM model with three layers surpassed them across all criteria. The study by Al-shaibani et al. (2020) utilized a distinct dataset; however, it similarly extracted poems from the 'Aldiwan' website. The Bi-GRU model employed in the mentioned study (Al-shaibani et al., 2020) shows worse performance measures relative to the proposed study findings. The variations in dataset construction and model design certainly led to the noted performance variances.

In the proposed study, the Bi-LSTM model with three hidden layers performs better than one or two hidden layers without removing diacritical text. In addition, it better predicts than the LSTM and GRU models for both full-verse and half-verse data. LSTM cannot use future tokens nor can local contextual information be extracted. This problem can be resolved using Bi-LSTM, which learns the sequence in forward and backward directions. GRUs are faster to train than the LSTM model but lack the output gate. The model achieved an accuracy of 97.53% for the full-verse data and 95.23% for the half-verse data.

TABLE 3 Performance measure of the Bi-LSTM model with test data.

Meter	Precision	Recall	f1-score	Accuracy
Basit	0.98	0.99	0.99	0.99
Khafif	0.98	0.98	0.98	0.98
Rajaz	0.94	0.93	0.94	0.93
Ramal	0.96	0.96	0.96	0.96
Sari	0.95	0.95	0.95	0.95
Tawil	0.99	0.99	0.99	0.99
Kamil	0.97	0.98	0.98	0.98
Mutadarik	0.91	0.90	0.91	0.90
Mutaqarib	0.98	0.97	0.98	0.97
Mujtath	0.91	0.95	0.93	0.95
Madid	0.91	0.90	0.91	0.90
Munsarih	0.96	0.94	0.95	0.94
Hazaj	0.80	0.80	0.80	0.80
Wafir	0.98	0.98	0.98	0.98

The results of the study suggest that the number of hidden layers significantly impacts the performance of the Arabic meter classification model using Bi-LSTM. The study achieved better accuracy in Arabic meter classification using Bi-LSTM models with three hidden layers than previous studies that used Bi-LSTM models with four and seven hidden layers. It suggests that increasing the number of hidden layers beyond a certain point may not always lead to better performance and that optimizing the number of hidden layers can be a crucial factor in achieving high accuracy.

A few baseline ML models were utilized in this study to evaluate their performance in comparison with the DL architectures used for the Arabic poetry meters' classification. It includes a decision tree (DT), random forest (RF), k-nearest neighbors (KNN), and extra tree (ET) classifier. These classifiers serve as effective benchmarks for evaluating the performance of more complex models. The DT model yielded an accuracy of 46% with an F1-score of 0.30, and KNN achieved 30% with a 0.20 F1-score, while the ensemble models RF and ET achieved 58 and 53% accuracy as well as 0.50 and 0.56 F1-score values, respectively.

The comparison with baseline models underscores the efficacy of the DL methodologies utilized in the proposed study. Although baseline models serve as a valuable foundation, advanced models (Bi-LSTM) exhibit significant enhancements in accuracy and overall performance. This highlights the need to employ DL methodologies for intricate tasks such as Arabic poetry meter classification, where conventional models might struggle to grasp the complex nature of the data.

5.1 Practical implications

The findings of the proposed study on the categorization of Arabic poetry meter using DL models have substantial practical applications in several fields. This research enhances NLP, text analytics, and cultural heritage preservation by attaining high accuracy in the classification of full and half verses of Arabic poetry.

- Accurate classification of Arabic poetry meters helps preserve Arabic literary legacy. Automating the study of poetic structures helps scholars and cultural organizations to better classify historical data, therefore guaranteeing their availability for the next generations.
- The proposed DL system may be included in learning environments to support academics and students in

TABLE 4 The results of increasing the layers of each model on the test accuracy of half-verse data.

Models	Hidden layers	Parameters (in millions)	Accuracy	Training epochs	Training time (in hours)
LSTM	1	0.34	0.9465	34	153.23
	2	0.86	0.9494	24	166.08
	3	1.39	0.9509	28	283.33
GRU	1	0.26	0.9455	34	305.82
	2	0.65	0.9470	34	238.78
	3	1.05	0.9459	60	667.97
Bi-LSTM	1	0.67	0.9446	18	153.98
	2	2.24	0.9496	33	510.00
	3	3.82	0.9523	36	711.05

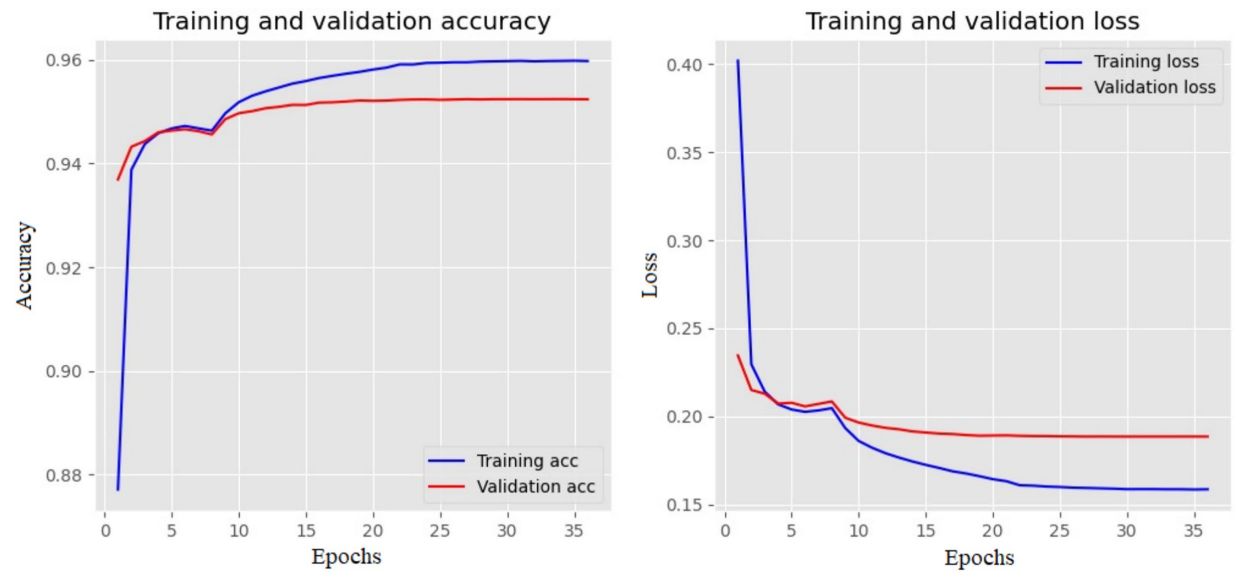

FIGURE 8
Training and validation plot of Bi-LSTM with three layers in half-verse. The left side shows the accuracy, and the right shows the loss values for each epoch.

comprehending Arabic poetry. By giving instantaneous feedback and poetic work analysis, interactive technologies that use meter classification can improve learning opportunities and help increase the importance of Arabic literature.

- Other kinds of Arabic literature can be examined using the approach developed in this study. Adapting the models to several literary genres allows scholars to investigate structures and patterns that define distinct kinds of Arabic literature, therefore enhancing the knowledge of the literary scene of the language.
- Using the knowledge acquired from the proposed study, NLP practitioners may increase the performance of the model in processing the Arabic text, therefore enhancing its applicability in fields such as social media analysis and automatic content development.

TABLE 5 Performance measure of the Bi-LSTM model with test data.

Meter	Precision	Recall	f1_ score	Accuracy
Basit	0.98	0.98	0.98	0.98
Khafif	0.96	0.96	0.96	0.96
Rajaz	0.88	0.83	0.85	0.83
Ramal	0.92	0.93	0.92	0.93
Sari	0.91	0.90	0.90	0.90
Tawil	0.99	0.98	0.98	0.98
Kamil	0.94	0.96	0.95	0.96
Mutadarik	0.84	0.83	0.83	0.83
Mutaqarib	0.95	0.96	0.95	0.96
Mujtath	0.86	0.89	0.87	0.89
Madid	0.84	0.82	0.83	0.82
Munsarih	0.93	0.89	0.91	0.89
Hazaj	0.71	0.74	0.73	0.74
Wafir	0.97	0.96	0.97	0.96

6 Conclusion

This study presents a significant advancement in the automatic classification of classical Arabic poetry meters using deep learning techniques. By utilizing a substantial dataset of 1,646,771 verses without removing diacritics, the Bi-LSTM models with three hidden layers were developed and evaluated. The Bi-LSTM model outperformed traditional LSTM and GRU models, achieving an accuracy of 97.53% on full-verse data and 95.23% on half-verse data. These results surpass those of previous studies that employed models with more hidden layers or smaller datasets.

The superior performance of the Bi-LSTM model underscores its effectiveness in capturing the complex rhythmic and phonetic patterns inherent in classical Arabic poetry. The ability of Bi-LSTM to process sequences in both forward and backward directions allows for a more comprehensive understanding of the linguistic structures involved. Importantly, retaining diacritics in the text preserved essential phonetic information, which proved crucial for accurate meter classification.

The findings of the study make a substantial contribution to computational linguistics and natural language processing, particularly in the context of Arabic language studies. The high accuracy achieved demonstrates the potential of the model for practical applications, such as automated literary analysis and educational tools that enhance the study and appreciation of Arabic poetry. This study also aligns with the Sustainable Development Goals by promoting quality education and fostering innovation in language technology.

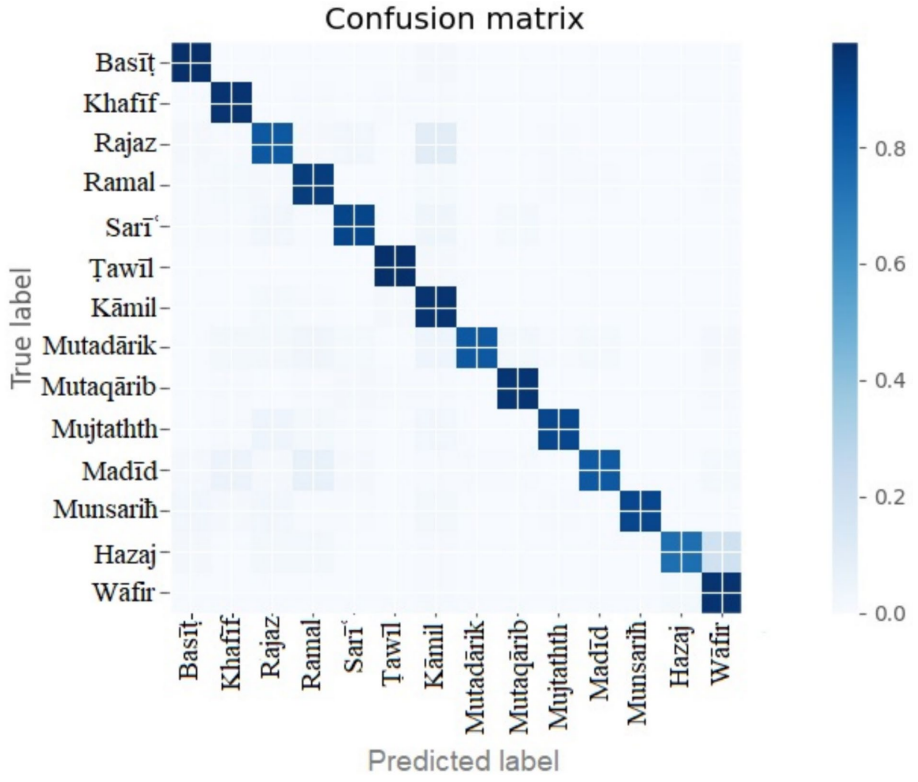

FIGURE 9
Confusion matrix of half-verse Bi-LSTM model.

TABLE 6 Comparison between related studies in literature and the proposed study.

Reference	Technique used—number of hidden layers	Dataset size	Accuracy	F1-score
Al-shaibani et al. (2020)	Bi-GRU-5	55,400 verses	94.32% (full-verse), 88.80% (half-verse)	-
Abandah et al. (2020)	Bi-LSTM-4	1,657,003 verses	97.27% (full-verse)	0.97 (full-verse)
Yousef et al. (2019)	Bi-LSTM-7	1,722,321 verses	96.38% (full-verse)	-
The proposed work	Bi-LSTM-3	1,646,771 verses	97.53% (full-verse), 95.23% (half-verse)	0.98 (full-verse), 0.95 (half-verse)

6.1 Limitations and future studies

The proposed study performs better with half-verse and full-verse Arabic poems. It indicates that although the average accuracy is elevated, some classes, especially those corresponding to meters with fewer verses, demonstrate diminished precision and recall. Future studies must concentrate on these underrepresented categories to enhance their classification efficacy. This can be accomplished using specific data augmentation procedures, such as the generation of synthetic examples of certain meters or the application of oversampling techniques to equilibrate the dataset.

Although several DL models were evaluated, their hyperparameters, such as optimizers and the number of units in layers, were not extensively tuned. Hyperparameter selection may greatly affect the model's performance. Future studies should consider using methodical hyperparameter tuning strategies to improve model performance. Another scope of future studies is to investigate the influence of other linguistic attributes on meter classification. It includes semantic and syntactic structure analysis.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: https://arxiv.org/abs/1905.05700.

Author contributions

AM: Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – review & editing. AA: Data curation, Investigation, Methodology, Software, Writing – original draft.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded by the Kuwait University Research (grant no. EO06/24).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016); TensorFlow: a system for large-scale machine learning. Available at: https://arxiv.org/abs/1605.08695 (Accessed April 5, 2024).

Abandah, G. A., Khedher, M. Z., Abdel-Majeed, M. R., Mansour, H. M., Hulliel, S. F., and Bisharat, L. M. (2020). Classifying and diacritizing Arabic poems using deep recurrent neural networks. *J. King Saud Univ. -Comp. Inform. Sci.* 34, 3775–3788. doi: 10.1016/j.jksuci.2020.12.002

Abandah, G. A., Suyyagh, A. E., and Abdel-Majeed, M. R. (2022). Transfer learning and multi-phase training for accurate diacritization of Arabic poetry. *J. King Saud Univ. -Comp. Inform. Sci.* 34, 3744–3757. doi: 10.1016/j.jksuci.2022.04.005

Abdulghani, F. A., and Abdullah, N. A. (2022). A survey on Arabic text classification using deep and machine learning algorithms. Iraqi journal of. *Science* 63, 409–419. doi: 10.24996/ijs.2022.63.1.37

Abuata, B., and Al-Omari, A. (2018). A rule-based algorithm for the detection of arud meter in classical Arabic poetry. *Int. Arab J. Inf. Technol.* 15, 1–5.

Ahmadoh, E. M., and Gutub, A. A.-A. (2015). Utilization of two diacritics for Arabic text steganography to enhance performance. *Lect. Notes Infor. Theory* 3, 42–47. doi: 10.18178/lnit.3.1.42-47

Ahmed, M. A., Hasan, R. A., Ali, A. H., and Mohammed, M. A. (2019). The classification of the modern arabic poetry using machine learning. *Telkomnika* 17, 2667–2674. doi: 10.12928/telkomnika.v17i5.12646

Albaddawi, M. M., and Abandah, G. A.. Pattern and poet recognition of Arabic poems using BiLSTM networks. IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). (2021):72–77. IEEE: Amman, Jordan

Almuhareb, A., Almutairi, W. A., Al-Tuwaijri, H., Almubarak, A., and Khan, M. (2015). Recognition of modern Arabic poems. *J. Softw.* 10, 454–464. doi: 10.17706/jsw.10.4.454-464

Alnagdawi, M., Rashaideh, H., and Aburumman, A. (2013). Finding Arabic poem meter using context free grammar. *J. Comm. Comput. Eng.* 3, 52–59. doi: 10.20454/jcce.2013.600

Alqasemi, F., Salah, A.-H., Abdu, N. A. A., Al-Helali, B., and Al-Gaphari, G. (2021). Arabic poetry meter categorization using machine learning based on customized feature extraction. International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE). 1–4. IEEE: Yemen

Al-shaibani, M. S., Alyafeai, Z., and Ahmad, I. (2020). Meter classification of Arabic poems using deep bidirectional recurrent neural networks. *Pattern Recogn. Lett.* 136, 1–7. doi: 10.1016/j.patrec.2020.05.028

Al-shathry, N., Al-onazi, B., Hassan, A. Q. A., Alotaibi, S., Alotaibi, S., Alotaibi, F., et al. (2024). Leveraging hybrid adaptive sine cosine algorithm with deep learning for Arabic poem meter detection. *ACM Trans. Asian Low-Resour. Lang. Inf. Process* 9:6963. doi: 10.1145/3676963

Al-Smadi, B. S. (2024). DeBERTa-BiLSTM: a multi-label classification model of Arabic medical questions using pre-trained models and deep learning. *Comput. Biol. Med.* 170:107921. doi: 10.1016/j.compbiomed.2024.107921

Al-Talabani, A. K. (2020). Automatic recognition of Arabic poetry meter from speech signal using long short-term memory and support vector machine. *ARO Sci. J. Koya Univ.* 8, 50–54. doi: 10.14500/aro.10631

Atassi, A., and El Azami, I. (2022). Comparison and generation of a poem in Arabic language using the LSTM, BILSTM and GRU. *J. Manag. Inform. Decis. Sci.* 25, 1–8.

Bacanin, N., Zivkovic, M., Stoean, C., Antonijevic, M., Janicijevic, S., Sarac, M., et al. (2022). Application of natural language processing and machine learning boosted with swarm intelligence for spam email filtering. *Mathematics* 10:4173. doi: 10.3390/math10224173

Baïna, K., and Moutassaref, H.. An efficient lightweight algorithm for automatic meters identification and error Management in Arabic Poetry. Proceedings of the 13th International Conference on Intelligent Systems: theories and Applications. Association for Computing Machinery: New York. (2020): 1–6.

Berkani, A., Holzer, A., and Stoffel, K.. Pattern matching in meter detection of Arabic classical poetry. 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA). (2020):1–8. IEEE: Antalya

Chen, H., Wu, L., Chen, J., Lu, W., and Ding, J. (2022). A comparative study of automated legal text classification using random forests and deep learning. *Inf. Process. Manag.* 59:102798. doi: 10.1016/j.ipm.2021.102798

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014); Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP):1724–1734. Association for Computational Linguistics: Doha, Qatar

Das, S., Dey, A., Pal, A., and Roy, N. (2015). Applications of artificial intelligence in machine learning: review and prospect. *Int. J. Comp. Appl.* 115, 31–41. doi: 10.5120/20182-2402

Davenport, T., and Kalakota, R. (2019). The potential for artificial intelligence in healthcare. *Future Healthc. J.* 6, 94–98. doi: 10.7861/futurehosp.6-2-94

Dey, R., and Salem, F. M.. Gate-variants of gated recurrent unit (GRU) neural networks. 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). (2017):1597–1600. IEEE: Boston

El Rifai, H., Al Qadi, L., and Elnagar, A. (2022). Arabic text classification: the need for multi-labeling systems. *Neural Comput. Applic.* 34, 1135–1159. doi: 10.1007/s00521-021-06390-z

Fietkiewicz, K., and Ilhan, A. Fitness tracking technologies: data privacy doesn't matter? The (un) concerns of users, former users, and non-users. Proceedings of the 53rd Hawaii International Conference on System Sciences. (2020). University of Hawaii: Honolulu, HI

Gochoo, M., Tahir, S. B. U. D., Jalal, A., and Kim, K. (2021). Monitoring real-time personal locomotion behaviors over smart indoor-outdoor environments via body-worn sensors. *IEEE Access* 9, 70556–70570. doi: 10.1109/ACCESS.2021.3078513

Grandini, M., Bagli, E., and Visani, G. (2020); Metrics for multi-class classification: an overview. Available at: https://arxiv.org/abs/2008.05756 (Accessed April 5, 2024).

Harrou, F., Sun, Y., Hering, A. S., Madakyaru, M., and Dairi, A. (2021). "Unsupervised recurrent deep learning scheme for process monitoring" in Statistical process monitoring using advanced data-driven and deep learning approaches. eds. F. Harrou, Y. Sun, A. S. Hering, M. Madakyaru and A. Dairi (London: Elsevier), 225–253.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. Available at: https://arxiv.org/abs/1508.01991 (Accessed April 5, 2024).

Iqbal, T., and Qureshi, S. (2022). The survey: text generation models in deep learning. *J. King Saud Univ. -Comput. Inf. Sci.* 34, 2515–2528. doi: 10.1016/j.jksuci.2020.04.001

Jones, A. (2011). Early Arabic poetry: select poems. Ithaca, NY: Ithaca Press.

Khalaf, Z., Alabbas, M., and Ali, S. (2009). Computerization of Arabic poetry meters. *UOS J. Pure App. Sci.* 6, 41–62.

Kharsa, R., Elnagar, A., and Yagi, S. (2024). BERT-based Arabic Diacritization: a state-of-the-art approach for improving text accuracy and pronunciation. *Expert Syst. Appl.* 248:123416. doi: 10.1016/j.eswa.2024.123416

Kingma, D. P., and Jimmy, B.. (2014); Adam: amethod for stochastic optimization. Available at: https://arxiv.org/abs/1412.6980 (Accessed April 5, 2024).

Kozakijevic, S., Jovanovic, L., Mihajlovic, M., Antonijevic, M., Jankovic, N., Radomirovic, B., et al. (2024). Consumer feedback sentiment classification improved via modified metaheuristic optimization natural language processing. *Int. J. Robot. Autom. Technol.* 11, 81–95. doi: 10.31875/2409-9694.2024.11.07

Lavzheh, T. (2009, 2024). A cognitive analysis of Mathnavi's Arabic poems. *J. Myst. Liter.* 16, 43–70. doi: 10.22051/jml.2023.44634.2497

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Deep Learn. Nat.* 521, 436–444. doi: 10.1038/nature14539

Li, Y., Harfiya, L. N., Purwandari, K., and Lin, Y.-D. (2020). Real-time Cuffless continuous blood pressure estimation using deep learning model. *Sensors* 20:20. doi: 10.3390/s20195606

Liang, D., and Zhang, Y. (2016); AC-BLSTM: Asymmetric convolutional bidirectional LSTM networks for text classification. Available at: https://ar5iv.labs.arxiv.org/html/1611.01884 (Accessed April 5, 2024).

Manoharan, S. (2019). An improved safety algorithm for artificial intelligence enabled processors in self driving cars. *J. Artif. Intell.* 1, 95–104. doi: 10.36548/jaicn.2019.2.005

Mladenovic, D., Antonijevic, M., Jovanovic, L., Simic, V., Zivkovic, M., Bacanin, N., et al. (2024). Sentiment classification for insider threat identification using metaheuristic optimized machine learning classifiers. *Sci. Rep.* 14:25731. doi: 10.1038/s41598-024-77240-w

Saleh, A.-Z. A. K., and Elshafei, M. (2012). Arabic poetry meter identification system and method. *US Patent* 8:219386.

Schuster, M., and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* 45, 2673–2681. doi: 10.1109/78.650093

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D* 404:132306. doi: 10.1016/j.physd.2019.132306

Sturm, B. L. (2013). Classification accuracy is not enough. *J. Intell. Inf. Syst.* 41, 371–406. doi: 10.1007/s10844-013-0250-y

Tharwat, A. (2020). Classification assessment methods. *Appl. Comput. Inform.* 17, 168–192. doi: 10.1016/j.aci.2018.08.003

Wazery, Y. M., Saleh, M. E., Alharbi, A., and Ali, A. A. (2022). Abstractive Arabic text summarization based on deep learning. *Comput. Intell. Neurosci.* 2022, 1–14. doi: 10.1155/2022/1566890

Wu, H., Han, H., Wang, X., and Sun, S. (2020). Research on artificial intelligence enhancing internet of things security: a survey. *IEEE Access* 8, 153826–153848. doi: 10.1109/ACCESS.2020.3018170

Yin, W., Kann, K., Yu, M., and Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. Available at: https://arxiv.org/abs/1702.01923 (Accessed April 5, 2024).

Yousef, W. A., Ibrahime, O. M., Madbouly, T. M., and Mahmoud, M. A. (2019). Learning meters of Arabic and English poems with recurrent neural networks: a step forward for language understanding and synthesis. Available at: https://arxiv.org/abs/1905.05700 (Accessed April 5, 2024).

Zeyada, S., Eladawy, M., Ismail, M., and Keshk, H.. A proposed system for the identification of modem Arabic poetry meters (IMAP). 2020 15th International Conference on Computer Engineering and Systems (ICCES). (2020):1–5. IEEE: Cairo, Egypt

Zivkovic, M., Stoean, C., Petrovic, A., Bacanin, N., Strumberger, I., and Zivkovic, T., (2021) A novel method for COVID-19 pandemic information fake news detection based on the arithmetic optimization algorithm. 2021 23rd international symposium on symbolic and numeric algorithms for scientific computing (SYNASC); 7–10. IEEE: Timisoara, Romania

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 01 May 2025
DOI 10.3389/frai.2025.1526293

OPEN ACCESS

EDITED BY
Motaz Saad,
Islamic University of Gaza, Palestine

REVIEWED BY
Nasredine Semmar,
CEA Saclay, France
Chatrine Qwaider,
Mohammed Bin Zayed University for Artificial Intelligence, United Arab Emirates

*CORRESPONDENCE
Jeehaan Algaraady
✉ jihan.amu@gmail.com
Mohammad Mahyoob
✉ eflu2010@gmail.com

RECEIVED 11 November 2024
ACCEPTED 31 March 2025
PUBLISHED 01 May 2025

CITATION
Algaraady J and Mahyoob M (2025) Exploring ChatGPT's potential for augmenting post-editing in machine translation across multiple domains: challenges and opportunities. *Front. Artif. Intell.* 8:1526293. doi: 10.3389/frai.2025.1526293

Exploring ChatGPT's potential for augmenting post-editing in machine translation across multiple domains: challenges and opportunities

Jeehaan Algaraady[1]* and Mohammad Mahyoob[2]*

[1]Department of Languages and Translation, Taiz University, Taiz, Yemen, [2]Department of Languages and Translation, Taibah University, Madina, Saudi Arabia

Introduction: Post-editing plays a crucial role in enhancing the quality of machine-generated translation (MGT) by correcting errors and ensuring cohesion and coherence. With advancements in artificial intelligence, Large Language Models (LLMs) like ChatGPT-4o offer promising capabilities for post-editing tasks. This study investigates the effectiveness of ChatGPT-4o as a natural language processing tool in post-editing Arabic translations across various domains, aiming to evaluate its performance in improving productivity, accuracy, consistency, and overall translation quality.

Methods: The study involved a comparative analysis of Arabic translations generated by Google Translate. These texts, drawn from multiple domains, were post-edited by two professional human translators and ChatGPT-4o. Subsequently, three additional professional human post-editors evaluated both sets of post-edited outputs. To statistically assess the differences in quality between humans and ChatGPT-4o post-edits, a paired *t*-test was employed, focusing on metrics such as fluency, accuracy, coherence, and efficiency.

Results: The findings indicated that human post-editors outperformed ChatGPT-4o in most quality metrics. However, ChatGPT-4o demonstrated superior efficiency, yielding a positive *t*-statistic of 8.00 and a *p*-value of 0.015, indicating a statistically significant difference. Regarding fluency, no significant difference was observed between the two methods (*t*-statistic = −3.5, *p*-value = 0.074), suggesting comparable performance in ensuring the natural flow of text.

Discussion: ChatGPT-4o showed competitive performance in English-to-Arabic post-editing, particularly in producing fluent, coherent, and stylistically consistent text. Its conversational design enables efficient and consistent editing across various domains. Nonetheless, the model faced challenges in handling grammatical and syntactic nuances, domain-specific idioms, and complex terminology, especially in medical and sports contexts. Overall, the study highlights the potential of ChatGPT-4o as a supportive tool in translation post-editing workflows, complementing human translators by enhancing productivity and maintaining acceptable quality standards.

KEYWORDS

post-editing, machine translation, ChatGPT-4o, natural language processing, artificial intelligence, LLMS

Introduction

Machine translation (MT) has a significant role in facilitating communication and enhancing global interactions. This role has gained more attention in various contexts, driven by remarkable natural language processing technology advancements that enabled more efficient translation (Raj et al., 2023). However, MT outputs must be post-edited to ensure their desired quality and meet productivity standards. Translation post-editing (TPE) is a critical step in the translation process that involves reviewing and refining machine-translated content. Post-editing is not a recent trend, and it emerged in the earlier days of MT (Vieira, 2019). Recently, post-editing MT gained considerable interest as a service and research topic due to the advancements in translation technology. Post-editing implies correcting grammatical errors in vocabulary, improving sentence structure, adjusting tone and style, ensuring cultural appropriateness, and refining the translation to align with the intended purpose and audience (Daems et al., 2013; Vardaro et al., 2019). Moreover, it allows for a more customized and tailored approach to translation, as post-editors can adapt the output to meet specific clients. According to Allen (2001), post-editing is correcting and refining the machine-generated translation (MGT) after translation from a source to a target language.

There are several types of post-editing, each catering to the number of corrections, efforts, and objectives required to achieve the desired translation. An early study on post-editing typology by Laurian (1984) proposed two types of post-editing: rapid post-editing and conventional post-editing. The former involves correcting the translated texts without paying attention to the translation style, while the latter implies deep correction to produce a human-like translation.

Allen (2003) suggests two types of post-editing: minimal and complete PEs. Minimal PE is for quick review, focusing mainly on critical errors and ensuring essential language accuracy, controlled by limited time and budget. However, complete PE aims to perform deep corrections closely resembling human translation standards.

van Egdom and Pluymaekers (2019) and Vieira (2017) established four levels of post-editing: "minimal," "light," "moderate," and "full," precisely. For post-editing quality guidelines, the Translation Automation User Society (TAUS, 2010) differentiates between two standards of expected target-text quality: "good enough" quality and quality "similar or equal to human translation." Indeed, these criteria almost correspond to "light" and "full" post-editing, respectively (Massardo et al., 2016). The TAUS guidelines stress that the level of post-editing depends on the deliberate purpose of the text and the quality of the raw MT output, making the target quality a more consistent factor for post-editing guidelines. Post-editors have no strict instructions about the issues they need to focus on. These instructions differ depending on whether they aim for "good enough" or "human translation quality." When machine translation (MT) errors impact meaning, for "good enough" quality, the focus is on semantics and comprehensibility, with less consideration given to syntactic or grammar. Conversely, post-editors should address style, syntax, grammar, and formatting issues when focusing on human translation quality. Additionally, they should handle terms that need to remain in the original language but may have been translated by the MT system.

In MT, post-editing has two paradigms, including static and interactive. In the former, the machine generates translation in the first step and then edits it in the second. The latter implies real-time collaboration between translators and MT systems (Vieira, 2019). In terms of these two paradigms, there are different findings; for example, Langlais and Lapalme (2002), in their TransType tool evaluation, evoked that interactive post-editing could lead to reduced productivity by up to 35% compared to static editing. Koehn et al. (2015) stated that interactive models with online learning seemed to require less technical effort, with post-editors becoming faster over time. However, it has also been proven that interactive post-editing may not notably affect target-text quality and could even result in errors (Underwood et al., 2014). Compared to static post-editing, interactive post-editing may take longer but result in higher-quality products (Green et al., 2014).

With the advent of advanced Neural Network systems, the generated translation becomes more accurate and naturally sounding (Qin, 2022). However, these translations still have inaccuracies, errors, and inappropriate phrasing. It is a vital step that bridges the gap between automated generated translation and human editors and linguistic expertise to enhance translation fluency, coherence, and linguistic appropriateness.

The collaborative interaction between artificial intelligence and human intervention offers a cost-effective and efficient approach to high-quality translation services in various domains where translation quality is critical, especially for legal, medical, and technical content. With the proliferation of these technologies, research on large language models (LLMs) and linguistic analysis, particularly in fields such as second language acquisition (Albuhairy and Algaraady, 2025), learner error analysis (Al-Garaady and Mahyoob, 2023), natural language processing (Mahyoob and Al-Garaady, 2018; Mahyoob, 2020), and academic writing development (Mahyoob et al., 2023), has become increasingly critical.

Though human post-editors of MGTs show high-quality products, their work is time-consuming, and they challenge both balanced speed and quality. This research investigates how ChatGPT-4o, an advanced language generation model, can enhance translation post-editing productivity, efficiency, and quality across various domains and how human editors benefit from ChatGPT-4o in their TPE tasks.

Research question

This work attempts to answer the following research questions as a starting point for exploring the role of ChatGPT-4o in various aspects of post-editing machine-generated translations.

1. Can ChatGPT-4o integration maintain human translators' productivity, consistency, and efficiency instead of a human editor during post-editing?
2. To what extent can ChatGPT-4o improve the overall quality of MGT through post-editing?
3. How does ChatGPT-4o's performance in post-editing compare to traditional post-editing methods?
4. What challenges and limitations are encountered when using ChatGPT-4o for post-editing in certain domains? Moreover, to what extent can these challenges be alleviated?

5. How much does using task-specific prompts improve ChatGPT-4o performance in PE?

Literature review

MTPE is the process of reviewing and correcting errors in machine-generated translations. This section provides an overview of the literature on translation post-editing and integrating language models like ChatGPT-4o in translation workflows. It discusses the challenges faced in translation post-editing, advancements in machine translation PE technologies, and the role of artificial intelligence in improving translation PE quality.

Screen (2019) compared post-edited translations with translations created from scratch in the Welsh text. He said post-translation editing was not found to improve. The two types of products are mainly similar in terms of comprehension and readability, which supports the use of MT in professional settings.

A study conducted with software instructions translated from English to Brazilian Portuguese found that even minimal post-editing significantly increased the usability of MT-based texts. The improvements were measured using eye-tracking metrics and self-reported satisfaction, highlighting the value of post-editing in enhancing text comprehensibility and accuracy (Castilho et al., 2014).

Koneru et al. (2023) made an Initial adjustment for direct translation. Therefore, researchers propose to use LLM as an automatic post editor (APE) instead. With Low-Rank-Adapter fine-tuning, they refined sentence- and document-level indicators. The ContraPro test achieved an accuracy of 89% in Anglo-German translations. In addition, including human corrections in document-level translations reduced the need for corrections in translation. Raunak et al. (2023) used GPT-4 for automatic post-editing in language pairs. It was found that there was an improvement in the accuracy and reliability of the WMT-22 English-Chinese, English-German, Chinese-English, and German-English tasks. However, sometimes GPT-4 might cause incorrect edits that demand caution in utilization. Chen et al. (2023) recommend improving iterative translation using large-scale language models for advanced translation and post-editing, especially for complex structures. However, this method showed limited scalability and computational challenges. Moreover, the model relies heavily on pre-trained models.

IntelliCAT, introduced by (Lee et al., 2021), is an interactive translation interface designed to improve post-machine translation editing. It uses sentence-level and word-level quality estimation (QE) to predict sentence quality and identify errors for improvement. The translation recommendation model includes word and phrase alternatives, while word alignments preserve the original document format. Experiments show that these features advance translation quality. User studies confirm that post-editing is 52.9% faster than translation from scratch. Turchi et al. (2017) explored machine translation (MT) improvements using human post-editing within a Neural Machine Translation (NMT) framework, highlighting the benefits of batch method customization. Continuously, It enables real-time optimization of new users and domains at low computational cost. Various online learning strategies are tested to refine existing models based on input data and after modification. Evaluating two language pairs showed a significant improvement over the static model.

Data collection and methodology

Data collection

To conduct our exploration, this research utilized translation data comprising source texts (English) and their corresponding Arabic MGTs produced by a neural network-based machine translator (Google Translator). This dataset spans different domains to simulate real-world translation scenarios, including sports, medical, business, idioms, and literary texts, to ensure a comprehensive assessment of ChatGPT-4o's potential across various domains. As detailed in Table 1, the source texts were collected from several online platforms such as UN news[1], Newatlas[2], Saudigazette[3], and American literature[4], comprising 6,203 English words (ws). Their Arabic translations produced by Google Translate [GT (A)] amount to 5,582 ws, while the human post-editing version [H-PE(A)] includes 5,393 ws, and the ChatGPT4o post-editing version [C- PE(A)] contains 5,451 ws.

Experiment/method

In this experiment, first, the collected texts undergo initial translation from English into Arabic using a neural network-based machine translator (Google translator) to establish a baseline for comparison. Second, the generated translations are post-edited in two modes, first by two professional human translators and then using ChatGPT-4o as a post-editing tool. ChatGPT-4o is requested to improve and revise the MGT to explore and assess the extent of ChatGPT-4o's capabilities in performing or enhancing post-editing machine-translated content. The two human translators were given different sets of data to post-edit to boost the diversity of post-edited translations and interpretations that reflect the Arabic richness and capture a broader range of editorial perspectives.

Third, a panel of three human editors (HEs) manually validated and evaluated the improvements and suggestions provided by human translators and ChatGPT-4o. Fourth, we compare the quality of the post-edited content by human translators and the quality of the post-edited content by ChatGPT-4o based on a set of evaluation metrics using *T*-test statistics. In addition, we compare the performance of ChatGPT-4o across different domains to assess its domain adaptation capabilities. Indeed, knowing ChatGPT-4o's ability to provide post-editing for machine translation would help make a clear decision to incorporate ChatGPT-4o's post-editing service for various stakeholders who benefit from post-editing translation.

1 https://news.un.org

2 https://newatlas.com/robotics/robot-designed-to-perform-breast-examination

3 https://www.saudigazette.com.sa/article/609348

4 https://americanliterature.com/

TABLE 1 Statistical description of the dataset.

Texts	Sports	Business	Medical	Literary	Total
Source (E)	1,580 ws	1,498 ws	1,539 ws	1,586 ws	6,203 ws
GT (A)	1,357 ws	1,283 ws	1,298 ws	1,564 ws	5,582 ws
H-PE (A)	1,332 ws	1,261 ws	1,258 ws	1,542 ws	5,393 ws
C- PE (A)	1,351 ws	1,268 ws	1,273 ws	1,559 ws	5,451 ws

TABLE 2 A sample of MGT, ChatGPT-4o's post-editing of MGT, and human's post-editing of MGT for business text.

S	The latest estimate is lower than the 3.1 percent GDP growth projected by the IMF in May
MGT	في المائة الذي توقعه صندوق النقد الدولي في مايو. 3.1 التقدير الأخير أقل من نمو الناتج المحلي الإجمالي بنسبة
ChatGPT-4oE	ويعتبر هذا التقدير الأخير أقل من التوقعات السابقة بنسبة ٣٪ لنمو الناتج المحلي الإجمالي التي أعلنها الصندوق النقد الدولي في شهر مايو.
HE	ويعتبر هذا التقدير الأخير لنمو الناتج المحلي الإجمالي اقل مما توقعه صندوق النقد الدولي سابقا في مايو بنسبة ٣٪.

TABLE 3 A sample of MGT, ChatGPT-4o's post-editing of MGT, and human's post-editing of MGT for idioms.

S	The shared interests provide a strong foundation for friendship, as friends engage in meaningful conversations and activities they enjoy, as said in "Birds Of A Feather Flock Together"
MGT	تشكل الاهتمامات المشتركة أساسًا قويًا للصداقة، حيث يشارك الأصدقاء الذين ينسجمون مع بعضهم البعض في محادثات وأنشطة هادفة يستمتعون بها. وكما يقولون: "الطيور تتجمع معًا
ChatGPT-4o E	"الاهتمامات المشتركة تشكل أساساً قوياً للصداقة، حيث يشارك الأصدقاء في محادثات ذات مغزى وأنشطة هادفة يستمتعون بها معًا، كما يقال: 'الطيور تتجمع معًا"
HE	"من منطلق المثل القائل: الطيور على اشكالها تقع, تشكل الاهتمامات المشتركة أساسًا قويًا للصداقة، حيث يستمتع الأصدقاء المنسجمون مع بعضهم بالمشاركة في محادثات وأنشطة هادفة.

Data analysis and evaluation

Evaluation measures for ChatGPT-4o and human post-editing of MGT across multiple domains

In this section, we analyze the impact of ChatGPT-4o on machine translation post-editing (MTPE). Based on this analysis, we attempt to identify patterns, challenges, and areas for improvement. We comprehensively compare the different post-editing modes (professional translator's post-editing and ChatGPT-4o post-editing) in terms of several key evaluation measures, including fluency, accuracy, efficiency, terminology, consistency, coherence, grammar, culture, and appropriateness. Generally, these criteria and standards are used to evaluate and improve the quality of translation as a machine product. Our analysis offers insights into ChatGPT-4o's ability to complement human expertise in post-editing, highlighting its strengths and limitations in enhancing the quality and efficiency of translation workflows.

After it is edited from a machine translation (MT) output, a text's linguistic smoothness and naturalness improve. These metrics focus on readability, grammar, syntax, and flow. As illustrated in Table 2, in terms of fluency (concentrate on readability, grammar, syntax, and flow), in the sentence extracted from a business text, the MGT version (a Google translate's generated translation) looks straight up, simple, and lacks fluency but still work as evaluated by HE. However, to some extent, when prompting ChatGPT-4o to evaluate the machine-generated translation MGT sentence structures for the source version (S), the ChatGPT-4oE version follows the natural flow of language compared to MGT, though it is not perfect like that in the HE version. ChatGPT-4oE provides a contextual version due to its conversational nature, enhancing the performance of translation studies. For accuracy, the ChatGPT-4o post-edited version shows proper punctuation usage. There are no spelling errors or typos, but there are slight errors in the translation grammar, including functional words usage such as articles as in ChatGPT-4oE phrase/"الصندوق النقد الدولي," "IMF"/, where it adds the article/"the,"ال" /in the word "الصندوق" inappropriately though it is correct in MGT version. However, the post-edited version by humans looks more cohesive as it maintains the coherence between sentences and paragraphs compared to the original version translated by Google Translate and the post-edited version by ChatGPT-4o.

Table 3 shows the output of the ChatGPT-4o post-editing of the literary text containing an idiomatic expression. It provides effective post-editing, showing substantial grammar, sentence structure, and readability improvements. However, it failed to maintain the idiomatic meaning for an Arabic audience. It provides accurate, unique literary phrases specific to the Arabic language that often carry cultural, historical, and contextual significance. As shown in the example below, both MGT and ChatGPT-4oE provide Arabic literary translation الطيور تتجمع" معًا for the idiom "Birds Of A Feather Flock Together." that is postedited by the human translator as/"Birds Of A Feather Flock Together, "الطيور على اشكالها تقع"/ and reflects a common challenge in AI's handling of nuanced and culturally specific elements and the need for AI to evolve beyond essential linguistic translations to encompass cultural interpretations, demanding more sophisticated training datasets and model enhancements.

As illustrated in Table 4, in the case of the medical terms, ChatGPT-4o failed to post-edit and provide the accurate Arabic

TABLE 4 A sample of medical MGT, ChatGPT-4o's post-editing of MGT, and human's post-editing of MGT medical texts.

S	If a person with astigmatism experiences a healthcare error, it may affect their overall wellbeing and exacerbate existing vision issues
MGT	إذا تعرض الشخص المصاب بالاستجماتيزم لخطأ في الرعاية الصحية، فقد يؤثر ذلك على صحته العامة ويؤدي إلى تفاقم مشكلات الرؤية الحالية.
ChatGPT-4oE	إذا تعرض الشخص المصاب بالاستجماتيزم لخطأ في الرعاية الصحية، فقد يؤثر ذلك على صحتهم العامة وقد يزيد من تفاقم مشكلات الرؤية الحالية.
HE	إذا تعرض الشخص المصاب بالابؤرية لخطأ طبي، فقد يؤثر ذلك على صحته العامة و يفاقم مشكلات الرؤية الحالية.

TABLE 5 A sample of MGT, ChatGPT-4o's post-editing of MGT, and human's post-editing of MGT for a literary text.

S	I had an elderly neighbor whose conversations were mind-blowing. He told me one day when his gathering was devoid of people, and no one was sitting with us: "Oh my son, a secret has been stirring in my heart for forty years regarding what has been discussed during this lengthy period
MGT	كان لدي جار مسن كانت محادثاته مذهلة. قال لي ذات يوم وقد خلا ناديه من الناس ولم يكن بيننا من يجلس: يا بني إن سر ما ذكر في هذه المدة الطويلة ما زال يتحرك في قلبي منذ أربعين سنة.
ChatGPT-4oE	1. كان لدي جار كبير في السن كانت محادثاته مذهلة. قال لي يوماً وكان ناديه خالياً من الناس ولم يكن هناك أحد يجلس بيننا: يا إبني، كانت أسرار ما تم ذكره خلال هذه الفترة الطويلة تحرك في قلبي لمدة أربعين عامًا.
HE	كان لي جار من الكهول، يخلب بسماع أحاديثه العقول، فقال لي ذات يوم — وقد خلا ناديه من الحضور، ولم يكن معنا فيه أحد من الجلَّاس: يا بُنيَّ، إنه يختلج في صدري منذ أربعين سنة سرٌّ ما جرى ذكره في هذه المُدَّة الطويلة على الألسنة.

TABLE 6 Sample of MGT, ChatGPT-4o post-editing of MGT, and human post-editing of MGT for sports text.

S	Tsuyoshi Kitazawa, a former member of Japan's national football team, stressed the role of sport in building bridges: "whatever you feel in the Games is made possible because the world is playing as one team," he said
MGT	وشدد تسويوشي كيتازاوا، العضو السابق في المنتخب الوطني الياباني لكرة القدم، على دور الرياضة في بناء الجسور: "كل ما تشعر به في الألعاب أصبح ممكنا لأن العالم يلعب كفريق واحد".
ChatGPT-4oE	تسويوشي كيتازاوا، عضو سابق في منتخب اليابان الوطني لكرة القدم، أكد دور الرياضة في بناء الجسور: "ما تشعر به في الألعاب يصبح ممكنًا لأن العالم يلعب كفريق واحد"،
HE	واكد العضو السابق في المنتخب الوطني الياباني لكرة القدم، تسويوشي كيتازاوا، على دور الرياضة في بناء جسور التواصل قائلا: " كل ما يجول بخاطرك حول الألعاب الرياضية أصبح ممكنا لأن العالم أصبح يلعب كفريق واحد".

equivalents for some medical terms; for example, ChatGPT-4oE provided literary translation for the phrase "healthcare errors" as "أخطاء الرعاية الصحية" of which the correct Arabic equivalent is "خطاء طبية." In some cases, ChatGPT-4oE failed to provide any translation for these terms, such as the word "astigmatism.". Instead, it gives transliteration for the term as "الاستجماتيزم." while the correct Arabic translated term is "الابؤرية " as edited by HE. In addition, there is a linguistic agreement error as seen in the anaphor (possessive pronoun "their") in the phrase/"their health," "صحتهم"/ which should be/"his health," "صحته"/ since this phrase refers to the singular antecedent/"a person," "الشخص"/. However, the anaphor generated by MT agreed with its antecedent. Compared to human editors, ChatGPT-4o failed to ensure and improve consistency in terminology and medical terms throughout the text.

ChatGPT-4o struggles to produce an efficient translation in the case of literary texts, as seen in Table 5 below. There is a grammatical error where the singular noun "a secret" in the phrase "a secret has been …" is translated inappropriately to plural noun /"secrets," "اسرار"/which should be translated to the Arabic singular noun "سر." Also, the syntactic structures look inferior compared to MGT and HE versions. ChatGPT-4oE, in the case of literary texts, shows significant issues in using correct and consistent terms and looks poor in its language smoothness and naturalness, cohesion, grammar, cultural aspects, and terminology handling.

Table 6 shows that ChatGPT-4o failed to appropriately edit the phrase (' in building bridges, "في بناء الجسور") and provide the same MGT version (literal translation for this phrase). However, the HE version/"in building bridges," "في بناء جسورالتواصل"/demonstrates a deeper and more accurate understanding and use of consistent terms. All these emphasize using ChatGPT-4o with caution in the translation industry because the HE edition emphasizes promoting proper contact and understanding between people, which is often implied when discussing "Building Bridges." This version not only maintains the source phrase's true meaning but also enriches the meaning by adding a more nuanced layer of meaning that is more appropriate and resonant for the reader. In the case of the phrase/"whatever you feel in the Games," "كل ما يجول بخاطرك حول الألعاب الرياضية"/, both MGT and ChatGPT-4o provide unnatural and inconsistent translation version/"ما تشعر به في الألعاب" "كل ما تشعر به في الألعاب,"/compared to that provided by HE version.

This demonstrates that ChatGPT-4o fails to communicate the deeper intent to the audience effectively. ChatGPT-4o provides accurate numbers, information, and proper names. However, concerns include sentence structure using compound words, function words, and word ordering, as seen in Table 6. All of this highlights the careful use of ChatGPT-4o in the translation industry.

Prompt engineering for enhancing ChatGPT-4o outcomes

Mostly, it is noticed that the performance of ChatGPT-4o becomes more meaningful and more profound when we specify the needs and provide context, background, and a comprehensive

TABLE 7 ChatGPT-4o post-editing with business texts after prompt engineering.

S	The latest estimate is lower than the 3.1 percent GDP growth projected by the IMF in May
MGT	التقدير الأخير أقل من نمو الناتج المحلي الإجمالي بنسبة 3.1 في المائة الذي توقعه صندوق النقد الدولي في مايو.
ChatGPT-4oE	1. ويعتبر هذا التقدير الأخير أقل من التوقعات السابقة بنسبة ٣٪ لنمو الناتج المحلي الإجمالي التي أعلنها الصندوق النقد الدولي في شهر مايو. 2. من قبل صندوق النقد الدولي في شهر مايو 3.1% و يعتبرالتقدير الأخير أقل من معدل نمو الناتج المحلي الإجمالي المتوقع عند.
HE	ويعتبر هذا التقدير الأخير لنمو الناتج المحلي الإجمالي اقل مما توقعه صندوق النقد الدولي سابقا في مايو بنسبة ٣٪.

TABLE 8 ChatGPT-4o outcomes in literary texts after prompt engineering.

S	I had an elderly neighbor whose conversations were mind-blowing. He told me one day when his gathering was devoid of people, and no one was sitting with us: "Oh my son, a secret has been stirring in my heart for forty years regarding what has been discussed during this lengthy period
MGT	كان لدي جار مسن كانت محادثاته مذهلة. قال لي ذات يوم وقد خلا ناديه من الناس ولم يكن بيننا من يجلس: يا بني إن سر ما ذكر في هذه المدة الطويلة ما زال يتحرك في قلبي منذ أربعين سنة.
ChatGPT-4oE	1. كان لدي جار كبير في السن كانت محادثاته مذهلة. قال لي يوماً وكان ناديه خالياً من الناس ولم يكن هناك أحد يجلس بيننا: يا إبني، كانت أسرار ما تم ذكره خلال هذه الفترة الطويلة تحرك في قلبي لمدة أربعين عامًا. 2. كان لدي جار مسن، محادثاته كانت استثنائية. في يوم ما كان ناديه خالياً من الناس و لم يكن هناك أحد آخرمعنا خلال تجمعنا، قال لي: 'يا بني، هناك سر يدور في قلبي منذ أربعين عامًا يشأن المناقشات التي جرت خلال هذه الفترة الطويلة.
HE	كان لي جار من الكهول، يخلب بسماع أحاديثه العقول، فقال لي ذات يوم — وقد خلا ناديه من الناس، ولم يكن معنا فيه أحد من الجلّاس: يا بُنيّ، إنه يختلج في صدري منذ أربعين سنة سرٌّ ما جرى ذكره في هذه المُدَّة الطويلة على الألسنة

TABLE 9 ChatGPT-4o post-editing in medical after prompt engineering.

S	If a person with astigmatism experiences a healthcare error, it may affect their overall wellbeing and exacerbate existing vision issues
MGT	إذا تعرض الشخص المصاب بالاستجماتيزم لخطأ في الرعاية الصحية، فقد يؤثر ذلك على صحته العامة ويؤدي إلى تفاقم مشكلات الرؤية الحالية.
ChatGPT-4oE	1. إذا تعرض الشخص المصاب بالاستجماتيزم لخطأ في الرعاية الصحية، فقد يؤثر ذلك على صحتهم العامة وقد يزيد من تفاقم مشكلات الرؤية الحالية. 2. اذا تعرض شخص مصاب بالاستجماتيزم لخطأ في الرعاية الصحية, فقد يؤثر ذلك على صحته العامة وقد يؤدي الى تفاقم مشكلات الرؤية الحالية.
HE	إذا تعرض الشخص المصاب بالابؤرية لخطأ طبي، فقد يؤثر ذلك على صحته العامة و يفاقم مشكلات الرؤية الحالية.

input "prompt." For example, giving these details, "post-edit the Arabic generated translation below from the linguistic perspective, take the role of a professional grammar corrector, identify business terms, avoid changing meaning as much as possible" to the prompt enhances the tool's outputs. This can be seen in the improvements in ChatGPT-4oE 2 in Table 7, where the article "the, ال" is appropriately used compared to that in the ChatGPT-4oE 1 in the phrase/اصندوق النقد "الدولي," "IMF"/.

When we give these details "post-edit the Arabic generated-translation below from the linguistic perspective, take the role of a professional grammar corrector, identify idiomatic phrases, avoid changing meaning as much as possible" to the prompt of ChatGPT-4o in the literary texts, ChatGPT4o corrects its translation and post-editing. The yield results were more natural and accurate, as seen in ChatGPT-4o E 2 in Table 8, which shows improvement in the sentence flow compared to ChatGPT-4oE 1 due to some grammatical and stylistic adjustments. For example, the Arabic equivalent of the word "elderly" looks more fluent in the ChatGPT-4o E 2 version as "مسن" compared to that in the ChatGPT-4o E 1 "كبير في السن". Also, the grammatical mistake in the ChatGPT-4o E 1 version is spotted in the ChatGPT-4o E 2 version, as the word "secret" is translated to a singular noun "سر" instead of plural noun "اسرار" like that in ChatGPT-4o E 1.

ChatGPT-4oE 1, in Table 9, displays the result of ChatGPT-4o outcomes when the prompt is "post-edit." At the same time, ChatGPT-4oE 2 shows the ChatGPT-4o outcomes with a comprehensive prompt, "post-edit the Arabic generated translation below from the linguistic perspective, take the role of a professional grammar corrector, identify medical terms, avoid changing meaning as much as possible." As seen in ChatGPT-4oE 2, the tool still shows a deficiency in providing the correct Arabic medical translated terms such as "الابؤرية" and "خطأ طبي" for the English medical terms "astigmatism" and "healthcare," even though the tool is provided with a comprehensive prompt. The output in ChatGPT-4oE 2 looks identical to that provided without prompt engineering except for the omission of the article "the, ال" in words "person, شخص" and "affected, مصاب". We notice grammatical and stylistic improvements in the ChatGPT-4oE 2 version compared to the ChatGPT-4oE 1 version, for example, the linguistic agreement error in the anaphora (possessive pronoun 'their') in the phrase/"their health," "صحتهم"/is correctly translated to/"his health," "صحته"/.

In Table 10, the ChatGPT-4o E 2 version shows an enhanced, fluent, and natural post-editing that highlights the role of prompt engineering in raising the tool's advanced linguistic capabilities. This version shows an accurate idiomatic expression, particularly after adding a perspective and a contextual background to our prompt. Interestingly, ChatGPT-4o delivers a precise and culturally appropriate Arabic translation, "الطيور على أشكالها تقع" for the English idiom "Birds Of A Feather Flock Together". However, the tool failed

TABLE 10 ChatGPT-4o post-editing with idioms after prompt engineering.

S	The shared interests provide a strong foundation for friendship, as friends engage in meaningful conversations and activities they enjoy, as said in "Birds Of A Feather Flock Together"
MGT	"تشكل الاهتمامات المشتركة أساسًا قويًا للصداقة، حيث يشارك الأصدقاء الذين ينسجمون مع بعضهم البعض في محادثات وأنشطة هادفة يستمتعون بها. وكما يقولون: "الطيور تتجمع معًا.
ChatGPT-4o E	1. "الاهتمامات المشتركة تشكل أساساً قوياً للصداقة، حيث يشارك الأصدقاء في محادثات ذات مغزى وأنشطة هادفة يستمتعون بها معًا، كما يقال: "الطيور تتجمع معًا.". 2. "تشكل الاهتمامات المشتركة أساسًا قويًا للصداقة، حيث يشارك الأصدقاء الذين يتألفون في محادثات وأنشطة هادفة يستمتعون بها. وكما يقول المثل: "الطيور على أشكالها تقع".".
HE	"من منطلق المثل القائل: الطيور على اشكالها تقع, تشكل الاهتمامات المشتركة أساسًا قويًا للصداقة، حيث يستمتع الأصدقاء المنسجمون مع بعضهم بالمشاركة في محادثات وأنشطة هادفة".

TABLE 11 ChatGPT-4o post-editing with idioms after prompt engineering.

S	Tsuyoshi Kitazawa, a former member of Japan's national football team, stressed the role of sport in building bridges: "whatever you feel in the Games is made possible because the world is playing as one team," he said
MGT	"وشدد تسويوشي كيتازاوا، العضو السابق في المنتخب الوطني الياباني لكرة القدم، على دور الرياضة في بناء الجسور: "كل ما تشعر به في الألعاب أصبح ممكنا لأن العالم يلعب كفريق واحد".
ChatGPT-4oE	1. "تسويوشي كيتازاوا، عضو سابق في منتخب اليابان الوطني لكرة القدم، أكد دور الرياضة في بناء الجسور: "ما تشعر به في الألعاب يصبح ممكنًا لأن العالم يلعب كفريق واحد"، 2. وشدد تسويوشي كيتازاوا، اللاعب السابق في المنتخب الوطني الياباني لكرة القدم، على أهمية دور الرياضة في بناء الجسور قائلاً: "كل ما تشعر به خلال الألعاب أصبح ممكناً لأن العالم يلعب كفريق واحد".
HE	"واكد العضو السابق في المنتخب الوطني الياباني لكرة القدم، تسويوشي كيتازاوا، على دور الرياضة في بناء جسور التواصل قائلا: " كل ما يجول بخاطرك حول الألعاب الرياضية أصبح ممكنا لأن العالم أصبح يلعب كفريق واحد".

TABLE 12 Human evaluator's scores for ChatGPT-4o and human post-editing performance across various.

Evaluators	Post-editors	Fluency	Accuracy	Efficiency	Terminology	Consistency	Cohesion	Syntax	Grammar	Cultural appropriateness
EV1	ChatGPT-4o	4	4	5	3	3	2	4	4	3
	Human	5	5	2	5	4	5	5	5	5
EV2	ChatGPT-4o	3	4	5	3	2	2	4	3	4
	Huma	5	5	2	5	5	5	5	5	5
EV3	ChatGPT-4o	4	3	5	3	3	3	4	3	3
	Huma	5	5	3	5	5	5	5	5	5

earlier in providing the appropriate Arabic equivalent idiomatic expression, as shown in ChatGPT-4o E 1.

In Table 11, the ChatGPT-4oE 2 version resulted after providing the tool this enhanced prompt, "post-edit the Arabic generated translation below from the linguistic perspective, take the role of a professional grammar corrector, identify sport terms, avoid changing meaning as much as possible". However, the structure of this version looks better; like ChatGPT-4oE 1 version, it failed to provide a suitable translation for the phrases, /"in building bridges," "في بناء جسورالتواصل"/ and / "whatever you feel in the Games," كل ما يجول بخاطرك حول الألعاب الرياضية/ that highlights the limited role of ChatGPT4o in providing satisfied translation in specific sport-terms as some expressions require deep understanding.

It is worth mentioning that when the tool was asked to spot mistakes and explain the corrections it made, it did not identify all the errors from the first prompt and often lacked in-depth explanations. Moreover, at times, it hallucinated, providing incorrect or irrelevant details. Thus, when the tool is applied to medical, legal, financial, or technical texts, this adequate performance, even slight errors or ambiguity, would cause damage consequences. Therefore, while the tool is valuable, it requires care and validation in high-stakes contexts.

Results and discussion

ChatGPT-4o's post-editing and human post-editing performance were evaluated by three human evaluators (EV1, EV2, EV3) across several linguistic aspects: Fluency, Accuracy, Efficiency, Terminology, Consistency, Cohesion, Syntax, Grammar, and Cultural for performing the quantitative and qualitative analysis. The results are measured on a 5-point Likert scale where 1 = Poor, 2 = Fair, 3 = Good, 4 = Very Good, and 5 = Excellent. After collecting the evaluators' rating scores, we applied a paired *t*-test for our statistical analysis because of its effectiveness in comparing differences between ChatGPT-4o and human post-editing and determining whether the observed differences were statistically significant, providing a reliable and quantitative assessment of the comparative performance, the average score for each aspect is depicted in Table 12.

The box-and-whisker plot in Figure 1 shows the average ratings for ChatGPT-4o and human post-editing across nine evaluation metrics, showing that human post-editing consistently outperforms ChatGPT-4o in terms of performance, with significantly higher ratings in all categories except efficiency. This highlights the superiority of human editors in maintaining quality, accuracy,

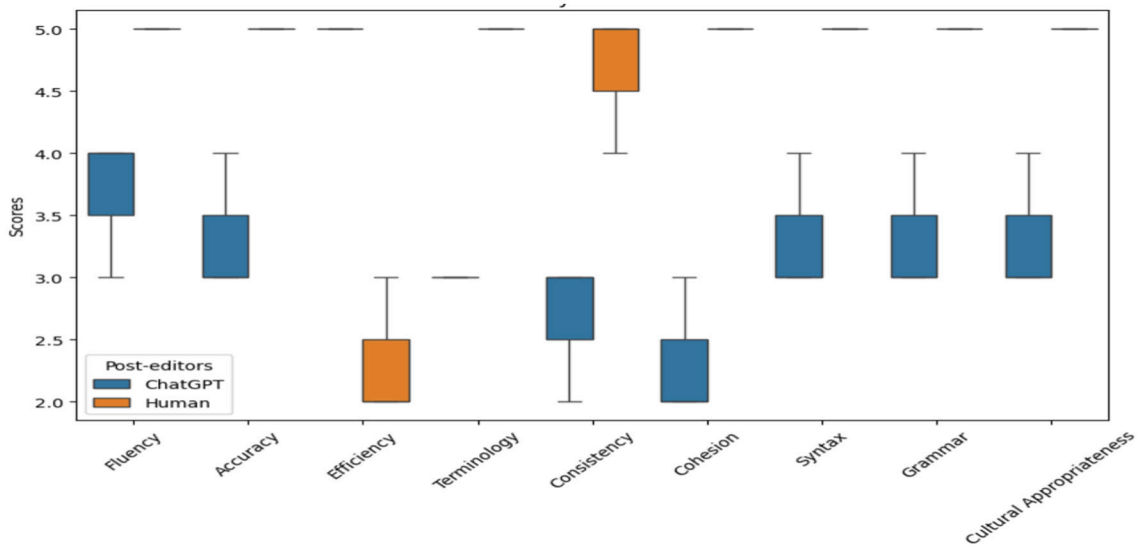

FIGURE 1
ChatGPT-4o and human post-editing across nine metrics. Human post-editing outperforms in all categories except efficiency, with higher medians and tighter interquartile ranges (IQRs) (orange boxes), indicating superior consistency in quality, accuracy, and fluency. ChatGPT-4o (blue boxes) shows lower ratings and wider IQRs, reflecting variability in handling nuanced language, terminology, and grammar. While ChatGPT-4o maintains fluency and coherence due to its conversational design, it struggles with technical terms and syntactic precision. Its strength lies in speed, making it useful for time-sensitive tasks. However, human expertise remains essential for high-quality translations requiring cultural and linguistic nuance.

cultural appropriateness, and fluency in translations, as seen from the higher median lines and smaller interquartile ranges (IQRs) in the orange boxes for human post-editing. The IQR indicates low variance and better overall performance. In contrast, ChatGPT-4o shows lower ratings across these aspects with larger IQRs in the blue boxes, suggesting more variability and lower overall performance than human performance. This reflects a common challenge in ChatGPT-4o's handling of nuanced and culturally specific elements and their idiomatic meaning. It shows some deficiency in language smoothness and syntax, such as agreement errors, word order, and grammatical mistakes related to articles used, as seen in the analysis section.

In addition, ChatGPT-4o shows significant issues in the use of correct and consistent terminological and technical terms and failed to effectively post-editing. It still appears fluent (Maintaining logical flow and coherence between sentences and paragraphs), precise, consistent in style and tone, and readable throughout the content due to ChatGPT-4o's conversational nature. Indeed, ChatGPT-4o has the potential for rapid processing and editing, making it a valuable tool for scenarios where speed is critical. While ChatGPT-4o excels in speed and efficiency, human post-editing remains crucial for achieving high-quality translations across these critical aspects.

The heat map in Figure 2 interprets the t-statistic and p-value values for each aspect when comparing ChatGPT-4o and human post-editing. The p-value gradient in the heatmap (represented in the bottom half of the heatmap) highlights statistical significance, with green indicating significant differences ($p < 0.05$). Most aspects are shaded green, confirming the reliability of the observed differences, except for fluency, which is shaded yellow. The t-statistic values are represented in the heatmap's top half, showing the direction and magnitude of differences in ratings. The t-statistics indicate that human post-editing generally outperforms ChatGPT-4o in most aspects, such as accuracy, terminology, consistency, cohesion, syntax, grammar, and cultural appropriateness, all showing significant negative values (ranging from −3.46 to −8) and corresponding p-values below 0.05, confirming that the differences are not only substantial but also statistically significant. However, regarding efficiency, ChatGPT-4o is rated significantly higher, with a positive t-statistic of 8.00 and a p-value of 0.015, indicating that it is more efficient than human post-editing. The only aspect where the difference is not statistically significant is fluency, with a t-statistic of −3.5 and a p-value of 0.074, suggesting that both methods perform similarly. Overall, the heatmap underscores ChatGPT-4o's strength in efficiency but highlights human post-editing's superiority in maintaining quality and accuracy across most aspects.

This study shows that, to some extent, ChatGPT-4o plays an influential role in improving the post-editing of machine-generated translations (MGT) in various domains attributed to its potential to generate fluent and natural translation reflecting relevant context and literature that is relatedly supporting the findings of Jiao et al. (2023) and Hendy et al. (2023). According to Peng et al. (2023), adapting ChatGPT-4o with optimized prompts and context improves its performance and makes it more suitable for specialized translation tasks. However, ChatGPT-4o's results may be similar to Google Translate or inaccurate without such optimization. Although ChatGPT-4o cannot provide completely accurate translations without human intervention, such integration would significantly reduce costs, time, and effort and provide considerable improvements and suggestions. Our analysis found that ChatGPT-4o can effectively contribute to post-edit generation and help identify translated content that may require further consideration or refinement. The results generated by ChatGPT-4o eliminate the need for skilled linguists to manually review

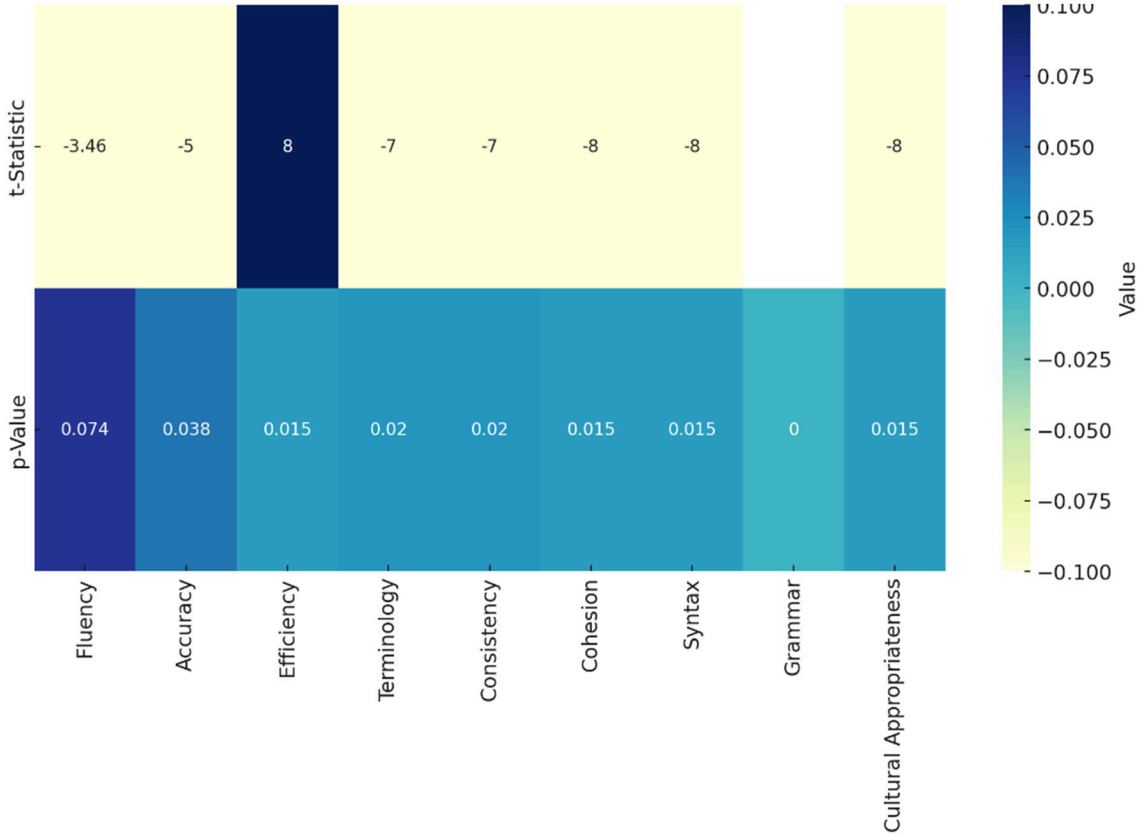

FIGURE 2
T-statistics and p-values comparing ChatGPT-4o and human post-editing. The *p*-value gradient (bottom half) shows statistical significance, with green ($p < 0.05$) indicating meaningful differences. Most aspects are green, except fluency (yellow). The top half shows *t*-statistics, revealing that human post-editing outperforms ChatGPT-4o in accuracy, terminology, consistency, cohesion, syntax, grammar, and cultural appropriateness ($t = -3.46$ to -8, $p < 0.05$). However, ChatGPT-4o excels in efficiency ($t = 8.00$, $p = 0.015$). Fluency shows no significant difference ($t = -3.5$, $p = 0.074$). The heatmap highlights ChatGPT-4o's efficiency advantage but confirms most aspects of human post-editing's superior quality.

TABLE 13 Inter-annotator agreement (IAA) scores.

Metrics	ChatGPT-4o post-edits	Human post-edits
Average pairwise Spearman's *rho*	0.85	0.99
Fless'Kappa (quadratic weights)	0.78	0.95

the text, catch errors, give appropriate feedback, and ensure cultural appropriateness (Khan, 2024; Yang et al., 2023). To assess to which extent the three evaluators agree in their rating and thus ensure their reliability, we calculated the Inter-Annotator Agreement (IAA) using Spearman's rank correlation coefficient for pairwise comparisons and Fless' Kappa with quadratic weighted for overall agreement as illustrated in Table 13. The evaluators exhibit a near-perfect agreement for human post-editing, with pairwise Spearman's rho value of 0.99 and Fless'Kappa value of 0.85. For ChatGPT4o editing, the evaluators' agreement with pairwise Spearman's rho value is 0.85, and the Fless'Kappa value is 0.78, which means there is a substantial agreement among the three human evaluators.

The values of IAA indicate a high level of reliability across the three evaluators (EV1, EV2, and EV3), stressing the robustness of our evaluation process of both human editors and ChatGPT4o as an editor.

Conclusion

This research provides valuable insights into ChatGPT-4o's potential to enhance the MGT post-editing service and its overall role in assisting human translators with post-editing tasks in various domains. This study evaluates the post-editing performance of ChatGPT-4o compared to human editing based on an evaluation by three human raters on multiple metrics. The results show that although human post-editing outperforms ChatGPT-4o in most evaluation metrics, the latter provides a fluent translation, which promises to improve quality, work efficiency, and translation workflows in various fields. Additionally, the study found that ChatGPT-4o's detailed guidance includes clear task instructions, contextual information, and a description of the desired results that will help improve ChatGPT-4o's functionality. Future research may explore ChatGPT versions' use within professional translation services, especially in enhancing post-editing workflows, addressing

the practical challenges, and identifying strategies to overcome these obstacles. Additionally, domain-specific fine-tuning of large-scale language models (LLMs) using specialized translation datasets needs exploration. Furthermore, creating and using diverse datasets that reflect a broader spectrum of Arabic dialects and text complexities to improve the generalizability and robustness of LLMs in translation tasks.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

Author contributions

JA: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. MM: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Albuhairy, M. M., and Algaraady, J. (2025). DeepSeek vs. ChatGPT: comparative efficacy in reasoning for adults' second language acquisition analysis. مجلة العلوم التربوية و الدراسات الإنسانية 44, 883–864.

Al-Garaady, J., and Mahyoob, M. (2023). ChatGPT's capabilities in spotting and analyzing writing errors experienced by EFL learners. *Arab. World Engl. J.* 3–17. doi: 10.24093/awej/call9.1

Allen, J. (2001). Postediting: an integrated part of a translation software program. *Lang. Int.* 13, 26–29.

Allen, J. (2003). "Post-editing. Benjamins Translation Library," in *Computers and translation: A Translator's Guide*, ed. H. Somers. Amsterdam: John Benjamin's Publishing Company, 297–317.

Castilho, S., O'Brien, S., Alves, F., and O'Brien, M. (2014). "Does post-editing increase usability? A study with Brazilian Portuguese as Target Language," in *Proceedings of the 17th Annual Conference of the European Association for Machine Translation*, 183–190.

Chen, P., Guo, Z., Haddow, B., and Heafield, K. (2023). Iterative translation refinement with large language models. *arXiv* [preprint] arXiv:2306.0385. doi: 10.48550/arXiv.2306.03856

Daems, J., Macken, L., and Vandepitte, S. (2013). "Quality as the sum of its parts: A two-step approach for the identification of translation problems and translation quality assessment for HT and MT+ PE," in *Proceedings of the 2nd Workshop on Post-Editing Technology and Practice*.

Green, S. J., Chuang, J., Heer, J., and Manning, C. D. (2014). "Predictive Translation Memory: A mixedinitiative system for human language translation", in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, HI USA: Association for Computing Machinery), 177–187.

Hendy, A., Abdelrehim, M., Sharaf, A., Raunak, V., Gabr, M., Matsushita, H., et al. (2023). How good are GPT models at machine translation? A comprehensive evaluation. *arXiv* [preprint] arXiv:2302.09210. doi: 10.48550/arXiv.2302.09210

Jiao, W., Wang, W., Huang, J., Wang, X., and Tu, Z. (2023). Is ChatGPT-4o a good translator? A preliminary study. *arXiv* [preprint] arXiv:2301.08745. doi: 10.48550/arXiv.2301.08745

Khan, F. (2024). Human-in-the-loop approaches to improving machine translation. *Acad. J. Sci. Technol.* 7, 1–8.

Koehn, P., Alabau, V., Carl, M., Casacuberta, F., García-Martínez, M., González-Rubio, J., et al. (2015). *CASMACAT: Final Public Report*. Available online at: http://www.casmacat.eu/uploads/Deliverables/final-public-report.pdf (accessed 12 December, 2018).

Koneru, S., Exel, M., Huck, M., and Niehues, J. (2023). "Contextual refinement of translations: large language models for sentence and document-level post-editing," in *arXiv* [preprint] arXiv:2310.1485. doi: 10.48550/arXiv.2310.14855

Langlais, P., and Lapalme, G. J. M. T. (2002). Trans type: development-evaluation cycles to boost translator's productivity. *Mach. Transl.* 17, 77–98. doi: 10.1023/B:COAT.0000010117.98933.a0

Laurian, A.-M. (1984). "Machine translation: what type of post-editing on what type of documents for what type of users," in *Proceedings of the 10th International Conference on Computational Linguistics* (Stanford: Association for Computational Linguistics), 236–238.

Lee, D., Ahn, J., Park, H., and Jo, J. (2021). *IntelliCAT: Intelligent Machine Translation Post-Editing with Quality Estimation and Translation Suggestion*, 11–19.

Mahyoob, M. (2020). Developing a simplified morphological analyzer for Arabic pronominal system. *Int. J. Nat. Lang. Comp.* 9, 9–19. doi: 10.2139/ssrn.3599719

Mahyoob, M., and Al-Garaady, J. (2018). Towards developing a morphological analyzer for Arabic noun forms. *Int. J. Linguist. Computat. Appl.* 5:7. doi: 10.30726/ijlca/v5.i3.2018.52012

Mahyoob, M., Al-Garaady, J., and Alblwi, A. (2023). A proposed framework for human-like language processing of ChatGPT in academic writing. *Int. J. Emerg. Technol. Learn.* 18:14.

Massardo, I., van der Meer, J., O'Brien, S., Hollowood, F., Aranberri, N., and Drescher, K. (2016). *MT PostEditing Guidelines*. Amsterdam: Translation Automation User society.

Peng, K., Ding, L., Zhong, Q., Shen, L., Liu, X., Zhang, M., et al. (2023). Towards making the most of ChatGPT-4o for machine translation. *arXiv* [preprint] arXiv:2303.13780. doi: 10.2139/ssrn.4390455

Qin, M. (2022). "Machine translation technology based on natural language processing. 2022," in *European Conference on Natural Language Processing and Information Retrieval (ECNLPIR),* 10–13.

Raj, A., Jindal, R., Singh, A. K., and Pal, A. (2023). "A study of recent advancements in deep learning for natural language processing," in *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)* (Sonbhadra: IEEE), 300–306.

Raunak, V., Sharaf, A., Wang, Y., Awadallah, H. H., and Menezes, A. (2023). Leveraging GPT-4 for automatic translation post-editing. *arXiv preprint* arXiv:2305.14878.

Screen, B. (2019). What effect does post-editing have on the translation product from an end-user's perspective? *J Special Transl.* 31, 133–157.

TAUS (2010). MT Post-editing Guidelines. Available online at: https://www.taus.net/academy/best-practices/postedit-bestpractices/machine-translation-post-editing-guidelines (accesed 20 May, 2016).

Turchi, M., Negri, M., Farajian, M., and Federico, M. (2017). Continuous learning from human post-edits for neural machine translation. *Prague Bullet. Mathem. Linguist.* 108, 233–244. doi: 10.1515/pralin-2017-0023

Underwood, N., Mesa-Lao, B., Martínez, M. G., Carl, M., Alabau, V., González-Rubio, J., et al. (2014). "Evaluating the effects of interactivity in a post-editing workbench," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC*'14) eds. N. Calzolari, K. Choukri, and T. Declerck. (Reykjavik: European Language Resources Association), 553–559.

van Egdom, G.-W., and Pluymaekers, M. (2019). Why go the extra mile? How different degrees of post-editing affect perceptions of texts, senders and products among end users. *J. Special. Transl.* 31, 158–176.

Vardaro, J., Schaeffer, M., and Hansen-Schirra, S. (2019). Translation quality and error recognition in professional neural machine translation post-editing. *Informatics* 6:41. doi: 10.3390/informatics6030041

Vieira, L. N. (2017). "From process to product: links between post-editing effort and post-edited quality", in *Translation in Transition: Between Cognition, Computing and Technology* eds. A. L. Jakobsen, and B. Mesa-Lao (Amsterdam: John Benjamins Publishing Company), 162–186. doi: 10.1075/btl.133.06vie

Vieira, L. N. (2019). "Post-editing of machine translation," in *The Routledge Handbook of Translation and Technology* (Routledge), 319–336.

Yang, X., Zhan, R., Wong, D. F., Wu, J., and Chao, L. S. (2023). Human-in-the-loop machine translation with large language model. *arXiv* [preprint] arXiv:2310.08908. doi: 10.48550/arXiv.2310.08908

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 19 June 2025
DOI 10.3389/frai.2025.1615800

OPEN ACCESS

EDITED BY
Shadi Abudalfa,
King Fahd University of Petroleum and Minerals, Saudi Arabia

REVIEWED BY
Hassina Aliane,
Research Center on Scientific and Technical Information, Algeria
Ashwag Alasmari,
King Khalid University, Saudi Arabia
Sultan Alrowili,
University of Delaware, United States

*CORRESPONDENCE
Ali Alkhathlan
✉ analkhathlan@kau.edu.sa

RECEIVED 21 April 2025
ACCEPTED 26 May 2025
PUBLISHED 19 June 2025

CITATION
Alkhathlan A, Alahmadi F, Kateb F and Al-Khalifa H (2025) Constructing and evaluating ArabicStanceX: a social media dataset for Arabic stance detection. *Front. Artif. Intell.* 8:1615800. doi: 10.3389/frai.2025.1615800

Constructing and evaluating ArabicStanceX: a social media dataset for Arabic stance detection

Ali Alkhathlan[1]*, Faris Alahmadi[1], Faris Kateb[2] and Hend Al-Khalifa[3]

[1]Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia, [2]Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia, [3]Information Technology Department, King Saudi University, Riyadh, Saudi Arabia

Arabic stance detection has attracted significant interest due to the growing importance of social media in shaping public opinion. However, the lack of comprehensive datasets has limited research progress in Arabic Natural Language Processing (NLP). To address this, we introduce ArabicStanceX, a novel and extensive Arabic stance detection dataset sourced from social media, comprising 14,477 tweets across 17 diverse topics. Utilizing the transformer-based MARBERTv2 model, we explore stance detection through Multi-Topic Single Model (MTSM) strategies, achieving a promising F1 score of 0.74 for detecting 'favor' and 'against' stances, and 0.67 overall. Our experiments highlight the model's capabilities and challenges, particularly in accurately classifying neutral stances and generalizing to unseen topics. Further investigations using zero-shot and few-shot learning demonstrate the model's adaptability to new contexts. This study significantly advances Arabic NLP, providing crucial resources and insights into stance detection methodologies and future research directions. The dataset is publicly available[1].

KEYWORDS

stance detection, Arabic language, opinion mining, social media analysis, Arabic NLP

1 Introduction

The digital era, marked by rapid technological advancements, constantly redefines our communication methods. New social media platforms emerge daily, promoting widespread connection and opinion sharing. Currently, over 58% of the global population uses social media, spending an average of 2–3 h online each day (Al Hendi, 2024).

A platform of significant interest to researchers is X.com (formerly Twitter), renowned for its ability to facilitate opinion expression. The diverse information within tweets provides valuable insights into public stance and behavior, fueling interest in "opinion mining" across fields such as Natural Language Processing (NLP) and social computing. The primary goal is to develop automated methods for measuring public opinion, supplementing traditional surveys.

Stance detection, a notable subfield of opinion mining, focuses on identifying whether an author's viewpoint in the text is supportive, opposing, or neutral toward a specific topic, such as an individual, legislation, or event. This task is crucial for applications like social media monitoring, opinion mining, and political analysis. For example, the tweet "Handguns should be banned in the US" illustrates a supportive stance on gun control.

1 https://github.com/AliAlkhathlan/ArabicStanceX and https://huggingface.co/datasets/Faris-ML/ArabicStanceX.

With the proliferation of online platforms for sharing opinions, NLP research in stance detection has grown substantially. A pivotal development was the release of a stance detection dataset by Mohammad et al. (2016). Recent advancements in NLP and deep learning, particularly the development of transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), have significantly enhanced stance detection capabilities. BERT's bidirectional fine-tuning approach allows it to understand the context of words within a sentence, making it highly effective for a wide range of NLP tasks.

Despite BERT's success in many languages, applying such models to Arabic text presents unique challenges due to the language's complex morphology, dialectal variations, and rich contextual semantics. Most stance detection research has focused on English due to the abundance of available datasets. However, other languages, like Arabic, have received less attention, with Arabic stance detection datasets being limited in terms of topic and diversity. This lack of comprehensive datasets represents a significant gap in NLP research.

This research aims to advance Arabic stance detection by introducing ArabicStanceX, a comprehensive and diverse dataset that can serve as a benchmark for a wide range of language models. To demonstrate its effectiveness, we evaluate it using MARBERTv2, a strong Arabic-specific baseline. It addresses the gap in available datasets by developing a comprehensive and diverse Arabic stance detection dataset from X.com tweets, called ArabicStanceX, focusing on Saudi Arabia due to its high X.com usage and active social media discussions. The number of X.com users in Saudi Arabia reached 5 million in 2012 and has since grown by 160%, reached ~13 million users by 2020 (Simsim, 2011). Addaitionally, recent legislation has sparked extensive discussions and debates among Saudis on social networks. While X.com is also widely used across other Arab countries, this study specifically focuses on Saudi Arabia due to both the platform's high penetration and the sociopolitical context that has triggered extensive public discourse in recent years. We acknowledge that this geographical focus may limit the generalizability of findings to other regions. However, the methodology and insights gained here lay the foundation for broader extensions to other Arabic-speaking communities.

This study introduces ArabicStanceX, an extensive dataset for Arabic stance detection comprising 14,477 instances across 17 topics, which will be publicly accessible to foster further research. It focuses on developing adaptable models for unseen topics using zero-shot and few-shot learning methodologies, evaluating various fine-tuning strategies with the MARBERTv2 model. The research investigates Single Topic Single Model (STSM) and Multi Topics Single Model (MTSM) approaches, enhancing MTSM with additional contextual information. Using F_{avg2} and F_{avg3} metrics, it assesses precision and recall for "favor" and "against" stances. Overall, the study makes significant contributions to Arabic NLP by providing a valuable dataset, exploring model adaptability, and evaluating effective fine-tuning and contextual strategies.

The rest of the paper is organized as follows: Section 2 reviews related work in stance detection, with a particular focus on previous datasets and methodologies. Section 3 details the methodology for developing the Arabic stance detection dataset, including data collection and annotation processes. Section 4 describes the experimental setup, including the BERT model, its hyperparameter tuning, and performance metrics. Section 5 presents the experimental results and their analysis. Finally, Section 6 concludes the paper and outlines promising directions for future research.

2 Related work and background

Stance detection research on social media platforms has gained significant traction in recent years. This research can be categorized into four main categories.

1. **Target-specific:** this category focuses on recognizing stances toward specific, predefined targets. For example, it identifies opinions related to particular issues like civil rights, where the stance is evaluated directly against a clearly defined subject.
2. **Multi-related targets:** in this approach, a single model is used to identify stances toward two or more interrelated subjects within the same text. For instance, the model might analyze the connection between civil rights and the death penalty, recognizing how opinions on one issue might influence or correlate with opinions on the other.
3. **Cross-target:** this category aims to develop classifiers that can transfer knowledge between various targets using a comprehensive dataset. The goal is to create models that are versatile and can apply learned stances from one target to different, previously unseen targets, thus enhancing the model's generalizability and adaptability.
4. **Target-independent:** this approach seeks to identify stances in comments related to news articles, focusing on tasks like confirming or denying the validity of the information or predicting whether different arguments support the same stance. This method does not rely on predefined targets but instead evaluates stances based on the context of the discussion.

These classifications help structure stance detection research, guiding the development of models and methods tailored to specific needs and applications in analyzing and understanding public opinions across various domains.

The field of stance detection received a significant boost with the launch of a shared task and the subsequent release of a publicly available dataset by Mohammad et al. (2016, 2017). This dataset, sourced primarily from X.com and focusing on predefined controversial topics like climate change and abortion, significantly increased research output compared to previous years (AlDayel and Magdy, 2021). Annotators on CrowdFlower categorized tweet-topic pairs into three stances: favor, against, or neutral.

Since then, additional stance detection datasets have emerged, catering to various domains. A substantial dataset of over 51,000 tweets focused on the financial domain was introduced in Conforti et al. (2020). The TW-BREXIT dataset, presented in Lai et al. (2020) contains 1,800 triplets of tweets related to the stance on leaving, remaining, or having no opinion on Brexit. Similarly, datasets addressing other controversial topics have been developed (Hosseinia et al., 2020; Grimminger and Klinger, 2021; Li et al., 2021; Gautam et al., 2020; Thakur and Kumar, 2021).

The investigation of stance detection has also expanded to include target-independent approaches, garnering considerable research interest. For instance, Gorrell et al. (2019) presented

RumourEval, a claim-based dataset designed for stance classification within the context of rumors. This dataset covers a broad spectrum of events and categorizes tweets into four distinct stances: support, deny, query, or comment. Similarly, Hanselowski et al. (2018) proposed another dataset aimed at assessing stances toward various news headlines. These efforts are just a few examples, with additional datasets emerging in this vein by Ferreira and Vlachos (2016); Bar-Haim et al. (2017). Research has also explored cross-target stance detection (Allaway and McKeown, 2020; Vamvas and Sennrich, 2020; Kaur et al., 2016) and multi-target stance detection (Sobhani et al., 2017). Furthermore, efforts have been made to extend stance detection research to non-English languages, including Italian (Cignarella et al., 2020) and Spanish/Catalan (Taulé et al., 2017).

While stance detection datasets abound for English, Arabic resources remain scarce. A notable contribution is the fact-checking corpus by Baly et al. (2018), which links 402 Arabic claims to retrieved documents using a four-class stance scheme (agree, disagree, discuss, unrelated), annotated via crowdsourcing. While the dataset includes rationale spans for some labels, it is oriented toward long-form claim-document verification rather than general-purpose stance modeling. The Arabic News Stance corpus by Khouja (2020) comprises 3,786 claims, annotated through a multi-stage crowdsourcing process. It employs a three-class scheme (agree, contradict, other), merging "discuss" and "unrelated" into a single label to reduce ambiguity. While the dataset emphasizes real news headlines and achieves high inter-annotator agreement, it exhibits class imbalance and possible paraphrasing-induced variability.

AraStance (Alhindi et al., 2021) offers 4,063 claim–article pairs across multiple domains and Arab countries, labeled by graduate annotators using a four-class scheme (agree, disagree, discuss, unrelated). While its broad topical scope and refined annotation process enhance reliability, the dataset remains rooted in formal news sources and exhibits class imbalance. Expanding the options for Arabic stance detection, Alturayeif et al. (2022) introduced MAWQIF, a multi-dimensional dataset containing 4,121 Arabic tweets annotated for stance, sentiment, and sarcasm via Appen crowdsourcing. The stance labels follow a target-specific three-class scheme (favor, against, none), applied across three controversial topics. Although MAWQIF supports multi-task learning and includes dialectal variation, its coverage is limited to predefined targets, and it exhibits class imbalance due to low representation of neutral stances. Additionally, Jaziriyan et al. (2021) introduced EXaASC, a target-based stance dataset containing 9,566 Arabic tweet–reply pairs annotated by trained native speakers using a three-class scheme. With over 180 unique targets, it offers broad generalization potential, though its reply-based structure introduces conversational bias and a high proportion of none labels.

Table 1 summarizes these datasets, providing details on their name, language, stance type, text source, and size.

Research in stance detection has advanced significantly, but several notable gaps persist. Firstly, there is a scarcity of data in non-English languages, with most research focusing on English datasets. While efforts like AraStance and MAWGIF have contributed to Arabic resources, they remain more minor and less diverse compared to their English counterparts. Secondly, existing models often struggle with generalizability, especially when faced with unseen topics or targets. Cross-target stance detection methods aimed at enhancing adaptability to new targets with limited data are still in development. Additionally, current models primarily focus on explicit language, overlooking the role of context and implicit cues in sentence analysis. Elements like sarcasm and humor can be challenging for these models to interpret accurately.

To bridge these gaps, this study prioritizes creating more prominent and varied datasets in Arabic and other languages. Techniques like few-shot learning and domain adaptation have the potential to enhance model generalizability. Furthermore, incorporating contextual cues and sentence analysis can better capture the subtleties of human language. Through these efforts, stance detection can evolve into a more powerful tool for deciphering public opinion across diverse linguistic and cultural landscapes.

3 Methodology for ArabicStanceX dataset development

In this section, we detail the methodologies utilized in constructing the ArabicStanceX dataset. Our primary aim is to create a comprehensive, multi-topic dataset in Arabic that sets itself apart from previous datasets by offering extensive coverage and suitability for addressing novel targets, thus expanding its potential applications. Our research focused on data spanning from 2015 to 2021 in Saudi Arabia, a period marked by significant controversies. The dataset was sourced from X.com, making it currently the most exhaustive Arabic stance dataset available. The methodology for developing the Arabic stance detection dataset is illustrated in Figure 1 and described in the following subsections.

3.1 Data collection and filtering

Our initial step was to create a collection of pre-defined, controversial topics that would elicit strong opinions. We achieved this by first extracting all hashtags from X.com within Saudi Arabia between 2015 and 2021. We then analyzed these hashtags to identify potential topics. Specifically, we manually reviewed the most frequently occurring hashtags and selected those that were associated with real-world events, public policies, or debates that sparked polarized public engagement. Hashtags were grouped into candidate topics if they reflected a clearly defined issue with both supportive and opposing discourse. Once a topic was identified, we used its relevant keywords to find all related hashtags, ensuring a broad spectrum of areas like sports, economy, education, health, religion, and culture (details in Table 1).

To capture a diverse range of viewpoints, we collected hashtags representing both supportive and opposing stances for each topic. For instance, on the topic of women driving, we included hashtags like "#WomenShouldDrive" and "#WomenShouldNotBeDriving." This approach ensured we captured a spectrum of opinions, from agreement to disagreement.

TABLE 1 Summary of stance detection datasets by name, language, source, and size.

Name	Language	Stance type	Text source	Size
SemEval2016-Task 6 (Mohammad et al., 2016, 2017)	English	Target specific	X.com	4,163 tweets
WT-WT (Conforti et al., 2020)	English	Target specific	X.com	51K
TW-BREXIT (Lai et al., 2020)	English	Target specific	X.com	1,800 triplets of tweets
Procon20 (Hosseinia et al., 2020)	English	Target specific	procon.org	6,094 of question and opinion
Hateful/offensive speech (Grimminger and Klinger, 2021)	English	Target specific	X.com	3K tweets
P-stance (Allaway and McKeown, 2020)	English	Target specific	X.com	21,574 tweets
MeTooMA (Gautam et al., 2020)	English	Target specific	X.com	9,973 tweets
RumourEval (Gorrell et al., 2019)	English	Target independent	X.com and Reddit	8,574 posts
FNC-1 (Hanselowski et al., 2018)	English	Target independent	News websites	75,385 instances and 2,587 news headlines
Emergent (Ferreira and Vlachos, 2016)	English	Target independent	Different websites	300 claims and 2,595 articles
IBM debater (Bar-Haim et al., 2017)	English	Target independent	Wikipedia	2,394 claims
Vast (Allaway and McKeown, 2020)	English	Cross target	News website	23,525 comments
X-stance (Vamvas and Sennrich, 2020)	Italian German French	Cross target	Smartvote.org	65 K
Multi-target SD (Sobhani et al., 2017)	English	Multi target	X.com	4,455 tweets
SardiStance (Cignarella et al., 2020)	Italian	Target Specific	X.com	3,242 tweets
IberEval (Taulé et al., 2017)	Spanish and Catalan	Target specific	X.com	11 K
Arabic fact checking (Baly et al., 2018)	Arabic	Target independent	Verify and Reuters	402 claims and 3,042 documents
Arabic news stance (Khouja, 2020)	Arabic	Target independent	News websites	3,786 pairs (claim, evidence)
AraStance (Alhindi et al., 2021)	Arabic	Target independent	Fact-checking websites	4,063 pairs of claim and article
MAWGIF (Alturayeif et al., 2022)	Arabic	Target specific	X.com	4,121 tweets
EXaASC (Jaziriyan et al., 2021)	Arabic	Cross-target	X.com	9,566 samples, and 180 targets

After collecting the data, we organized it into distinct domains, each containing specific topics with their associated hashtags and tweets. We then performed several preprocessing steps:

1. **Language filtering:** we filtered out all non-Arabic tweets, keeping only Arabic content.
2. **Noise removal:** we removed retweets, user mentions, URLs, and duplicate tweets. To identify subtle duplicates, we employed SentenceTransformer "paraphrase-xlm-r-multilingual-v1" by Reimers and Gurevych (2019) to measure tweet similarity. Tweets with a cosine similarity exceeding 0.95 were discarded.
3. **Advertisement removal:** analysis of a random sample of 1,000 tweets revealed that tweets with four or more hashtags were predominantly advertisements. Consequently, we eliminated all such tweets from the dataset.

Table 2 provides a list of the domains and their associated target topics.

3.2 Data annotations

To ensure the accuracy of our stance labels, we partnered with Wosom, a Saudi company staffed with native Arabic speakers (Wosom, 2024). Wosom took on the responsibility of both conducting the annotations and upholding high-quality standards throughout the process.

Before embarking on the main annotation task, we initiated a pilot test using a smaller subset of the data. The purpose of this pilot test was to confirm the clarity of our annotation guidelines and validate the functionality of the annotation tools. We conducted the pilot test through multiple iterations, reviewing a random sample of 50 tweets from various topics after each iteration to identify and address any potential issues.

Three native Saudi speakers were meticulously selected based on their language proficiency, attention to detail, and relevant domain expertise to annotate each tweet. Subsequently, these annotators underwent rigorous training on the annotation guidelines and the Wosom annotation platform. They were provided with clear instructions and relevant examples to ensure the accuracy of their annotations. Throughout the annotation process, continuous feedback from reviewers and validators was incorporated to maintain high-quality standards. Each of the 14,477 tweets was independently annotated by all three annotators to ensure consistent labeling and enable majority agreement.

In instances of disagreement regarding the classification of a tweet, an adjudication method was implemented. This involved applying established criteria or engaging in group discussions facilitated by a designated team member to reach a consensus.

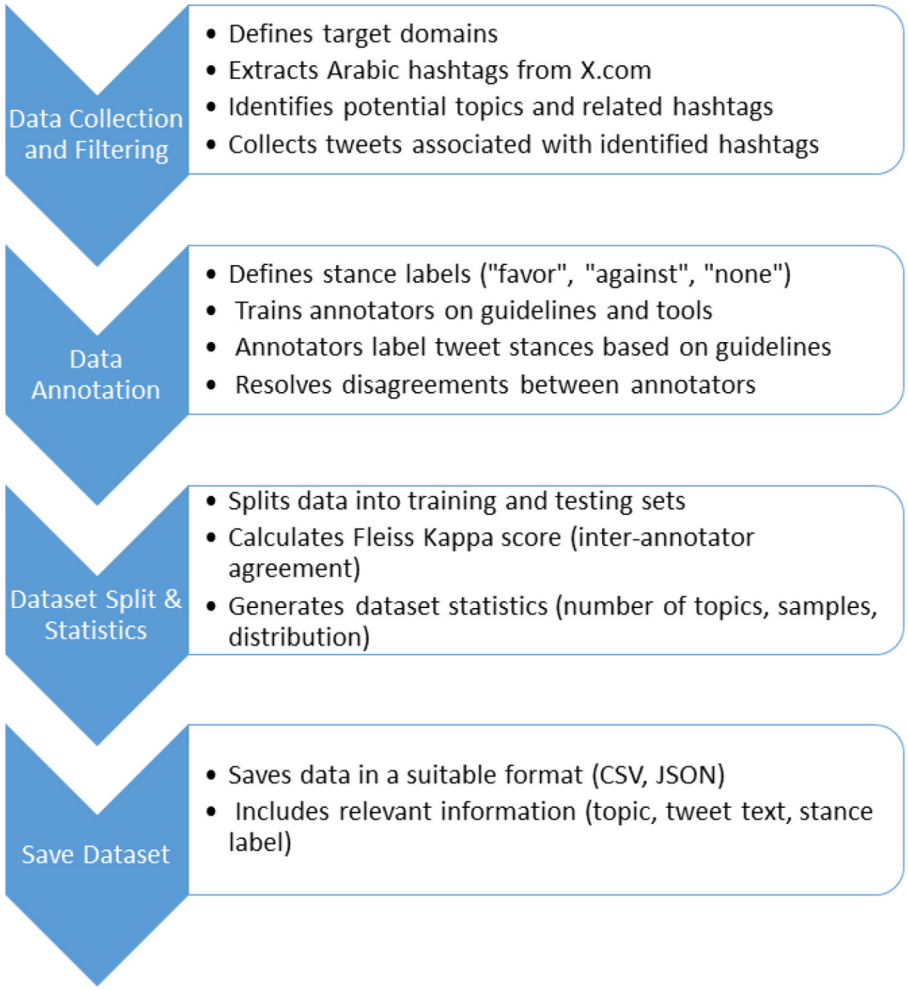

FIGURE 1
Methodology—ArabicStanceX dataset creation.

The annotators categorized tweets related to each topic into three distinct categories: "favor," "against," or "none." Tweets expressing explicit or implicit support for the topic were labeled as "favor," while those opposing the topic in either direct or indirect ways were labeled as "against." Tweets that did not express a stance or were unrelated to the topic, such as advertisements, were categorized as "none."

3.3 Dataset statistics

The ArabicStanceX dataset comprises 17 distinct topics with a total of 14,477 samples. To gauge the agreement between annotators, we computed an average Fleiss Kappa score of 0.54 across all topics. Subsequently, we partitioned the dataset into training and testing sets, utilizing an 80:20 split for model development and evaluation. Detailed statistics for individual topics within both sets are presented in Table 3.

Figure 2 illustrates the distribution of topics within the dataset, with a predominant focus on Religion/Culture (31.2%), followed by Education (19.1%), Economy (18.7%), Other (12.9%), and Health (12.9%). Sports constitute the most minor portion at 5.04%.

Further granularity is provided in Figure 3, which delineates the distribution of training and testing samples across these domains. This meticulously organized structure underscores the dataset's diversity and its coverage of a wide array of topics. Such diversity lays a robust groundwork for conducting thorough analyses and developing resilient Arabic stance detection models. The structured approach facilitates nuanced research and model training, thereby contributing to advancements in Arabic computational linguistics.

4 Experimental setup

In evaluating the efficacy of the ArabicStanceX dataset, we harnessed the power of the BERT (Bidirectional Encoder Representations from Transformers) architecture across different contexts. This section provides insights into BERT and the particular models we utilized for assessment. Additionally, we delve into the experimental configuration, encompassing

TABLE 2 Details of the specific domains and their related topics.

Domain	Topic	Topic description
Economy	Aramco Share Selling	Aramco made available a part of their total company shares, amounting to 1.5%, for trading among the general public.
	Al-Qiddiya Project	Al-Qiddiya is a Saudi sport, cultural, and entertainment project which will be located in the city of Al-Qiddiya, which serves as a high-quality entertainment and social destination.
	Neom City	The Kingdom of Saudi Arabia has planned to construct a novel urban district, Neom, in the northwestern Tabuk Province.
Education	Teaching Chinese Language at School	The Saudi Ministry of Education has announced to include Chinese language in the curriculum of Saudi public schools.
	Improve School Curriculum	The Saudi Ministry of Education unveiled a new educational system and curriculum that comprises new subjects and a reduction in the number of classes for religious studies.
	Online Learning	Transitioning from conventional to online teaching during COVID-19
Health	COVID-19 Vaccine	The Saudi authorities are mandating that Saudi citizens receive the COVID-19 vaccine.
	Vaccine Booster Dose	The Saudi authorities are mandating that Saudi citizens receive the COVID-19 booster dose.
Sports	Prince Abdulaziz bin Turki Head of Sports Minister	Appointing Prince Abdulaziz bin Turki as a minister of sports.
	Prince Faisal bin Turki as Resignation from a Saudi club	Prince Faisal bin Turki as resignation from Al-nasser Saudi club.
Religion/ Cultural	Sex Education	Implementing a sex education curriculum in Saudi public school.
	Coexistence with Religions	The peaceful coexistence and dialogue among religions.
	Women driving	Allowing women to drive in Saudi Arabia.
	Mosques Speakers	Limiting the utilization of mosque loudspeakers exclusively for the Adhan (the call to prayer) while retaining their use within the mosque premises during prayer times.
	Polygamous marriage	Deciding whether to endorse the concept of simultaneous multiple spouses.
Other	Domestic tourism	Supporting domestic tourism in the Kingdom of Saudi Arabia
	Military conscription	The mandatory enlistment of Saudi citizens in the armed forces

hyperparameter adjustments, and elucidate the performance metrics employed to measure the effectiveness of the models.

4.1 Model selection

This research leverages the power of Bidirectional Encoder Representations from Transformers (BERT) as the cornerstone of the ArabicStanceX dataset model. Developed by Google AI, BERT stands out for its exceptional ability to grasp the intricate relationships between words within a sentence (Devlin et al., 2018). Unlike traditional models that process text word by word, BERT employs a bidirectional approach. It analyzes both the preceding and following words, enabling it to capture the subtle nuances of language with remarkable precision. This bidirectional processing allows BERT to unlock the more profound meaning inherent in the text. By pre-training on massive amounts of text data, BERT learns to encode rich contextual information. This empowers it to excel in various Natural Language Processing (NLP) tasks, including sentiment analysis, text classification, and question answering.

In the realm of stance detection, where understanding an author's sentiment toward a topic is crucial, BERT's bidirectional processing proves invaluable. It delves into the full context of an Arabic sentence to discern whether the author's stance is supportive, opposing, or neutral regarding the embedded topic. However, to harness BERT's full potential for Arabic stance detection, fine-tuning is essential. This process involves adjusting BERT's internal parameters specifically for this task. Essentially, we train BERT to recognize the subtle ways in which stance is expressed within Arabic text. Through fine-tuning, BERT becomes adept at navigating the nuances of the Arabic language, offering valuable insights into public opinion and sentence across diverse topics and discussions.

We investigate different approaches for fine-tuning BERT during this phase, as outlined below:

1. **Single Topic Single Model (STSM):** in the STSM strategy, we employ a single input BERT structure. Initially, our focus was on fine-tuning a dedicated BERT-based model for each specific topic. This involved adjusting the weights of the pre-trained model to understand better the overall context and unique characteristics of each topic. The objective was to develop specialized models tailored to individual subject areas. However, we ultimately reconsidered this approach due to its consistent failure to capture the "None" stance across various topics effectively. This limitation revealed challenges in generalizing the models and accurately representing less common classes within single-topic analysis.
2. **Multi Topics Single Model (MTSM):** in the MTSM approach, we simultaneously fine-tune a single BERT-based model across all topics. This method allows the model to learn from a diverse range of subject matters in a unified manner, potentially improving its ability to discern commonalities and differences

TABLE 3 Data statistics for each label across all topics, segmented into the training and testing sets.

Domain	Topics	# Training samples (80%)				# Testing samples (20%)				Total samples
		Favor	Against	None	Total	Favor	Against	None	Total	
Education	Teaching Chinese language at school	336	297	65	698	85	75	17	177	875
	Improve School Curriculum	308	390	87	785	77	98	22	197	982
	Online Learning	297	326	111	734	75	82	28	185	919
Health	COVID-19 Vaccine	330	361	46	737	83	91	12	186	923
	COVID-19 Vaccine Booster Dose	280	372	105	757	70	93	27	190	947
Economy	Aramco Share Selling	297	293	132	722	75	74	34	183	905
	Al-Qiddiya Project	500	128	80	708	125	32	21	178	886
	Neom City	406	193	133	732	102	49	34	185	917
Other	Domestic Tourism	340	183	216	739	85	46	54	185	924
	Military Conscription	328	324	106	758	82	81	27	190	948
Sport	Prince Abdulaziz bin Turki Head of Sports Minister	63	72	100	235	16	18	60	94	329
	Prince Faisal bin Turki's Resignation from a Saudi club	100	61	123	284	70	16	31	117	401
Religion/ Culture	Women Driving	372	268	116	756	93	68	30	191	947
	Mosques Speakers	140	428	106	674	35	107	27	169	843
	Polygamous marriage	306	252	112	670	77	64	28	169	839
	Sex education	324	336	113	773	81	84	29	194	967
	Coexistence with religions	253	168	317	738	64	43	80	187	925
	Total	4980	4452	2068	11500	1295	1121	561	2977	14477

among topics. By fine-tuning the model on a broader dataset, we aim to enhance its generalization capabilities and its proficiency in handling multiple topics within a single framework. MTSM involves fine-tuning a combined dataset with variations in input data structure:

- **MTSM-None:** this model utilizes a single input sequence BERT architecture, fine-tuning the language model based solely on the tweet content without additional contextual information. The aim is to evaluate the model's stance inference capability from tweet text alone.
- **MTSM-Keywords:** employing a two-input-sequence BERT architecture, this method incorporates topic-specific keywords along with the tweets during fine-tuning. Including keywords aims to enhance the model's sensitivity to topic-specific nuances.
- **MTSM-Topic Description:** to ensure the model adequately captures topic-related nuances, we explore two strategies for providing it with sufficient topic description. The first strategy involves manually crafting a template-based description for each topic, guiding the content of the descriptions. The second strategy leverages GPT-4-ChatGPT to automatically generate relevant descriptions for each topic, potentially increasing scalability. An example of MTSM-Topic Description for teaching Chinese language in Saudi schools is provided in Figure 4.

4.2 Experimental design

This section elucidates the specific variant of the BERT model employed in our study, the process of hyperparameter tuning, and the performance metrics utilized for evaluation.

4.2.1 BERT model used

In this study, we employed the MARBERTv2 model, renowned for its exceptional performance in handling various Arabic dialectal tasks (Elmadany et al., 2022). The selection of MARBERTv2 was motivated by its state-of-the-art capabilities in comprehending and processing the intricacies of Arabic dialects, rendering it particularly well-suited for our stance detection task across a wide array of topics sourced from social media data. MARBERTv2 was fine-tuned on our dataset, as outlined in the Model section, utilizing the Multi Topics Single Model (MTSM) approach simultaneously across all topics. Additionally, we experimented with both single and two-input

BERT architectures. In all our methodologies, we utilized the BERT [CLS] token as the text representation embedding of the input text.

4.2.2 Hyperparameters tuning

In optimizing the hyperparameters for the MARBERTv2 model, our strategy aimed to fine-tune the settings to improve both fine-tuning efficiency and model performance. We employed the AdamW optimizer (Kingma and Ba, 2014), renowned for its effectiveness in handling sparse gradients on noisy problems. Our experiments utilized a constant learning rate of 2e-5, supplemented by beta coefficients of 0.9 and 0.999, and an epsilon value of 1e-8 to ensure robust convergence. To prevent overfitting, the model underwent a weight decay of 0.001 and employed a dropout rate of 0.1. The fine-tuning spanned 25 epochs with a batch size of 32. Input sequences were restricted to 128 tokens for single inputs and extended to 512 for composite inputs involving topics, balancing computational resources with comprehensive contextual understanding.

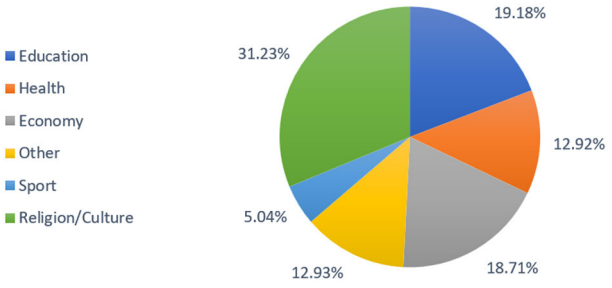

FIGURE 2
Distribution of samples across dataset domains.

4.2.3 Evaluation metrics

Our evaluation of the baseline models centers on two specialized metrics: F_{avg2} and F_{avg3} scores. The F_{avg2} score represents a macro-average F1 score tailored for the "favor" and "against" stance labels, deliberately excluding the "none" class due to its minimal presence in our dataset. The F_{avg2} score is computed using Equation 1.

$$F_{avg2} = \frac{F_{favor} + F_{against}}{2} \tag{1}$$

Here, F_{favor} and $F_{against}$ represent the F1 scores for the "favor" and "against" classes, respectively. These scores are derived from the precision and recall of each class as per Equations 2–3.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

We opted for the F_{avg2} metric to ensure alignment with other stance detection studies that report their findings using the same metric (Mohammad et al., 2016).

In addition to F_{avg2}, we present results using the F_{avg3} metric, which accounts for all stance labels, including "none". The F_{avg3} score represents an average of the F1 scores for all three stances and is calculated as per Equation 4.

$$F_{avg3} = \frac{F_{favor} + F_{against} + F_{none}}{3} \tag{4}$$

By reporting both F_{avg2} and F_{avg3} scores, our evaluation provides a comprehensive reflection of the model's performance in stance detection, encompassing both specific and overall detection capabilities.

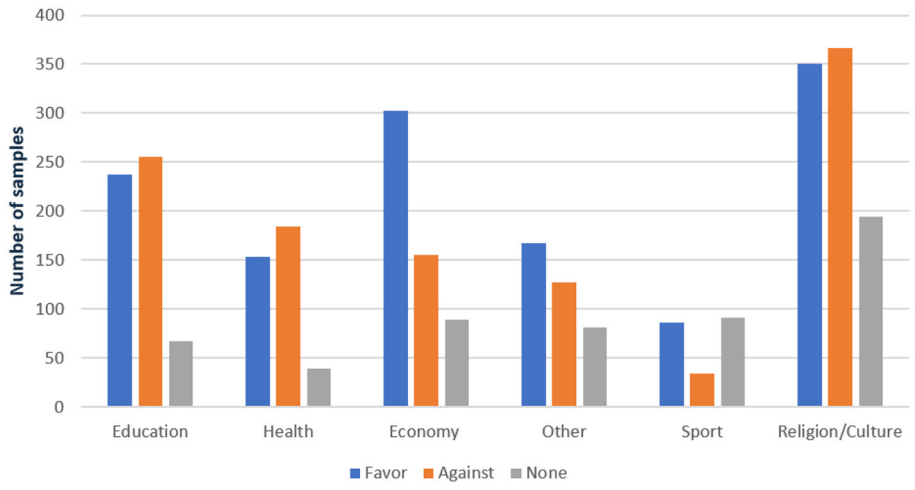

FIGURE 3
Distribution of class labels for training and testing sets across domains in the dataset.

تعليم اللغة الصينية في المدارس السعودية. وقد بدأ تدريس اللغة الصينية في الفصل الثاني من العام الدراسي 2020م. يأتي هذا القرار بعد اتفاق السعودية والصين، في فبراير 2019 على وضع خطة لإدراج اللغة الصينية في جميع مراحل التعليم العام والجامعي في المملكة، في خطوة تدعم سعي البلدين لتعزيز علاقاتهما على كافة المستويات. تأييد هذا القرار هو الموافقة ودعم تعليم اللغة الصينية في المدارس. ولكن رفض القرار هو عدم الموافقة على تعليم اللغة الصينية في المدارس.

English Translation

Chinese language instruction in Saudi schools began in the second semester of the academic year 2020. This decision follows an agreement between Saudi Arabia and China in February 2019 to develop a plan to incorporate the Chinese language into all stages of public and university education in the Kingdom, a step that supports the two countries' efforts to enhance their relations at all levels. Supporting this decision means agreeing with and endorsing the teaching of the Chinese language in schools. However, opposing the decision means disagreeing with the teaching of the Chinese language in schools.

(a)

إدراج اللغة الصينية في المدارس السعودية خطوة استراتيجية تعكس التوجه العالمي نحو تعزيز العلاقات مع الصين. تماشياً مع رؤية السعودية 2030، تهدف هذه الخطوة إلى إثراء المناهج الدراسية وفتح آفاق جديدة للطلاب في أكبر الاقتصادات العالمية. بدأت العملية في 2020 إثر اتفاق بين المملكة والصين لتدريس اللغة على مختلف المستويات التعليمية. تسهم الخطوة في تقوية العلاقات الثقافية والاقتصادية بين البلدين، وتمكين الشباب السعودي من مهارات متعددة. رغم الحماس الكبير لهذا التغيير، هناك آراء معارضة تشير إلى تحديات تتعلق بتعقيد اللغة وموارد التعليم.

English Translation

Incorporating Chinese into Saudi schools is a strategic step that reflects the global trend towards strengthening relations with China. In line with Saudi Vision 2030, this move aims to enrich the curriculum and open new horizons for students in one of the world's largest economies. The process began in 2020 following an agreement between the Kingdom and China to teach the language across various educational levels. This step contributes to strengthening the cultural and economic ties between the two nations and empowers Saudi youth with diverse skills. Despite the significant enthusiasm for this change, there are opposing views that point to challenges related to the complexity of the language and educational resources.

(b)

FIGURE 4
Example of manually crafted and ChatGPT generation of topic description for the topic of teaching Chinese language in Saudi schools. **(a)** Manually crafted topic description. **(b)** opic description generation by ChatGPT-GPT4.

5 Experiments and result analysis

We assessed the efficacy of the ArabicStanceX dataset, MARBERTv2, an Arabic Language Model, for stance detection across a range of topics. Our evaluations encompassed various fine-tuning approaches within the MTSM framework, including scenarios involving few-shot learning. Performance of different methods was gauged based on the ArabicStanceX dataset using performance metrics outlined in Section 4.2.3.

5.1 Performance analysis of MTSM model

We performed a series of experiments using the MTSM model with the ArabicStanceX dataset. The results are showcased in Table 4 employing the MTSM-None approach. In this experimental setup, the model fine-tunes a BERT-based language model solely on tweets without supplementary context, leading to notable performance variations across different topics. For example, the model achieves high F1 scores for "favor" and "against" classes in education-related topics like "Teaching Chinese Language at School." However, scores are notably lower for topics involving specific individuals, such as "Prince Abdulaziz bin Turki, Head of Sports Minister," suggesting challenges in stance detection when the input lacks contextual cues. The average F1 scores indicate that while the model performs adequately in some areas, it struggles in contexts requiring a deeper understanding of sentence, as evidenced by lower scores in complex social topics.

Table 4 shows the performance of the MTSM-None approach, which uses BERT to classify stances based solely on the tweet content for various topics in the dataset. The table includes F1 scores for three categories: "favor," "against," and "none." The F1 score is a metric that balances precision (accuracy of identifications) and recall (completeness of identifying positive cases). The obtained results are explained below.

1. Overall performance: the average F1 score across all topics considering both "favor" and "against" stances (F_{avg2}) is 0.74, with an average considering all three stances (F_{avg3}) being 0.66. This indicates that the model performs moderately well in stance detection using only tweet content.
2. Topic-wise performance: the performance varies depending on the topic. Some topics like "Teaching Chinese Language at

TABLE 4 F1 scores for "favor," "against," and "none" stances using MTSM-None).

Topic	F_{favor}	$F_{against}$	F_{none}	F_{avg2}	F_{avg3}
Teaching Chinese Language at School	0.90	0.91	0.46	0.90	0.75
Improve School Curriculum	0.91	0.887	0.40	0.89	0.73
Online learning	0.88	0.88	0.67	0.88	0.81
COVID-19 Vaccine	0.80	0.80	0.31	0.80	0.64
COVID-19 Vaccine Booster Dose	0.82	0.77	0.54	0.79	0.71
Aramco Share Selling	0.84	0.89	0.66	0.87	0.80
Al-Qiddiya Project	0.89	0.63	0.41	0.76	0.64
Neom City	0.90	0.79	0.65	0.84	0.78
Domestic Tourism	0.74	0.67	0.52	0.70	0.64
Sex Education	0.75	0.75	0.54	0.75	0.68
Coexistence with Religions	0.60	0.51	0.66	0.56	0.59
Military Conscription	0.76	0.77	0.64	0.77	0.73
Prince Abdulaziz bin Turki Head of Sports Minister	0.40	0.51	0.63	0.45	0.51
Prince Faisal bin Turki as Resignation from a Saudi club	0.48	0.36	0.45	0.42	0.43
Women_Driving	0.79	0.69	0.45	0.74	0.65
Mosques Speakers	0.57	0.76	0.24	0.67	0.53
Polygamous Marriage	0.83	0.83	0.49	0.83	0.71
AVERAGE OVER Avg2 & Avg3				0.74	0.66

TABLE 5 F1 scores for "favor," "against," and "none" stances using MTSM-Keywords.

Topic keywords	With topic keywords				
	F_{favor}	$F_{against}$	F_{none}	F_{avg2}	F_{avg3}
Teaching Chinese Language at School	0.9	0.91	0.62	0.9	0.81
Improve School Curriculum	0.79	0.83	0.43	0.81	0.68
Online Learning	0.92	0.91	0.73	0.91	0.85
COVID-19 Vaccine	0.83	0.8	0.29	0.82	0.64
COVID-19 Vaccine Booster Dose	0.82	0.84	0.59	0.83	0.75
Aramco Share Selling	0.81	0.88	0.64	0.85	0.78
Al-Qiddiya Project	0.88	0.67	0.5	0.78	0.68
Neom City	0.89	0.75	0.66	0.82	0.77
Domestic Tourism	0.64	0.57	0.49	0.61	0.57
Sex Education	0.72	0.8	0.56	0.76	0.69
Coexistence with Religions	0.44	0.44	0.63	0.44	0.5
Military Conscription	0.72	0.7	0.56	0.71	0.66
Prince Abdulaziz bin Turki Head of Sports Minister	0.27	0.43	0.68	0.35	0.46
Prince Faisal bin Turki as Resignation from a Saudi club	0.43	0.44	0.44	0.44	0.44
Women_Driving	0.77	0.68	0.55	0.72	0.67
Mosques Speakers	0.55	0.78	0.33	0.67	0.56
Polygamous Marriage	0.84	0.84	0.59	0.84	0.76
AVERAGE OVER Avg2 & Avg3				0.72	0.66

School" and "Aramco Share Selling" achieved high F1 scores for both "favor" and "against" stances (above 0.9 for F_{avg2}). This suggests the model can effectively classify tweets expressing explicit opinions on these topics.

3. Neutral stance ("none") classification: the model struggles with identifying neutral stances ("none") across most topics. This is evident from the consistently lower F1 scores for "none" compared to "favor" and "against." Topics like "Coexistence with Religions" and "Mosques Speakers" show particularly low scores for "none," indicating difficulty in distinguishing neutral tweets from those expressing an opinion on these sensitive subjects.

Overall, the results suggest that the MTSM-None approach achieves reasonable performance in stance detection for some topics with explicit opinions expressed in the tweets. However, the model has limitations in identifying neutral stances, especially for sensitive or complex topics. This highlights the potential need for incorporating additional information beyond just tweet content, such as topic descriptions or keywords, to improve the model's ability to handle diverse stances and topics.

Table 5 shows the performance of the MTSM-Keywords fine-tuning approach for stance detection on various Arabic topics. Each row represents a specific topic identified by its keywords. The columns "F_{favor}," "$F_{against}$," and "F_{none}" present the F1 scores, a metric used to evaluate model performance, for tweets classified as "favor," "against," and "none" stances on that topic, respectively. The "F_{avg2}" and "F_{avg3}" columns represent the average F1 scores across two different evaluation methods (potentially macro and micro averaging). Looking at the average scores at the bottom of the table (AVERAGE OVER Avg2 & Avg3), we see that the model performs moderately well overall, with an average F1 score of 0.72 for identifying tweets expressing a stance ("favor" or "against") and 0.66 for classifying tweets with a neutral stance ("none"). However, the performance varies across topics. Some topics, like "Online Learning" and "Aramco Share Selling," achieved high F1 scores for all stances, indicating the model's ability to classify tweets related to these topics accurately. Conversely, topics like "Coexistence with Religions" and "Prince Faisal bin Turki's Resignation" resulted in lower F1 scores, suggesting the model struggled to distinguish stances on these subjects. It's important to note that some topics might be inherently more challenging due to the nature of the discussion. For instance, "Coexistence with Religions" might involve a wider range of nuanced opinions that are difficult to categorize definitively as "favor" or "against." Overall, the results suggest that the MTSM-Keywords approach offers a promising foundation for stance detection in Arabic text. However, further investigation might be needed to improve performance on specific topics.

TABLE 6 F1 scores for "favor," "against," and "none" stances using MTSM as topic description.

Topic	Manual topic description					GPT4 topic description				
	F_{favor}	$F_{against}$	F_{none}	F_{avg2}	F_{avg3}	F_{favor}	$F_{against}$	F_{none}	F_{avg2}	F_{avg3}
Teaching Chinese Language at School	0.90	0.91	0.47	0.91	0.76	0.88	0.91	0.53	0.90	0.77
Improve School Curriculum	0.88	0.90	0.48	0.89	0.75	0.85	0.87	0.44	0.86	0.72
Online Learning	0.91	0.88	0.71	0.89	0.83	0.84	0.83	0.62	0.84	0.76
COVID-19 Vaccine	0.81	0.81	0.23	0.81	0.62	0.81	0.77	0.28	0.79	0.62
COVID-19 Vaccine Booster Dose	0.84	0.82	0.61	0.83	0.76	0.77	0.78	0.52	0.77	0.69
Aramco Share Selling	0.84	0.88	0.61	0.86	0.78	0.85	0.88	0.59	0.87	0.78
Al-Qiddiya Project	0.90	0.68	0.47	0.79	0.68	0.88	0.65	0.40	0.76	0.64
Neom City	0.90	0.75	0.71	0.83	0.79	0.90	0.78	0.67	0.84	0.78
Domestic Tourism	0.71	0.62	0.54	0.66	0.62	0.72	0.67	0.54	0.70	0.64
Sex Education	0.76	0.73	0.52	0.75	0.67	0.70	0.67	0.55	0.68	0.64
Coexistence with Religions	0.61	0.39	0.66	0.50	0.55	0.60	0.41	0.65	0.51	0.56
Military Conscription	0.77	0.76	0.57	0.77	0.70	0.71	0.67	0.64	0.69	0.67
Prince Abdulaziz bin Turki Head of Sports Minister	0.46	0.47	0.72	0.47	0.55	0.43	0.33	0.58	0.38	0.45
Prince Faisal bin Turki as Resignation from a Saudi club	0.43	0.43	0.50	0.43	0.46	0.58	0.20	0.39	0.39	0.39
Women_Driving	0.78	0.62	0.51	0.70	0.64	0.78	0.67	0.56	0.72	0.67
Mosques Speakers	0.48	0.76	0.33	0.62	0.52	0.54	0.78	0.36	0.66	0.56
Polygamous Marriage	0.80	0.85	0.53	0.82	0.73	0.83	0.84	0.45	0.84	0.71
AVERAGE OVER Avg2 & Avg3				0.74	0.67				0.72	0.65

Table 6 shows the results (F1 scores) for the MTSM (Multi-Topic, Single Model) approach with two different topic descriptions: manually crafted and generated by GPT-4. F1 score is a metric that balances precision and recall, providing an overall measure of model performance. Looking across the table, we see that both topic description methods achieved similar performance on average. The average F1 score for both "favor" and "against" stances is around 0.8 for both manual and GPT-4 descriptions, indicating good model performance in identifying supportive and opposing opinions. However, the results for the "none" stance, which represents tweets that don't express a clear opinion, are lower. The average F1 score for "none" is around 0.5 for both methods, suggesting more difficulty in accurately classifying neutral tweets.

There are some interesting variations between topics. For instance, both methods performed well on topics like "Teaching Chinese Language at School" and "Aramco Share Selling," achieving high F1 scores across all stances. Conversely, topics like "Coexistence with Religions" and "Mosques Speakers" proved more challenging, with lower F1 scores especially for the "none" stance. This suggests that these topics might be more nuanced or have a higher prevalence of neutral language, making stance detection more difficult. Overall, the results indicate that the MTSM approach with either manually crafted or GPT-4 generated topic descriptions can effectively identify supportive and opposing stances in Arabic text for a variety of topics. However, there's room for improvement in accurately classifying neutral tweets, and some topics may require further investigation or model improvements for better performance.

5.2 Performance analysis of few-shot learning model

This section explores the effectiveness of ArabicStanceX dataset in real-world situations where it might encounter entirely new topics, which were unseen during fine-tuning. This is particularly relevant for stance detection as new topics frequently emerge and quickly capture public attention. To address this challenge, we employed few-shot learning, specifically a methodology called "K-shot learning," which involves fine-tuning the model using only K examples per stance class (favorable, against, neutral) for a new topic. This ensures balanced representation across different stances even with limited data.

To evaluate our model's adaptability, we fine-tuned it on a comprehensive set of topics, excluding six specific ones reserved for testing (detailed in Table 7 through Table 10). This approach simulates a realistic scenario where new topics arise with scarce data available.

Table 7 shows the results (F1 scores) for the zero-shot learning scenario of the stance detection model using manually crafted topic descriptions. In a zero-shot setting, where the model encounters unseen topics, performance is understandably lower compared to

TABLE 7 Results for Zero-shot learning.

Topic	Manual topic description				
	F_{favor}	$F_{against}$	F_{none}	F_{avg2}	F_{avg3}
Online Learning	0.77	0.77	0.45	0.77	0.66
Neom City	0.82	0.57	0.41	0.69	0.60
Domestic Tourism	0.53	0.31	0.39	0.42	0.41
Military Conscription	0.66	0.53	0.49	0.60	0.56
Mosques Speakers	0.45	0.47	0.34	0.46	0.42
Multi Marriage	0.59	0.61	0.30	0.60	0.50
AVERAGE OVER Avg2 & Avg3				0.59	0.52

TABLE 8 Results for 10-shot learning.

Topic	Manual topic description				
	F_{favor}	$F_{against}$	F_{none}	F_{avg2}	F_{avg3}
Online Learning	0.87	0.85	0.47	0.86	0.73
Neom City	0.83	0.66	0.45	0.74	0.64
Domestic Tourism	0.70	0.58	0.46	0.64	0.58
Military Conscription	0.69	0.71	0.42	0.70	0.61
Mosques Speakers	0.31	0.67	0.33	0.49	0.44
Multi Marriage	0.68	0.75	0.36	0.71	0.60
AVERAGE OVER Avg2 & Avg3				0.69	0.60

previously trained topics. The average F1 score for both "favor" and "against" stances hovers around 0.6, indicating a basic ability to identify sentence but with less accuracy. The results for the "none" stance, representing neutral tweets, are even lower with an average F1 score of 0.34. This underscores the significant challenge the model faces in classifying neutral stances on completely new topics without any specific data for fine-tuning.

Examining individual topics, the model shows varied performance. It performed better on topics like "Online Learning" (average F1 score of 0.71), where opinions are likely more polarized. Conversely, topics such as "Domestic Tourism" and "Mosques Speakers" resulted in lower scores (average F1 score around 0.4), suggesting these topics might be more nuanced or contain more neutral language, complicating stance detection in a zero-shot scenario. Overall, the zero-shot learning results highlight the model's limitations when encountering entirely new topics. While it can still make some basic sentence predictions, the accuracy is significantly lower compared to trained topics. This emphasizes the importance of having some topic-specific data for improved performance in real-world applications.

We then employed incremental fine-tuning, progressively adapting the model with increasing amounts of data (10, 20, and 40 examples per class) for the new topics (Tables 8–10). This step-by-step approach allows us to observe the model's ability to learn from limited topic-specific data, which is crucial for real-world deployments. The significant performance improvements at the 40-shot level, with an average F_{avg2} score of 0.75, demonstrate that even a small amount of data can substantially enhance the model's effectiveness on unseen topics.

Table 8 shows the results (F1 scores) for stance detection on unseen topics using 10-shot learning with manually crafted topic descriptions, where the F1 score balances precision and recall to measure overall model performance. The average F1 score across all stances ("favor," "against," and "none") is 0.69 for F_{avg2} and 0.60 for F_{avg3}, indicating moderate performance on unseen topics even with limited data. Performance varies across topics, with higher scores for "Online Learning" and "Neom City" (around 0.7) and lower scores for "Mosques Speakers" and "Military Conscription" (around 0.5), highlighting challenges in these specific domains. The model struggles more with identifying neutral stances, consistently showing lower F1 scores for "none" compared to "favor" and "against." Overall, the results suggest that while the model can adapt to new topics with some success using 10-shot learning, there is a need for improvement in handling neutral stances and certain topic domains.

Table 9 presents the results (F1 scores) for stance detection on unseen topics using 20-shot learning with manually crafted topic descriptions, where the F1 score balances precision and recall for an overall measure of performance. The model performed well in identifying tweets expressing favorable (F_{favor}) and opposing ($F_{against}$) stances for most topics, with average F1 scores around 0.74, indicating effective learning of basic stance with limited data (20 examples per stance class). However, accurately classifying neutral tweets ("None") proved more challenging, with an average F1 score of around 0.46, highlighting difficulties in distinguishing neutral language from weakly expressed opinions on unseen topics. Performance varied across topics, with "Online Learning" and "Military Conscription" showing good performance across all stances. At the same time "Fix Domestic Tourism" and "Mosques Speakers" resulted in lower scores, particularly for the "None" stance, suggesting that topic complexity and the prevalence of neutral language influence the model's adaptability with limited data. Overall, the results demonstrate the model's potential for handling unseen topics with 20-shot learning, though improvement is needed in accurately classifying neutral stances and specific topic domains.

Table 10 shows the F1 scores achieved by the model using 40-shot learning with manually crafted topic descriptions. The F1 score, which balances precision and recall, provides an overall measure of model performance for each stance ("favor," "against," "none") on a specific topic. The average F1 scores (F_{avg2} and F_{avg3}) around 0.75 indicate that the model performs well on average, effectively identifying supportive and opposing opinions in Arabic text with just 40 examples per stance class for a new topic. However, performance varies across topics. For example, topics like "Online Learning" and "Military Conscription" achieved good results across all stances, with average F1 scores above 0.7, suggesting that the model can readily learn the stance patterns associated with these topics even with limited data. Conversely,

TABLE 9 Results for 20-shot learning.

Topic	Manual topic description				
	F_{favor}	$F_{against}$	F_{none}	F_{avg2}	F_{avg3}
Online Learning	0.89	0.89	0.67	0.89	0.81
Neom City	0.85	0.75	0.49	0.80	0.70
Domestic Tourism	0.69	0.65	0.49	0.67	0.61
Military Conscription	0.73	0.73	0.55	0.73	0.67
Mosques Speakers	0.46	0.60	0.36	0.53	0.47
Multi Marriage	0.81	0.78	0.46	0.80	0.68
AVERAGE OVER Avg2 & Avg3				0.74	0.66

TABLE 10 Results for 40-shot learning.

Topic	Manual topic description				
	F_{favor}	$F_{against}$	F_{none}	F_{avg2}	F_{avg3}
Online Learning	0.87	0.88	0.63	0.88	0.79
Neom City	0.87	0.74	0.61	0.80	0.74
Domestic Tourism	0.72	0.66	0.51	0.69	0.63
Military Conscription	0.72	0.72	0.59	0.72	0.67
Mosques Speakers	0.49	0.76	0.45	0.62	0.57
Multi Marriage	0.81	0.79	0.49	0.80	0.70
AVERAGE OVER Avg2 & Avg3				0.75	0.68

topics like "Fix Domestic Tourism" and "Mosques Speakers" proved more challenging, with lower average F1 scores, particularly for the "none" stance, indicating inherent complexity or specific challenges in identifying neutral stances in these contexts. Overall, the results are encouraging, demonstrating that the model can effectively adapt to new topics with 40 examples per stance, achieving good overall performance in stance detection for Arabic text while also highlighting the importance of considering topic-specific characteristics in real-world deployments.

6 Conclusion and discussion

This research focused on developing and evaluating a robust Arabic stance detection dataset, called ArabicStanceX, using a dataset derived from social media data. It addresses the lack of available Arabic stance detection datasets. Using the BERT architecture, we fine-tuned it to identify sentences across various topics in Arabic text.

Our exploration of different fine-tuning approaches revealed limitations with single-topic models, particularly in capturing the "none" stance and generalizing across diverse topics. In contrast, the MTSM approach showed promising results, especially when combined with manually crafted or GPT-4 generated topic descriptions.

Few-shot learning evaluations highlighted the model's potential for real-world applications, achieving good stance detection performance even with limited data (40 examples per stance class) for unseen topics. This adaptability is crucial for handling the dynamic nature of online discourse, where new topics frequently emerge.

Our findings emphasize the importance of considering topic-specific characteristics when deploying the model. Specific topics pose more significant challenges due to their complexity or the prevalence of neutral language. Future research should explore techniques to enhance performance on these nuanced topics and incorporate additional information sources beyond textual data. The results indicate that the MTSM approach, particularly with topic descriptions, holds promise for Arabic stance detection. The inclusion of topic keywords and descriptions provides the model with the necessary context for more informed predictions. Notably, manual topic descriptions were more effective than those generated by GPT-4, highlighting the potential need for human intuition in understanding nuanced topics.

However, the study has several limitations. The dataset focuses exclusively on Saudi Arabia and is sourced solely from X.com, which may restrict the generalizability of findings to other Arabic-speaking regions or platforms. Another limitation lies in class imbalance within specific topics, which may have negatively impacted the model's ability to detect minority stances. Additionally, the model struggled to handle nuanced language features such as sarcasm, implicit stances, and neutrality. Future work could expand the dataset to include other Arab countries and social media platforms, as well as explore alternative modeling approaches to better capture subtle linguistic cues. Addressing class imbalance could involve dataset resampling or data augmentation techniques.

In general, this work advances Arabic NLP by providing a foundation for effective stance detection in various topics of Arabic text. The developed model offers valuable insights into public stance and opinion dynamics within the Arabic-speaking world, with potential applications in social media analysis, market research, and other fields that rely on understanding audience perspectives. Future work should aim to improve the model's ability to detect neutral stances and enhance performance on complex and sensitive topics.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

AA: Data curation, Writing – original draft. FA: Validation, Writing – original draft, Project administration. FK: Investigation, Writing – review & editing. HA-K: Writing – review & editing, Writing – original draft.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. The Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia has funded this project under grant no. (KEP-1-611-42).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Al Hendi, K. D. (2024). Social media addiction and usage amongst dental students in saudi arabia-a comparative study. *J. Adv. Med. Dent. Sci. Res.* 12, 6–13.

AlDayel, A., and Magdy, W. (2021). Stance detection on social media: state of the art and trends. *Inf. Process Manag.* 58:102597. doi: 10.1016/j.ipm.2021.102597

Alhindi, T., Alabdulkarim, A., Alshehri, A., Abdul-Mageed, M., and Nakov, P. (2021). Arastance: a multi-country and multi-domain dataset of arabic stance detection for fact checking. *arXiv preprint arXiv:2104.13559*. doi: 10.18653/v1/2021.nlp4if-1.9

Allaway, E., and McKeown, K. (2020). Zero-shot stance detection: a dataset and model using generalized topic representations. *arXiv preprint arXiv:2010.03640*. doi: 10.18653/v1/2020.emnlp-main.717

Alturayeif, N. S., Luqman, H. A., and Ahmed, M. A. K. (2022). "Mawqif: a multi-label arabic dataset for target-specific stance detection," in *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)* (Stroudsburg, PA: Association for Computational Linguistics), 174–184. doi: 10.18653/v1/2022.wanlp-1.16

Baly, R., Mohtarami, M., Glass, J., Màrquez, L., Moschitti, A., and Nakov, P. (2018). Integrating stance detection and fact checking in a unified corpus. *arXiv preprint arXiv:1804.08012*. doi: 10.18653/v1/N18-2004

Bar-Haim, R., Bhattacharya, I., Dinuzzo, F., Saha, A., and Slonim, N. (2017). "Stance classification of context-dependent claims," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 251–261. doi: 10.18653/v1/E17-1024

Cignarella, A. T., Lai, M., Bosco, C., Patti, V., Rosso, P., et al. (2020). "Sardistance@ evalita2020: overview of the task on stance detection in italian tweets," in *CEUR Workshop Proceedings* (Aachen: CEUR), 1–10. doi: 10.4000/books.aaccademia.7084

Conforti, C., Berndt, J., Pilehvar, M. T., Giannitsarou, C., Toxvaerd, F., and Collier, N. (2020). Will-they-won't-they: a very large dataset for stance detection on twitter. *arXiv preprint arXiv:2005.00388*. doi: 10.18653/v1/2020.acl-main.157

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. doi: 10.48550/arXiv.1810.04805

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). "Bert: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (Stroudsburg, PA: Association for Computational Linguistics), 4171–4186.

Elmadany, A., Nagoudi, E. M. B., and Abdul-Mageed, M. (2022). Orca: a challenging benchmark for arabic language understanding. *arXiv preprint arXiv:2212.10758*. doi: 10.18653/v1/2023.findings-acl.609

Ferreira, W., and Vlachos, A. (2016). "Emergent: a novel data-set for stance classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (San Diego, CA: ACL). doi: 10.18653/v1/N16-1138

Gautam, A., Mathur, P., Gosangi, R., Mahata, D., Sawhney, R., and Shah, R. R. (2020). "#metooma: multi-aspect annotations of tweets related to the metoo movement," in *Proceedings of the International AAAI Conference on Web and Social Media, volume 14*, 209–216. doi: 10.1609/icwsm.v14i1.7292

Gorrell, G., Kochkina, E., Liakata, M., Aker, A., Zubiaga, A., Bontcheva, K., et al. (2019). "Semeval-2019 task 7: rumoureval 2019: determining rumour veracity and support for rumours," in *Proceedings of the 13th International Workshop on Semantic Evaluation: NAACL HLT 2019* (Minneapolis, MN: Association for Computational Linguistics), 845–854. doi: 10.18653/v1/S19-2147

Grimminger, L., and Klinger, R. (2021). Hate towards the political opponent: a twitter corpus study of the 2020 us elections on the basis of offensive speech and stance detection. *arXiv preprint arXiv:2103.01664*.

Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., et al. (2018). A retrospective analysis of the fake news challenge stance detection task. *arXiv preprint arXiv:1806.05180*. doi: 10.48550/arXiv.1806.05180

Hosseinia, M., Dragut, E., and Mukherjee, A. (2020). Stance prediction for contemporary issues: data and experiments. *arXiv preprint arXiv*:2006.00052. doi: 10.18653/v1/2020.socialnlp-1.5

Jaziriyan, M. M., Akbari, A., and Karbasi, H. (2021). "ExaASC: a general target-based stance detection corpus in arabic language," in *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)* (IEEE), 424–429. doi: 10.1109/ICCKE54056.2021.9721486

Kaur, R., Sachdeva, M., and Kumar, G. (2016). Nature inspired feature selection approach for effective intrusion detection. *Indian J. Sci. Technol.* 9, 1–9. doi: 10.17485/ijst/2016/v9i42/101555

Khouja, J. (2020). Stance prediction and claim verification: an arabic perspective. *arXiv preprint arXiv:2005.10410*. doi: 10.18653/v1/2020.fever-1.2

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. doi: 10.48550/arXiv.1412.6980

Lai, M., Patti, V., Ruffo, G., and Rosso, P. (2020). #brexit: leave or remain? the role of user's community and diachronic evolution on stance detection. *J. Intell. Fuzzy Syst.* 39, 2341–2352. doi: 10.3233/JIFS-179895

Li, Y., Sosea, T., Sawant, A., Nair, A. J., Inkpen, D., and Caragea, C. (2021). "P-stance: a large dataset for stance detection in political domain," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (Stroudsburg, PA: Association for Computational Linguistics), 2355–2365. doi: 10.18653/v1/2021.findings-acl.208

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., and Cherry, C. (2016). "Semeval-2016 task 6: detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (Stroudsburg, PA: Association for Computational Linguistics), 31–41. doi: 10.18653/v1/S16-1003

Mohammad, S. M., Sobhani, P., and Kiritchenko, S. (2017). Stance and sentiment in tweets. *ACM Trans. Internet Technol.* 17, 1–23. doi: 10.1145/3003433

Reimers, N., and Gurevych, I. (2019). Sentence-bert: sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*. doi: 10.18653/v1/D19-1410

Simsim, M. T. (2011). Internet usage and user preferences in Saudi Arabia. *J. King Saud Univ. Eng. Sci.* 23, 101–107. doi: 10.1016/j.jksues.2011.03.006

Sobhani, P., Inkpen, D., and Zhu, X. (2017). "A dataset for multi-target stance detection," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 551–557. doi: 10.18653/v1/E17-2088

Taulé, M., Martí, M. A., Rangel, F. M., Rosso, P., Bosco, C., Patti, V., et al. (2017). "Overview of the task on stance and gender detection in tweets on catalan independence at ibereval 2017," in *CEUR Workshop Proceedings, volume 1881* (CEUR-WS) (Stroudsburg, PA: Association for Computational Linguistics), 157–177.

Thakur, K., and Kumar, G. (2021). Nature inspired techniques and applications in intrusion detection systems: recent progress and updated perspective. *Arch. Comput. Methods Eng.* 28, 2897–2919. doi: 10.1007/s11831-020-09481-7

Vamvas, J., and Sennrich, R. (2020). X-stance: a multilingual multi-target dataset for stance detection. *arXiv preprint arXiv:2003.08385*. doi: 10.48550/arXiv.2003.08385

Wosom. (2024). Wosom. Available online at: https://wosom.ai/ (accessed May 26, 2024).

frontiers | Frontiers in Artificial Intelligence

TYPE Review
PUBLISHED 26 August 2025
DOI 10.3389/frai.2025.1638743

OPEN ACCESS

EDITED BY
Shadi Abudalfa,
King Fahd University of Petroleum and Minerals, Saudi Arabia

REVIEWED BY
Vasudevan Nedumpozhimana,
Trinity College Dublin, Ireland
Mohammed Erritali,
Université Sultan Moulay Slimane, Morocco
Shridhar Devamane,
Amrita Vishwa Vidyapeetham, India
Aadil Ganie,
Universitat Politécnica de Valéncia, Spain

*CORRESPONDENCE
Alaaeddine Ramadan
✉ alaaeddine.ramadan@aubh.edu.bh

RECEIVED 31 May 2025
ACCEPTED 31 July 2025
PUBLISHED 26 August 2025

CITATION
Saadiyeh O, Ramadan A, Hajjar M and Bernard G (2025) A comparative study of Arabic syntactic analyzers.
Front. Artif. Intell. 8:1638743.
doi: 10.3389/frai.2025.1638743

A comparative study of Arabic syntactic analyzers

Omar Saadiyeh[1], Alaaeddine Ramadan[2]*, Mohammad Hajjar[3] and Gilles Bernard[1]

[1]Paragraphe Research Lab, University of Paris VIII, Paris, France, [2]College of Engineering and Computing, American University of Bahrain, Riffa, Bahrain, [3]Faculty of Technology, Lebanese University, Saida, Lebanon

Syntactic analysis stands at the heart of Natural Language Processing (NLP), serving as the cornerstone upon which deeper linguistic understanding is built—particularly for morphologically complex languages such as Arabic. This paper delivers a comprehensive comparative study of contemporary syntactic analyzers designed explicitly for Arabic, dissecting the strengths and limitations of rule-based, statistical, machine learning, and hybrid methodologies, and recent neural network and transformer-based models. Given Arabic's intricate morphological structure and rich syntactic variation, accurately capturing syntactic relationships poses a significant challenge. To address this complexity, our study meticulously evaluates existing algorithms, highlighting advancements, performance gaps, and practical trade-offs. In addition, recognizing that robust syntactic parsing is anchored in high-quality annotated datasets, we provide a thorough overview of available Arabic treebanks and annotated corpora, emphasizing their critical role and contribution to syntactic parsing advancements. By synthesizing current efforts in the domain, this comparative analysis not only offers clarity on the state-of-the-art but also guides future research directions. Ultimately, our work seeks to empower NLP practitioners and researchers with nuanced insights, enabling more informed choices in the development of powerful, accurate, and linguistically insightful Arabic syntactic analyzers.

KEYWORDS

Arabic NLP, Arabic treebank, syntactic analysis, rule-based parsing, statistical parsing, hybrid parsing, neural parsing, transformer models

1 Introduction

Arabic is a Semitic language characterized by complex morphology, rich inflectional patterns, and flexible syntactic structures, posing significant challenges to natural language processing (NLP). Syntactic analysis, commonly referred to as parsing, is a critical step in NLP tasks such as machine translation, information retrieval, and sentiment analysis. Parsing Arabic, however, is particularly challenging due to linguistic phenomena such as diglossia, morphological ambiguity, and relatively free word order (Habash, 2010). Numerous parsing approaches have been proposed for Arabic, ranging from traditional rule-based systems to modern statistical and machine learning-based parsers. Early rule-based systems, primarily grounded in classical Arabic grammar rules, provided foundational insights but were limited by their scalability and adaptability (Othman et al., 2003). The advent of annotated corpora such as the Penn Arabic Treebank (PATB) facilitated data-driven methodologies, leading to significant advancements, including probabilistic context-free grammars (PCFGs), support vector machines (SVMs), and more recently, deep learning models utilizing contextualized word embeddings (Taji et al., 2017).

This paper provides a comprehensive survey of state-of-the-art Arabic syntactic analyzers developed in recent years. It systematically discusses key syntactic parsing approaches, exploring both rule-based and data-driven paradigms. Further, the paper evaluates prominent Arabic syntax treebanks and related resources that have enabled significant progress in parser development. Subsequently, we compare the performance of existing syntactic analyzers across various linguistic domains and applications. Finally, the study addresses ongoing challenges and limitations within the field, outlining avenues for future research.

2 Related work

Most existing review papers on Arabic syntactic parsing either broadly cover general NLP tasks or have become outdated in their specific analyses of syntactic parsing for Arabic. Dedicated comparative studies with a strict syntactic orientation remain scarce, and those available often overlook recent datasets or state-of-the-art parsing techniques.

Zaki et al. (2016) conducted one of the earlier comprehensive surveys focusing exclusively on Arabic syntactic parsers developed up to 2016. They categorize the parsers based on methodological approaches—rule-based, statistical, and hybrid—and clearly outline their advantages and limitations. Despite the breadth of this work, it now lacks coverage of subsequent developments in annotated datasets and parsing methodologies introduced post-2016. A more recent comparative study by Aqel et al. (2019) addressed advancements in Arabic parsing systems, highlighting their strengths and limitations, and providing suggestions to mitigate common parsing challenges. Although informative and relatively current, this work similarly falls short in referencing the latest syntactic annotation schemes and updated parsing datasets that have emerged after its publication.

Recent surveys addressing broader linguistic contexts have also appeared, such as those by Hamed et al. (2025), examining code-switched Arabic NLP, and Xu et al. (2025), exploring multilingual large language models. While valuable, these studies primarily focus on multilingual and cross-lingual scenarios and do not specifically target syntactic parsing of Arabic, highlighting a clear gap in the literature for a dedicated, syntax-focused comparative study for Arabic.

In summary, the literature reflects a notable scarcity of recent and specialized comparative studies that focus explicitly on Arabic syntactic parsing. The present study addresses this gap by offering a comprehensive and current analysis specifically targeted at syntactic parsers for Arabic, incorporating insights from recent developments and datasets.

To better contextualize the reviewed work, we briefly outline the fundamental concepts and methodologies in syntactic analysis. Syntactic analysis, or parsing, refers to the process of analyzing sentences by identifying their syntactic structure according to a set of grammatical rules. This task is fundamental in natural language processing (NLP) and computational linguistics, as it helps in understanding sentence structure and meaning. In the context of Arabic, syntactic analysis can be approached in several ways, each offering distinct advantages depending on the complexity and formality of the grammar involved.

2.1 Approaches to syntax analysis

Syntactic analysis can be approached using two primary methods:

- Top-Down Parsing: This method starts with the entire sentence and breaks it into smaller parts (constituents) using grammar rules. These parts are further divided until you reach individual words. This strategy works well with grammars that focus on sentence structure (Aho et al., 2006).
- Bottom-Up Parsing: This method begins with the words in the sentence, assigning each a grammatical label. These labels are then combined to form higher-level structures (like phrases) until the full sentence structure is built. This method works with many types of grammar (Aho et al., 2006).

2.2 Available parsing algorithms

The selection of parsing algorithms is critical to the efficiency and effectiveness of syntactic analysis. Two prominent algorithms are:

- Cocke-Younger-Kasami Algorithm: A fast, table-based parsing method for context-free grammars, especially effective when the grammar is in Chomsky Normal Form (Brandt and Walter, 2001).
- Earleys Algorithm: A flexible algorithm that works with both normalized and non-normalized context-free grammars (Tendeau, 1997).

2.3 Parsing techniques

Several approaches to syntactic analysis in Arabic focus on different methods and techniques, including:

- Rule-based approach: which uses a well-defined formal grammar based on the knowledge of linguists on the language concerned;
- Statistical approaches: which uses machine learning techniques to create grammar rules from a corpus annotated (TreeBank), then analyzes the sentences using these rules;
- Hybrid approach: which uses both a predefined grammar and a statistical module (for example a disambiguation module) allowing to improve the results and to resolve the ambiguities.

2.4 Depth of parsing

In syntactic analysis, the term "depth of parsing" refers to the extent and precision of syntactic information extracted from a given sentence. This concept plays a critical role in shaping the goals and applications of parsing systems, especially for morphologically rich and structurally flexible languages such as Arabic. Generally, parsing approaches fall into two broad categories based on depth: deep parsing and shallow parsing.

- **Deep parsing:** Deep parsing involves generating a full syntactic structure for a sentence, capturing the complete grammatical relationships among words and phrases. This typically results in hierarchical representations like constituency trees or dependency graphs, which identify syntactic roles such as subjects, objects, and modifiers. For Arabic, deep parsers often rely on resources like the Penn Arabic Treebank and are capable of handling sophisticated linguistic features, albeit with significant computational cost (Habash, 2010; Taji et al., 2017). These parsers are valuable for tasks requiring nuanced understanding of sentence structure, such as machine translation and semantic analysis.
- **Shallow parsing:** Also known as chunking, shallow parsing focuses on identifying the main syntactic units within a sentence, such as noun phrases or verb groups, without delving into their internal grammatical structure or hierarchical organization. This approach is generally faster and more robust, particularly in noisy or resource-scarce settings. In Arabic NLP, shallow parsing is often used in applications like named entity recognition and basic information extraction, where full parsing is unnecessary (Shaalan and Khaled, 2010).

Each method presents advantages depending on the use case. Deep parsing provides comprehensive syntactic insight but demands more processing power and annotated data. Shallow parsing offers efficiency and adaptability, especially for preliminary or large-scale language tasks. In practice, hybrid models that combine both levels of analysis are becoming increasingly common in Arabic syntactic processing.

3 Arabic syntax treebanks and resources

The development of Arabic syntactic parsers relies heavily on annotated treebanks, which provide valuable resources for training and evaluating parsers. Notable Arabic treebanks include:

Penn Arabic Treebank (PATB) employs a statistical approach for annotating Modern Standard Arabic, focusing on structural morphology and syntactic analysis. It includes comprehensive annotations for parts of speech (POS), morphology, gloss, and syntactic trees. The corpus consists of 599 articles from the Lebanese newspaper *An Nahar*, totaling 402,291 word tokens. The annotations, following the Penn Treebank guidelines, are used for syntactic parsing and language modeling. Evaluation results across multiple versions demonstrate high accuracy, with more than 99% of tokens correctly tagged for POS and morphological analysis, ensuring robust reliability for linguistic and computational applications (Maamouri et al., 2004, 2005).

Prague Arabic Dependency Treebank (PADT) is grounded in a theoretical approach inspired by the Functional Generative Description framework and the Prague Dependency Treebank. It includes over 113,500 tokens with detailed syntactic and morphological annotations. This treebank is designed to aid dependency parsing and has been utilized in the CoNLL shared tasks, showcasing its utility in parsing experiments. The dataset covers 212,500 words, with a strong focus on syntactic dependencies. Its evaluation results highlight the accuracy of dependency relations, supporting the treebank's role in both theoretical and practical parsing tasks (Hajič et al., 2004, 2006).

Columbia Arabic Treebank (CATiB) adopts a simplified dependency-based approach that emphasizes annotation speed and efficiency. It provides syntactic analyses, including over 1 million tokens, with 841,000 words and 31,319 trees from newswire feeds and other sources. CATiB uses a reduced set of syntactic labels compared to PATB, prioritizing accessibility for annotators with less linguistic expertise. The evaluation results indicate a balance between simplicity and depth, offering a practical resource for rapid syntactic analysis while maintaining high accuracy for basic syntactic relations in Arabic (Habash and Roth, 2009).

CAMEL Treebank (CAMELTB) is a comprehensive dependency treebank for both Modern Standard Arabic and Classical Arabic, annotated using guidelines aligned with CATiB. It includes approximately 188,000 words and 242,000 tokens from a variety of genres, including poetry, religious texts, and modern media. CAMELTB uses tools like CamelTools for tokenization and POS tagging, and the MALT parser for syntactic parsing. Its manual annotation process ensures high accuracy, with four native Arabic speakers involved in annotating and editing dependency relations. Evaluation results show the treebanks broad applicability across different Arabic dialects and registers, making it a valuable resource for linguistic research and NLP applications (Habash et al., 2022).

Universal dependencies for Arabic project utilizes dependency-based annotations from the Prague Arabic Dependency Treebank (PADT) and the Penn Arabic Treebank (NYUAD version) (Taji et al., 2017; Hajič et al., 2004). These datasets provide a robust foundation for analyzing Arabic syntax and morphology, addressing the challenges posed by the language's rich inflection and word formation. The annotations cover several layers, including part-of-speech tags, lemmas, morphological features, and syntactic relations. The project adopts a consistent approach to tokenization and morphological representation across different Arabic dialects, ensuring broad linguistic coverage. Evaluation of these treebanks emphasizes syntactic accuracy, with UD Arabic-PADT featuring 7,664 sentences and 242,056 tokens, and UD Arabic-NYUAD containing 19,738 sentences and 629,295 tokens. These treebanks offer comprehensive linguistic resources, enabling in-depth analysis of Arabic within the Universal Dependencies framework.

AQMAR Arabic Wikipedia dependency tree corpus (Habash et al., 2009) is derived from Arabic Wikipedia articles, annotated with part-of-speech (POS) tags and syntactic dependencies. This corpus comprises 1,262 sentences and 36,202 tokens, created with a semi-automated annotation process using the Brat annotation tool. The initial POS tagging was performed using the MADA system, followed by manual corrections. Dependency annotations were applied according to the CATiB Arabic dependency framework (Habash and Roth, 2009), ensuring high-quality syntactic representations. The dataset includes diverse topics, such as nuclear technology and football, providing valuable resources for semantic and syntactic analysis in various domains. While the

annotations also cover named entities and semantic supersenses, the evaluation results primarily highlight improvements in syntactic parsing and dependency structure accuracy.

ARL Arabic dependency Treebank, developed by the US Army Research Laboratory (ARL) (Tratz, 2016), focuses on Arabic news and broadcast sources. This treebank is a restructured version of the Arabic Treebank (ATB) from the Linguistic Data Consortium, and it adopts a dependency grammar approach. Each sentence is analyzed based on a verb-centered structure, with other elements linked to the verb through directed relationships. The annotations include 11 columns, detailing the syntactic dependencies, POS tags, and lemmata, with each word or affix uniquely identified. Evaluation of the treebank involves measuring the quality of dependency relations and syntactic parsing, making it a crucial resource for Arabic language processing in military and defense applications. The dataset is available for further use in research and development of Arabic language technologies.

OntoNotes 5.0 (Weischedel et al., 2013) is a large annotated corpus containing multiple linguistic layers, including syntactic, semantic, and discourse-level annotations. The Arabic portion, comprising 300K words, includes part-of-speech tagging, coreference, named entity recognition, and word sense disambiguation. The syntactic annotations use the Treebank framework, while the semantic annotations link word senses to an ontology. Evaluation results demonstrate high quality in both syntactic and semantic annotations, with comprehensive coverage of co-reference and named entities. The corpus provides a valuable resource for training machine learning models and evaluating Arabic language processing tasks. Available in both relational database format and text files, OntoNotes supports a range of research applications, including cross-linguistic studies and deep semantic parsing.

I3rab Treebank (Halabi et al., 2020) is a new Arabic dependency treebank that introduces innovative approaches to tokenization and dependency representation, focusing on the identification of primary words and the treatment of joined and implicit pronouns. The corpus is compared against a subset of the Prague Arabic Dependency Treebank (part-PADT), with evaluation results showing significant improvements in parsing performance. The I3rab dataset demonstrated a 7.5% increase in Unlabeled Attachment Score (UAS) and an 18.8% improvement in Labeled Attachment Score (LAS), highlighting the effectiveness of its unique approach. This treebank is intended to advance Arabic language processing by addressing gaps in previous dependency frameworks and offering a more accurate representation of syntactic relations in Arabic.

Arabic Poetry Dependency Treebank (ArPoT) (Al-Ghamdi et al., 2021) introduced ArPoT, the first dependency treebank specifically targeting classical Arabic poetry. The corpus consists of 2,685 verses (35,460 tokens) from 34 poets, annotated using the CATiB scheme, which is rooted in traditional Arabic grammar and supports future conversion to Universal Dependencies. ArPoTs annotation pipeline involved automatic parsing (using a tool trained on MSA) followed by extensive manual correction, with explicit attention to poetic-specific phenomena such as elision and cross-verse syntactic relations. Unlike most previous Arabic treebanks (e.g., Penn Arabic Treebank, CATiB, PADT) which are constructed for Modern Standard Arabic (MSA), ArPoT is dedicated to CA and captures its unique syntactic characteristics, making it a novel resource for the study of syntactic analysis in Arabic poetry.

NArabizi Treebank (Riabi et al., 2023) is a syntactically annotated corpus for North African Arabic (specifically Algerian dialect) written in Latin script—commonly known as NArabizi. The dataset consists of approximately 1,300 user-generated sentences, primarily sourced from online forums and song lyrics, with significant code-switching (36% French tokens). The latest version introduces major improvements, including standardized tokenization, corrections of morpho-syntactic and syntactic annotations following Universal Dependencies (UD) guidelines, and enhanced translation quality. Two new annotation layers were added: named entity recognition and offensive language detection, making the resource more versatile for downstream tasks. The treebank focuses exclusively on dialectal Arabic and does not include Modern Standard Arabic (MSA). However, its syntactic annotation—covering POS tags, morphological features, and dependency parses—serves as an essential benchmark for NLP tasks on noisy, low-resource Arabic varieties written in non-Arabic scripts. Experimental results showed that improving syntactic annotation quality led to significant gains in downstream dependency parsing and NER. The resource is freely available for research purposes.

AraFast (Alrayzah et al., 2024) is a large-scale, freely available Modern Standard Arabic (MSA) corpus aimed at addressing the shortage of comprehensive datasets for Arabic NLP research. The authors developed a multi-stage pipeline, combining automated and manual discovery of Arabic corpora from major repositories (such as GitHub, Kaggle, and Huggingface), followed by strict filtering for quality and genre, and extensive cleaning using custom algorithms. This process included deduplication, removal of noise, normalization, and segmentation with the WordPiece tokenizer. The final AraFast corpus comprises 112 GB of high-quality MSA and classical Arabic text from 48 different sources, reduced from an initial 833 GB of raw data through rigorous preprocessing. Importantly, it should be noted that AraFast is *not* a syntactically annotated resource such as a treebank; it does not include part-of-speech or syntactic structure annotations. Instead, AraFast provides a high-quality, segmented text corpus specifically designed for pretraining large transformer-based language models, using dynamic span-masking objectives. Both "base" (full corpus, 110M parameters) and "mini" (10GB) models were trained and evaluated. The experimental results showed that using segmented, clean data substantially improved model learning and stability (evidenced by lower training loss), while web-scraped noisy data led to training failures due to noise and data artifacts. While AraFast itself does not provide direct syntactic labels or parsing, its quality and scale make it a valuable foundational dataset. It indirectly supports advances in Arabic syntactic parsing by enabling the training of robust pre-trained language models, which can later be fine-tuned or adapted for downstream syntactic analysis tasks. Thus, AraFast serves as an important resource for both general and syntactic NLP applications in Arabic.

4 Available syntactic analyzers

Over the years, a wide array of Arabic syntactic analyzers have been developed, mirroring the progression of parsing techniques. Early parsers predominantly relied on manually crafted grammar rules and limited evaluation datasets, whereas subsequent systems leveraged machine learning trained on treebanks. In recent years, neural network and transformer-based parsers have achieved new state-of-the-art results by incorporating contextualized language models. The following subsections review representative Arabic parsers across these different paradigms, highlighting their approaches and reported performance.

4.1 Traditional syntactic analyzers for arabic

Analyzer based on a recursive transition network is a syntactic analyzer developed by Bataineh and Bataineh (2009) uses a Top-Down parsing approach based on Recursive Transition Networks (RTN), a concept derived from recursive transition grammars. The grammar for this parser is context-free, tailored to capture the most frequent sentence structures in Arabic. The approach applies both pattern-based rules and context-free rules, treating them as complementary. It was tested on 90 Arabic sentences, achieving an accuracy rate of 85.6%. However, the parser struggled with ungrammatical sentences and those outside the grammar's coverage, with 14.4% of sentences being unparseable.

A'reb, developed by Al-Daoud and Basata (2009), is a recursive, Top-Down parser designed to handle both lexical and syntactic analysis for Arabic sentences, focusing on verbal sentences. It utilizes recursive functions closely tied to production rules, allowing the parser's structure to reflect the grammar it interprets. Despite its functionality, the authors noted that further refinement is needed for complete effectiveness, with no quantitative evaluation results provided.

Parse trees of Arabic sentences using NLTK (Shatnawi and Belkhouche, 2012) is a rule-based approach utilizing Context-Free Grammar (CFG). The parser applies the NLTK recursive-descent algorithm to generate parsing trees for general and Quranic Arabic. Although it supports several NLP tasks, the authors pointed out that the model does not address more complex tasks like parsing dependencies, and no quantitative performance metrics were provided.

Chart parser for analyzing Arabic sentences (Al-Taani et al., 2012) is a Top-Down chart parser based on Context-Free Grammar (CFG) to analyze Arabic sentences. The parser's accuracy was evaluated on a small corpus of 70 sentences, with an average sentence length of 3.98 words, achieving 94.3% accuracy. However, the authors emphasized the need for further evaluation with a broader corpus to test the parser's reliability in diverse contexts.

Context-free Grammar analysis top-down technique (Al-qrainy et al., 2012) developed an Arabic parser based on Context-Free Grammar (CFG) and Top-Down recursive descent parsing using NLTK. The parser was tested on 150 Arabic sentences, achieving a high accuracy rate of 92% for verbal sentences and 98% for nominal sentences. However, the test set was small, and the types of sentences evaluated were unspecified, which limits the reliability of the results.

ARSYPAR (Khoufi et al., 2013) introduced an Arabic parser that uses supervised machine learning techniques, specifically Support Vector Machines (SVM). The parser was trained using features derived from the Arabic Treebank and focused on syntactic word classes. It was evaluated on a portion of the Arabic Treebank, achieving an F-score of 84.38%, demonstrating the efficacy of statistical methods in syntactic analysis.

Industrial-strength parser (Redjaimia et al., 2014) developed an advanced Arabic parser combining rule-based and statistical approaches to provide robust dependency and hierarchical constituent parsing. The parser underwent rigorous testing on a corpus of 300 Arabic sentences, achieving an F-score of 82%. This hybrid approach proved effective for applications like opinion mining in Arabic social media content, although the specific evaluation methodology was not detailed.

Robust large-scale parser using AGFL formalism (Ouersighni, 2014) used a rule-based approach with Affix Grammars over Finite Lattice (AGFL) formalism for parsing Arabic. The parser's robust performance was tested on 200 Arabic sentences, achieving a 95% success rate. However, it suffered from high ambiguity, with an average of 23.12 possible analyses per sentence, highlighting the trade-off between robustness and precision in this approach.

Transducers parser (Hammouda and Haddar, 2018) employed a transducers-based approach to parse Arabic nominal sentences. The system, which includes segmentation, preprocessing, and disambiguation phases, achieved a precision rate of 80% and a recall rate of 90% when tested on a corpus of 200 Arabic sentences. This method proved effective for nominal sentence parsing but may require further refinement for broader sentence structures.

Inductive learning algorithm (ILA) (Abu-Soud et al., 2018) developed an ILA to parse Arabic nominal and verbal sentences. The ILA generates parsing rules from a training dataset and achieved a 92.63% accuracy for previously unseen sentences. However, it performed better on verbal sentences compared to nominal ones, due to the structural complexity of the latter. The method demonstrated its potential for Arabic Natural Language Processing (ANLP) applications but highlighted the challenges of segmenting and tagging sentences accurately.

Arabic parser based on CFG and classical grammar rules (Ababou et al., 2017) proposed an Arabic parser using Context-Free Grammar (CFG) integrated with classical grammar rules. The system achieved 97% accuracy when tested on 200 nominal sentences, effectively identifying dependency relations. However, some verb tagging errors were noted, and the method's simplicity allows easy integration with other techniques, enhancing its adaptability in parsing Arabic sentences.

Syntactic parsing using the NooJ linguistic platform is syntactic analyzer employs a rule-based, linguistically driven approach for Arabic syntactic parsing (Bourahma et al., 2018). Focusing on enhancing lexicon classification, resolving ambiguities from morphological analysis, and modeling grammar based on nominal sentence structures. The evaluation of the system on 120 nominal sentences demonstrated a parsing accuracy of 95%, with

disambiguation achieving an 86% accuracy. Despite the success, ambiguities remain in complex sentence structures, highlighting the challenge of fully capturing Arabics syntactic nuances. The approach proves effective in handling agglutination and word order variability.

Multitask easy-first dependency parsing uses a bottom-up parsing strategy with a multitask learning approach (Kankanampati et al., 2020). It simultaneously learns from two Arabic dependency treebanks (CATiB and UD) by parsing both syntactic and semantic features. Their model jointly parses sentences into both syntactic representations using shared and task-specific components, allowing partial parse trees in one formalism to inform decisions in the other. This approach is evaluated on parallel CATiB and UD treebanks—both automatically converted from parts 1–3 of the PATB—with standard train/dev/test splits. While these converted treebanks are not originally designed for dependency parsing, they are widely used as gold standards for syntactic analysis in Arabic NLP research. The multitask parser achieves substantial improvements over strong single-task baselines, with labeled attachment scores (LAS) of 86.15 for CATiB and 84.76 for UD, representing 9.9% and 6.1% error reductions respectively. The study highlights that explicit sharing of partial tree structures, rather than just neural parameter sharing, yields the largest gains, especially in complex syntactic constructions such as Idafa and modifiers.

An Arabic probabilistic parser based on a property grammar is a parser that uses a hybrid approach combining statistical modeling and rule-based parsing, based on a Property Grammar (PG) formalism (Bensalem et al., 2023). The parser applies a bottom-up parsing strategy using a Probabilistic Context-Free Grammar (PCFG) combined with a probabilistic Property Grammar (PPG). It integrates syntactic constraints and utilizes the CYK algorithm optimized with the Viterbi method. Evaluation on a test set of 400 sentences from ATB highlights the parser's ability to parse complex Arabic constructs with high precision. Compared to the Stanford parser (Dozat et al., 2017), it demonstrates better precision for specific linguistic phenomena, such as verbal sentences (88.3% vs. 81.9%) and nominal phrases (75.2% vs. 74.0%). However, it faces challenges in recall, particularly in capturing all relevant syntactic features.

Bel-Arabi combines both rule-based and learning-based approaches for Arabic syntactic parsing (Ibrahim et al., 2016). The system adopts a machine learning strategy for tasks like POS tagging and chunking, employing Conditional Random Fields (CRF) classifiers. The framework also integrates rule-based modules for grammatical marking, ensuring accurate syntactic analysis. With a high precision rate (90.44%) for analyzing 600 sentences, the system excels at identifying grammatical roles and diacritical marks. However, its performance declines when dealing with constructs like passive verbs, indicating areas for improvement, particularly in semantic analysis.

Arabic parser using deep learning employs deep learning techniques to tackle the complexities of Arabic syntax, utilizing bidirectional LSTM (BILSTM) models (Maalej et al., 2021). The system employs a statistical approach for syntactic parsing, utilizing deep learning models such as LSTM, GRU, and BILSTM, which are trained on word embeddings derived from the Penn Arabic Treebank (ATB). The BILSTM model demonstrated superior accuracy, achieving over 99% accuracy across various syntactic levels. The system effectively captures bidirectional contextual dependencies, making it a promising approach for Arabic syntactic parsing in NLP applications.

Stanford Arabic parser is a component of the Stanford CoreNLP suite that provides syntactic analysis of Arabic sentences using probabilistic context-free grammar (PCFG) models (Green and Manning, 2010). It is trained on the Penn Arabic Treebank (PATB) and operates in two main stages: first, it performs tokenization and segmentation—often using the Stanford Arabic Segmenter, and then applies syntactic parsing to produce hierarchical phrase structure trees.

The parser generates both constituency trees and part-of-speech (POS) tags, enabling deeper syntactic understanding necessary for downstream tasks like information extraction, question answering, and machine translation. It utilizes the CYK (Cocke–Younger–Kasami) parsing algorithm and supports features like n-best parses and probabilistic scoring, making it both powerful and flexible for diverse NLP applications. Although the parser itself doesn't perform sentiment analysis, its output supports sentiment models. Grammar-checking tools use the parser to identify and correct errors, and NER systems benefit from its contextual information. In educational settings, the parser teaches syntax and sentence structure, while businesses use it for text analytics, such as market research and customer feedback analysis. The parser's comprehensive applications demonstrate its versatility in understanding and processing natural language text.

The parser's performance on development test data for sentences under 40 words shows a factored F1 score (factF1) of 77.44% and dependency accuracy (factDA) of 84.05%. For the ATB part 3 Buckwalter grammar. These results highlight strong dependency parsing performance and suggest that inconsistencies in constituency annotations may account for the relatively lower F1 scores.

Arabic tree adjoining grammar (ArabTAG V2.0) or Arabic Tree Adjoining Grammar version 2.0, is an advanced syntactic and semantic analysis framework specifically designed for Modern Standard Arabic. Developed as part of a project led by researchers like Ben Khelil et al. (2023) and her collaborators, this grammar addresses the unique challenges posed by NLP, including its flexible word order, rich morphology, and the omission of diacritics in written texts. ArabTAG V2.0 builds on a prior manually defined grammar, enhancing it with an abstract representation called a meta-grammar. This abstraction allows linguists to describe both the syntax and semantics of Arabic more efficiently, facilitating the maintenance and expansion of the grammar. The framework includes 1,074 non-lexicalized syntactic rules and 27 semantic frames, focusing on predicate-argument structures.

The grammar is semi-automatically generated and is designed to cover a wide range of syntactical structures and linguistic phenomena. Experimental evaluations have shown that ArabTAG V2.0 can achieve a precision rate of 88.76% in syntactic analysis and about 95.63% in semantic analysis. This high level of accuracy demonstrates its capability to handle the complexity of Arabic syntax and semantics effectively.

MASAQ parser (Sawalha et al., 2025b) is a recent statistical parser developed for Classical Arabic, based on the newly released MASAQ dataset (Sawalha et al., 2025a). It applies supervised machine learning (Random Forest, LinearSVC, Logistic Regression) for fine-grained morphosyntactic analysis, focusing on dependency parsing in accordance with traditional Arabic *irab*. The MASAQ corpus includes 131,930 morphemes and 123,565 annotated syntactic functions over 77,408 Quranic words. Evaluation experiments report a best accuracy of 99.0% for syntactic role assignment using Random Forest, setting a new benchmark for Arabic syntactic analysis.

4.2 Modern neural and transformer-based approaches to arabic syntactic analysis

Camel parser, which includes versions 1.0 and 2.0 (Elshabrawy et al., 2023), integrates machine learning, specifically leveraging BERT-based embeddings for better contextual understanding, and applies biaffine attention mechanisms for dependency parsing. CamelParser 2.0 outperforms its predecessor by integrating advanced neural models, yielding improved parsing performance with a Labeled Attachment Score (LAS) of 91.3% and an Unlabeled Attachment Score (UAS) of 92.4%. The use of BERT and biaffine parsing results in a significant reduction in parsing errors, making it a robust tool for Arabic dependency parsing.

Out-of-domain dependency parser (Mokh et al., 2024) address the challenge of dependency parsing for Arabic dialects in an out-of-domain setting, given the lack of syntactically annotated dialectal corpora. Their approach uses a neural biaffine dependency parser (Dozat and Manning, 2016), trained on the Columbia Arabic Treebank (CATiB; Habash and Roth, 2009) and the Modern Standard Arabic (MSA) portion of the MADAR parallel corpus (Bouamor et al., 2018), and tested on a manually annotated set of Gulf, Levantine, Egyptian, and Maghrebi dialect sentences. They focus on the parsing of Idafa and coordination constructions, which are particularly challenging and structurally variable across dialects. The authors employ various domain adaptation strategies, including filtering training data by sentence length, removing sentential coordination, selecting structurally similar sentences based on POS bigram perplexity, and experimenting with different BERT-based embeddings. For in-domain evaluation, they used two syntactically annotated MSA datasets: CATiB and the MSA portion of the MADAR corpus, which consists of 2,000 sentences with full dependency. When trained and evaluated on CATiB, their parser achieved a Unlabeled Attachment Score (UAS) of 90.3% and a Labeled Attachment Score (LAS) of 88.7%. On the MADAR MSA dataset (2,000 annotated sentences), the parser reached a UAS of 97.9% and a LAS of 84.9%. However, performance drops significantly out-of-domain (e.g., UAS: 55.1–57.5%, LAS: 23.2–27.5% across dialects), but targeted adaptation techniques can raise LAS by up to 24 points for specific constructions. These results serve as an upper bound for parsing performance in MSA, given matched domain and annotation style.

AraT5 (Nagoudi et al., 2022) is an Arabic text-to-text Transformer model trained on large-scale MSA and dialectal corpora, including AraNews (Nagoudi et al., 2020), El-Khair (El-Khair, 2016), and OSCAR (Suárez et al., 2020). While AraT5 does not function as an explicit syntactic analyzer, its sequence-to-sequence architecture and pretraining enable it to learn syntactic structures *implicitly*, as demonstrated by strong results on the ARGEN benchmark across seven tasks. AraT5 outperformed mT5 on 52 of 59 test splits, highlighting the effectiveness of implicit syntax modeling for Arabic language generation and understanding tasks.

AraBERT (Antoun et al., 2020) is a transformer-based language model specifically pre-trained for Arabic. Built on the BERT-base architecture (12 encoder layers, 768 hidden dimensions, 110M parameters), AraBERT introduces an Arabic-specific preprocessing pipeline by segmenting words into stems, prefixes, and suffixes using Farasa (Abdelali et al., 2016), followed by sub-word tokenization (SentencePiece, vocab size: 64K). The model is pre-trained on a large, diverse corpus comprising 70 million sentences (24GB) gathered from major Arabic news sources [notably the 1.5B Arabic Corpus (El-Khair, 2016) and OSIAN (Zeroual et al., 2019)], Modern Standard Arabic (MSA), and dialectal variants. Although AraBERT is not an explicit syntactic parser, its deep contextualized embeddings have shown strong performance on tasks highly dependent on syntactic and morphological understanding, making it widely adopted as a backbone for downstream syntactic analysis tasks. In evaluations across sentiment analysis, named entity recognition (NER), and question answering (QA), AraBERT consistently outperformed multilingual BERT and previous state-of-the-art models. The size and diversity of the training corpus and the Arabic-specific tokenization are key contributors to its robust syntactic modeling.

MARBERT (Abdul-Mageed et al., 2021) is a pre-trained deep bidirectional Transformer model specifically designed to address the diversity and informality of Arabic language varieties, especially on social media. Built on the BERT-base architecture (12 layers, 768 hidden units, 163M parameters), MARBERT is trained from scratch on a massive dataset of 1 billion Arabic tweets (128GB, 15.6B tokens), using a 100K WordPiece vocabulary. The pre-processing is intentionally minimal—removing only diacritics and normalizing URLs, usernames, and hashtags—to maximize the model's exposure to authentic, naturally occurring dialectal and noisy text. Importantly, while MARBERT is not a syntactic parser in the traditional sense, its deep contextualized representations have shown substantial impact on downstream tasks that depend on syntactic and morphosyntactic cues, such as named entity recognition, dialect identification, and question answering. For evaluation, MARBERT was assessed using the ARLUE benchmark (Abdul-Mageed et al., 2021), which consists of 42 diverse datasets across six task clusters (including tasks closely tied to syntactic analysis). MARBERT achieves state-of-the-art results on 37 out of 48 classification tasks, with an overall ARLUE macro-average score of 75.99, outperforming many larger multilingual models (such as XLM-RLarge, which is more than three times larger in parameters). Notably, MARBERT's strength is most pronounced in dialect identification and social meaning tasks—domains where syntactic variation is high and previous MSA-focused models struggled. To further address performance in tasks requiring longer context, the authors introduce MARBERTv2, which is obtained by continued

pre-training of MARBERT on the same MSA data as ARBERT and the AraNews dataset, using a longer sequence length (512 tokens) for 40 additional epochs, resulting in exposure to 29 billion tokens.

Dialect-specific pre-trained language models: In addition to multidialect models like AraBERT and MARBERT, recent research has introduced several dialect-specific pre-trained language models, including CAMeLBERT (Inoue et al., 2021), SaudiBERT (Qarah, 2024b), and EgyBERT (Qarah, 2024a). CAMeLBERT comprises a suite of BERT-based models, each trained on a specific Arabic variant (Modern Standard Arabic, dialectal Arabic, or Classical Arabic), with pre-training corpora ranging up to 167GB and over 17 billion tokens. SaudiBERT is developed for the Saudi dialect using a corpus of 141 million Saudi tweets and forum data (totalling over 26GB), while EgyBERT targets the Egyptian dialect with more than 10GB of Egyptian tweets and forum texts. These models follow the BERT architecture and employ minimal pre-processing to preserve dialectal characteristics. Though not syntactic parsers, their contextualized representations significantly improve the performance of downstream tasks that require syntactic sensitivity.

Al-Ghamdi et al. (2023) proposed a novel approach for Arabic dependency parsing by fine-tuning BERT-based pre-trained language models, formulating the parsing task as a sequence labeling problem. Each token is assigned a composite label encoding both the head position and the dependency relation, and three head-encoding strategies (naive positional, relative positional, and relative POS-based) were systematically compared. The authors evaluated nine Arabic BERT-based models—including AraBERTv2, AraBERTv1, Camel-MSA, Camel-CA, ARBERT, and GigaBERT—on three treebanks: the Prague Arabic Dependency Treebank (PADT, Hajič et al., 2004), the Columbia Arabic Treebank (CATiB, Habash and Roth, 2009), and the Classical Arabic Poetry Dependency Treebank (ArPoT, Al-Ghamdi et al., 2021). Experimental results demonstrate that AraBERTv2 achieved the highest accuracy, reaching up to 84.03% UAS and 80.26% LAS on PADT, 87.54% UAS and 86.41% LAS on CATiB, and 79.79% UAS and 74.13% LAS on ArPoT. It should be noted that the work by Al-Ghamdi et al. (2023) does not propose a novel parser architecture, but rather adapts and thoroughly evaluates the sequence labeling approach using existing BERT-based pre-trained models for Arabic dependency parsing.

The provided Table 1 offers a comprehensive overview of Arabic syntactic analyzers, grouped primarily by their underlying methodologies: rule-based, hybrid, and neural approaches. Rule-based parsers, such as Recursive Transition Network (RTN), Chart Parser, AGFL Parser, and NooJ-based Analyzer, rely heavily on manually crafted grammatical rules and lexicons. These systems exhibit notable accuracy on controlled and limited sentence sets (85.6%–95%), yet they tend to struggle with linguistic coverage, robustness, and scalability to more complex or diverse texts. Hybrid approaches, including ARSYPAR, the Industrial-Strength Parser, Probabilistic Parser, and Bel-Arabi, integrate statistical or machine learning methods with linguistic rules. These parsers generally achieve intermediate levels of accuracy (82%–90%) and show enhanced robustness and broader linguistic coverage compared to purely rule-based methods. However, their performance is contingent upon annotated corpora and careful feature engineering, thus posing challenges in adaptability and maintenance. Neural network-based parsers, such as Camel Parser, AraBERT variants, and Deep-Learning Parsers utilizing transformer architectures, currently deliver state-of-the-art results (LAS and UAS typically ranging from 80% to over 90%). These models benefit significantly from extensive annotated corpora (PADT, CATiB, ATB) and demonstrate superior handling of Arabic morphology, syntactic ambiguity, and out-of-vocabulary words. Nonetheless, neural models require substantial computational resources and large annotated datasets, and they may face performance issues when encountering domain shifts or dialectal variations not represented in training data.Overall, these comparisons indicate that while early parsers laid important groundwork, the highest parsing accuracies for Arabic are currently achieved by transformer-based models and other recent neural approaches. While current parsers demonstrate substantial progress, future research directions include addressing domain and dialect adaptability, interpretability of neural models, and overcoming resource limitations through semi-supervised learning and multilingual transfer techniques. Such advancements will further bridge existing gaps and improve parser applicability across varied Arabic language scenarios.

5 Challenges in arabic syntactic analysis

Many of the difficulties in Arabic syntactic analysis are well-known, recent advances in machine learning, computational linguistics, and deep learning bring forth a new set of advanced challenges. These challenges not only stem from the traditional complexities of the language but also from the need to create sophisticated models capable of handling both contemporary and evolving linguistic phenomena. Below are some of the challenges that researchers are facing in Arabic syntactic analysis:

5.1 Unannotated domain-specific data and formalization gaps

While resources like the Penn Arabic Treebank (PATB) exist, they are heavily focused on formal texts and standard written Arabic, such as news articles. As more Arabic data comes from informal domains like social media, blogs, SMS, and chat conversations, syntactic structures in these domains become more difficult to annotate and generalize. These domains often contain non-standard spelling, abbreviations, and internet slang, and their syntax deviates from the rigid structures of MSA. Furthermore, Arabic-language syntactic structures in domain-specific applications (e.g., medical texts, legal documents, technical manuals) often require specialized syntactic theories and rules that current parsers are not equipped to handle. For example, the grammatical norms in technical writing might differ from colloquial speech, and handling these nuances requires more sophisticated annotation schemes that current treebanks and parsing models lack.

TABLE 1 Comparative performance of Arabic syntactic analyzers.

Analyzer	Approach followed	Evaluation results	Corpus size/name
Recursive transition network	Top-down RTN; context-free + pattern rules	85.6 % accuracy	90 Arabic sentences
A'reb	Recursive top-down parser; production rules	–	Not specified
NLTK parser	Rule-based; CFG; recursive-descent	–	Not specified
Chart parser	Top-down chart parser; CFG	94.3% accuracy	70 Arabic sentences
CFG top-down	Recursive-descent CFG	92% verbal, 98% nominal accuracy	150 Arabic sentences
ARSYPAR	Supervised ML (SVM)	F-score 84.38%	Arabic Treebank subset
Industrial-strength parser	Hybrid (rule-based + statistical)	F-score 82%	300 Arabic sentences
AGFL parser	Rule-based; AGFL formalism	95% successful parses; high ambiguity	200 Arabic sentences
Transducers parser	Finite-state transducers; segmentation + disambiguation	Precision 80%, Recall 90%	200 Arabic sentences
Inductive learning algorithm	Rule induction from examples	92.63% accuracy	Unspecified (unseen sentences)
CFG + classical grammar	CFG plus traditional grammar rules	97% accuracy	200 nominal sentences
NooJ-based analyzer	Rule-based linguistic model	95% syntactic, 86% disambiguation accuracy	120 nominal sentences
Camel parser	BERT + biaffine dependency (ML)	UAS/LAS: 92.4/91.3	Not specified (likely ATB)
Multitask easy-first	Bottom-up, multitask learning	UAS/LAS: 88.08/86.15	CATiB Treebanks
Probabilistic parser	PCFG + property grammar, CYK	Precision 88.3% (verbal), 75.2% (nominal)	400 ATB sentences
Bel-Arabi	Hybrid ML (CRF) + rules	Precision 90.44%	600 sentences
Deep-learning parser	BiLSTM/LSTM/GRU	>99% accuracy	Penn Arabic Treebank
Stanford Arabic parser	PCFG + CYK	FactF1 77.44%, FactDA 84.05%	Penn Arabic Treebank
ArabTAG v2.0	Tree-adjoining grammar; meta-grammar	Precision 88.76% (syntax), 95.63% (semantics)	Not specified
MASAQ	Statistical parser (Random Forest)	Accuracy: 99.0%	MASAQ dataset: 123,565 syntactic functions
Camel-MSA	Fine-tuned BERT-based sequence labeling	UAS/LAS: 83.10/79.17	PADT: 282,384
Camel-MSA	Fine-tuned BERT-based sequence labeling	UAS/LAS: 86.47/85.29	CATiB: 169,319
AraBERTv1	Fine-tuned BERT-based sequence labeling	UAS/LAS: 82.76/78.82	PADT: 282,384
AraBERTv1	Fine-tuned BERT-based sequence labeling	UAS/LAS: 86.76/85.57	CATiB: 169,319
AraBERTv2	Fine-tuned BERT-based sequence labeling	UAS/LAS: 84.03/80.26	PADT: 282,384
AraBERTv2	Fine-tuned BERT-based sequence labeling	UAS/LAS: 87.54/86.41	CATiB: 169,319
ARBERT	Fine-tuned BERT-based sequence labeling	UAS/LAS: 80.37/76.11	PADT: 282,384
ARBERT	Fine-tuned BERT-based sequence labeling	UAS/LAS: 78.31/75.95	CATiB: 169,319
Arabic BERT	Fine-tuned BERT-based sequence labeling	UAS/LAS: 80.02/76.52	PADT: 282,384
Arabic BERT	Fine-tuned BERT-based sequence labeling	UAS/LAS: 82.65/80.59	CATiB: 169,319

5.2 Ambiguities in syntactic structures due to ellipsis and zero pronouns

Arabic syntax features phenomena like ellipsis and zero pronouns that introduce ambiguity into sentence structure. These phenomena are particularly common in conversational Arabic and can result in incomplete syntactic structures that require contextual information to resolve. For instance, a sentence like "He went to the market, and she [went] to the store" in English uses an ellipsis, which may be straightforward to resolve in English, but in Arabic, this can be more complex due to the omission of verb phrases or pronouns without clear agreement. Zero pronouns, where the subject or object is omitted from a sentence because it can be inferred from context, add another layer of complexity. Accurately resolving these ellipses and zero pronouns in both MSA and dialectal varieties remains an unsolved challenge in syntactic

parsing, particularly for systems that rely heavily on surface form rather than deeper contextual understanding.

5.3 Model generalization and domain adaptation

One of the most pressing challenges in Arabic syntactic analysis is the generalization of models across domains. While Arabic parsers have become quite effective for general text (e.g., news), they often fail when transferred to specific domains, such as healthcare, finance, or legal documents. Domain-specific vocabulary, sentence structures, and jargon can lead to significant degradation in performance when the models are not adapted properly. Traditional training methodologies that focus on general-purpose data are less effective for domain-specific tasks, and fine-tuning models for specialized domains remains an open area of research.

6 Conclusion and future directions

Arabic syntactic analysis has made significant strides over the past decade, transitioning from rule-based systems to more sophisticated machine learning and neural network models. Despite these advancements, several challenges remain, including handling dialectal variation, resolving ambiguities due to the lack of diacritics, and the need for larger, more diverse annotated datasets. As new systems and approaches are developed, the evaluation of Arabic syntactic analyzers will remain a critical challenge. Establishing more diverse and standardized benchmarks for evaluating Arabic parsers across dialects, genres, and domains is essential for guiding future improvements.

This paper systematically surveys and compares state-of-the-art methods for Arabic syntactic parsing, clearly highlighting the strengths and limitations of existing rule-based, statistical, machine learning, and hybrid approaches. It has also provided a comprehensive evaluation of essential resources, including prominent Arabic syntax treebanks. The comparative insights presented here serve as a foundational reference for researchers seeking to address the inherent complexities of Arabic NLP.

Future research should focus on leveraging advances in transformer-based models, such as multilingual and domain-adaptive language models, to enhance parser robustness across dialects and diverse textual domains. Joint models capable of simultaneously addressing morphological segmentation, POS tagging, and syntactic parsing should be developed to mitigate cascading errors. Additionally, increased efforts toward interpretability in neural systems and richer semantic annotations in Arabic Treebanks will significantly improve downstream NLP applications. Exploring cross-lingual transfer learning and semi-supervised learning techniques will be vital in overcoming current limitations related to the scarcity of annotated data, particularly for dialectal and low-resource Arabic varieties.

In conclusion, while significant progress has been made in Arabic syntactic analysis, ongoing challenges and evolving linguistic phenomena offer ample opportunities for further research. Advances in deep learning, multilingual modeling, and the expansion of dialectal resources are likely to drive the next wave of breakthroughs in the field.

Author contributions

OS: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. AR: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing, Funding acquisition. MH: Conceptualization, Data curation, Investigation, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing. GB: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work was supported by the American University of Bahrain, which covered the publication fees for this research article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

References

Ababou, N., Mazroui, A., and Belehbib, R. (2017). Parsing Arabic nominal sentences using context free grammar and fundamental rules of classical grammar. *Int. J. Intell. Syst. Applic.* 9, 11–24. doi: 10.5815/ijisa.2017.08.02

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). "Farasa: a fast and furious segmenter for Arabic," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations*, 11–16. doi: 10.18653/v1/N16-3003

Abdul-Mageed, M., Elmadany, A. A., and Nagoudi, E. M. B. (2021). ARBERT MARBERT: deep bidirectional transformers for Arabic. *CoRR, abs/2101.01785*. doi: 10.18653/v1/2021.acl-long.551

Abu-Soud, S., Abdelrazaq, D. J., and Awajan, A. (2018). "Distinguishing nominal and verbal Arabic sentences: a machine learning approach," in *ACIT'2017*.

Aho, A. V., Lam, M. S., Sethi, R., and Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., USA.

Al-Daoud, E., and Basata, A. (2009). A framework to automate the parsing of Arabic language sentences. *Int. Arab J. Inf. Technol.* 6, 191–195.

Al-Ghamdi, S., Al-Khalifa, H., and Al-Salman, A. (2021). *Arpot: The classical Arabic poetry dependency treebank*. Journal of King Saud University - Computer and Information Sciences.

Al-Ghamdi, S., Al-Khalifa, H., and Al-Salman, A. (2023). Fine-tuning bert-based pre-trained models for Arabic dependency parsing. *Appl. Sci.* 13. doi: 10.3390/app13074225

Al-qrainy, S., Muaidi, H., and Alkoffash, M. (2012). Context-free grammar analysis for Arabic sentences. *Int. J. Comput. Applic.* 53, 7–11. doi: 10.5120/8399-2167

Alrayzah, A., Alsolami, F., and Saleh, M. (2024). Arafast: Developing and evaluating a comprehensive modern standard Arabic corpus for enhanced natural language processing. *Appl. Sci.* 14:5294. doi: 10.3390/app14125294

Al-Taani, A. T., Msallam, M. M., and Wedian, S. A. (2012). A top-down chart parser for analyzing Arabic sentences. *Int. Arab J. Inf. Technol.* 9, 109–116.

Antoun, W., Baly, F., and Hajj, H. (2020). "AraBERT: Transformer-based model for Arabic language understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (Marseille, France: European Language Resource Association), 9–15.

Aqel, D., AlZu'bi, S., and Hamadah, S. (2019). "Comparative study for recent technologies in Arabic language parsing," in *2019 Sixth International Conference on Software Defined Systems (SDS)* (IEEE), 209–212. doi: 10.1109/SDS.2019.8768587

Bataineh, B., and Bataineh, E. (2009). "An efficient recursive transition network parser for Arabic language," in *Proceedings of the World Congress on Engineering, Vol. 2* (London), 1–3.

Ben Khelil, C., Ben Othmane Zribi, C., Duchier, D., and Parmentier, Y. (2023). Generating Arabic tag for syntax-semantics analysis. *Nat. Lang. Eng.* 29, 386–424. doi: 10.1017/S1351324922000109

Bensalem, R., Haddar, K., and Blache, P. (2023). An Arabic probabilistic parser based on a property grammar. *ACM Trans. Asian Low-Resour. Lang. Inf. Proc.* 22, 1–25. doi: 10.1145/3612921

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., et al. (2018). "The MADAR Arabic dialect corpus and lexicon," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (Miyazaki, Japan: European Language Resources Association (ELRA)).

Bourahma, S., Mbarki, S., Mourchid, M., and Mouloudi, A. (2018). "Syntactic parsing of simple Arabic nominal sentence using the NooJ linguistic platform," in *Communications in Computer and Information Science* (Springer International Publishing), 244–257. doi: 10.1007/978-3-319-73500-9_18

Brandt, U., and Walter, H. (2001). The cocke-younger-kasami algorithm (revised). *Bull. EATCS* 74, 193–228.

Dozat, T., and Manning, C. D. (2016). Deep biaffine attention for neural dependency parsing. *CoRR, abs/1611.01734*.

Dozat, T., Qi, P., and Manning, C. D. (2017). "Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task," in *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (Vancouver, Canada: Association for Computational Linguistics), 20–30. doi: 10.18653/v1/K17-3002

El-Khair, I. A. (2016). 1.5 billion words Arabic corpus. *CoRR, abs/1611.04033*.

Elshabrawy, A., AbuOdeh, M., Inoue, G., and Habash, N. (2023). "CamelParser2.0: a state-of-the-art dependency parser for Arabic," in *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*. doi: 10.18653/v1/2023.arabicnlp-1.15

Green, S., and Manning, C. D. (2010). "Better Arabic parsing: baselines, evaluations, and analysis," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* (Beijing, China: Coling 2010 Organizing Committee), 394–402.

Habash, N., AbuOdeh, M., Taji, D., Faraj, R., El Gizuli, J., and Kallas, O. (2022). "Camel treebank: An open multi-genre Arabic dependency treebank," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (Marseille, France: European Language Resources Association), 2672–2681.

Habash, N., Rambow, O., and Roth, R. (2009). "Mada+tokan: a toolkit for Arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*.

Habash, N., and Roth, R. (2009). "Catib: The columbia Arabic treebank," in *Annual Meeting of the Association for Computational Linguistics. doi: 10.3115/1667583.1667651*

Habash, N. Y. (2010). *Introduction to Arabic Natural Language Processing*. Cham: Springer International Publishing. doi: 10.1007/978-3-031-02139-8

Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Pajas, P., Štěpánek, J., et al. (2006). "Prague dependency treebank 2.0," in *LDC Catalog No.: LDC2006T01* (Linguistic Data Consortium).

Hajič, J., Smrž, O., Zemánek, P., Pajas, P., Šnaidauf, J., Beška, E., et al. (2004). "Prague Arabic dependency treebank 1.0," in *LDC2004T23* (Linguistic Data Consortium).

Halabi, D., Fayyoumi, E., and Awajan, A. A. (2020). I3rab: a new Arabic dependency treebank based on Arabic grammatical theory. *Trans. Asian Low-Resour. Lang. Inf. Proc.* 21, 1–32. doi: 10.1145/3472295

Hamed, I., Sabty, C., Abdennadher, S., Vu, N. T., Solorio, T., and Habash, N. (2025). "A survey of code-switched Arabic NLP: Progress, challenges, and future directions," in *Proceedings of the 31st International Conference on Computational Linguistics* (Abu Dhabi, UAE: Association for Computational Linguistics), 4561–4585.

Hammouda, N. G., and Haddar, K. (2018). Parsing Arabic nominal sentences with transducers to annotate corpora. *Comput. Sist.* 21, 647–656. doi: 10.13053/cys-21-4-2867

Ibrahim, M., Mahmoud, N., and El-Reedy, D. (2016). Bel-Arabi: advanced Arabic grammar analyzer. *Int. J. Soc. Sci. Human.* 6, 341–346. doi: 10.7763/IJSSH.2016.V6.669

Inoue, G., Alhafni, B., Baimukan, N., Bouamor, H., and Habash, N. (2021). "The interplay of variant, size, and task type in Arabic pre-trained language models," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Online)* (Association for Computational Linguistics).

Kankanampati, Y., Roux, J., Tomeh, N., Taji, D., and Habash, N. (2020). "Multitask easy-first dependency parsing: Exploiting complementarities of different dependency representations," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2497–2508. doi: 10.18653/v1/2020.coling-main.225

Khoufi, N., Aloulou, C., and Belguith, L. H. (2013). "Arsypar: a tool for parsing the Arabic language based on supervised learning," in *The International Arab Conference on Information Technology*.

Maalej, R., Khoufi, N., and Aloulou, C. (2021). "Parsing Arabic using deep learning technology," in *Tunisian-Algerian Joint Conference on Applied Computing*.

Maamouri, M., Bies, A., Buckwalter, T., and Jin, H. (2004). *Arabic treebank: Part 2 v 2.0*. ISBN 1–58563-282–1.

Maamouri, M., Bies, A., Buckwalter, T., and Jin, H. (2005). *Arabic treebank: Part 1 v 3.0*. ISBN 1–58563-330–5.

Mokh, N., Dakota, D., and KÃbler, S. (2024). "Out-of-domain dependency parsing for dialects of Arabic: a case study," in *Proceedings of The Second Arabic Natural Language Processing Conference*, 170–182. doi: 10.18653/v1/2024.arabicnlp-1.16

Nagoudi, E. M. B., Belkebir, A., Maghraoui, N., Elasri, Z., and El-Haj, M. (2022). "Arat5: text-to-text transformers for Arabic language generation," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 492–507. doi: 10.18653/v1/2022.acl-long.47

Nagoudi, E. M. B., Elmadany, A. A., Abdul-Mageed, M., Alhindi, T., and Cavusoglu, H. (2020). Machine generation and detection of Arabic manipulated and fake news. *CoRR, abs/2011.03092*.

Othman, E., Shaalan, K., and Rafea, A. (2003). "A chart parser for analyzing modern standard Arabic sentence," in *Workshop on Machine Translation for Semitic Languages: Issues and Approaches*.

Ouersighni, R. (2014). Robust rule-based approach in Arabic processing. *Int. J. Comput. Applic.* 93, 31–37. doi: 10.5120/16269-6001

Qarah, F. (2024a). Egybert: a large language model pretrained on egyptian dialect corpora. *arXiv preprint arXiv:2408.03524*.

Qarah, F. (2024b). Saudibert: a large language model pretrained on saudi dialect corpora. *arXiv preprint arXiv:2405.06239*.

Redjaimia, A., Strebkov, D., Hilal, N., and Skatov, D. (2014). "The experience of building industrial-strength parser for Arabic," in *Computational Linguistics and Intelligent Technologies*, 668–680. doi: 10.13140/RG.2.1.3692.5606

Riabi, S., Mahamdi, M., and Seddah, D. (2023). "Enriching the narabizi treebank: a multifaceted approach for dialectal arabizi processing," in *Proceedings of LAW-XVII 2023*.

Sawalha, M., Alshargi, F., Yagi, S., AlShdaifat, A. T., and Hammo, B. (2025b). "MASAQ parser: A fine-grained MorphoSyntactic analyzer for the Quran," in *Proceedings of the New Horizons in Computational Linguistics for Religious Texts* (Abu Dhabi, UAE: Association for Computational Linguistics), 67–75.

Sawalha, M., Al-Shargi, F., Yagi, S., AlShdaifat, A. T., Hammo, B., Belajeed, M., et al. (2025a). Morphologically-analyzed and syntactically-annotated quran dataset. *Data Brief* 58:111211. doi: 10.1016/j.dib.2024.111211

Shaalan, K., and Khaled (2010). "Rule-based approach in Arabic natural language processing," in *The International Journal on Information and Communication Technologies (IJICT)*, 3.

Shatnawi, M., and Belkhouche, B. (2012). *Parse trees of Arabic sentences using the natural language toolkit*. College of IT, UAE University, Al Ain.

Suárez, P. O., Sagot, B., and Romary, L. (2020). "Oscar: a multilingual dataset for language modeling, translation and linguistic studies," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 23–31.

Taji, D., Habash, N., and Zeman, D. (2017). "Universal dependencies for Arabic," in *Proceedings of the Third Arabic Natural Language Processing Workshop* (Valencia, Spain: Association for Computational Linguistics), 166–176. doi: 10.18653/v1/W17-1320

Tendeau, F. (1997). "An Earley algorithm for generic attribute augmented grammars and applications," in *Proceedings of the Fifth International Workshop on Parsing Technologies* (Boston/Cambridge, Massachusetts, USA: Association for Computational Linguistics), 199–209.

Tratz, S. (2016). *Arl Arabic dependency treebank*. Technical report, US Army Research Laboaratory Adelphi United States. doi: 10.21236/AD1003943

Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., et al. (2013). *Ontonotes release 5.0*. Technical Report LDC2013T19, Linguistic Data Consortium, Philadelphia, PA.

Xu, Y., Hu, L., Zhao, J., Qiu, Z., Xu, K., Ye, Y., et al. (2025). A survey on multilingual large language models: corpora, alignment, and bias. *Front. Comput. Sci.* 19:1911362. doi: 10.1007/s11704-024-40579-4

Zaki, Y., Hajjar, H., Hajjar, M., and Bernard, G. (2016). "A survey of syntactic parsers of Arabic language," in *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, 1–10. doi: 10.1145/3010089.3010116

Zeroual, I., Goldhahn, D., Eckart, T., and Lakhouaja, A. (2019). "Osian: open source international Arabic news corpus - preparation and integration into the clarin-infrastructure," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, 175–182. doi: 10.18653/v1/W19-4619

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 04 September 2025
DOI 10.3389/frai.2025.1639147

OPEN ACCESS

EDITED BY
Shadi Abudalfa,
King Fahd University of Petroleum and Minerals, Saudi Arabia

REVIEWED BY
Miodrag Zivkovic,
Singidunum University, Serbia
Judy Simon,
SRM Arts and Science College, India
Nourredine Oukas,
University of Bouira, Algeria
Aadil Ganie,
Universitat Politècnica de València, Spain

*CORRESPONDENCE
Fawaz S. Al-Anzi
✉ fawaz.alanzi@ku.edu.kw

RECEIVED 01 June 2025
ACCEPTED 11 August 2025
PUBLISHED 04 September 2025

CITATION
Al-Anzi FS and Sundaram Thankaleela BS (2025) Arabic speech recognition model using Baidu's deep and cluster learning.
Front. Artif. Intell. 8:1639147.
doi: 10.3389/frai.2025.1639147

Arabic speech recognition model using Baidu's deep and cluster learning

Fawaz S. Al-Anzi* and Bibin Shalini Sundaram Thankaleela

Department of Computer Engineering, College of Engineering and Petroleum, Kuwait University, Kuwait

This study involves extracting the spectrum from the Arabic raw, unlabeled audio signal and producing Mel-frequency cepstral coefficients (MFCCs). The clustering algorithm groups the retrieved MFCCs with analogous features. The K-means clustering technique played a crucial role in our research, enabling the unsupervised categorization of unlabeled Arabic audio data. Employing K-means on the extracted MFCC features allowed us to classify acoustically similar segments into distinct groups without prior knowledge of their characteristics. This initial phase was crucial for understanding the inherent diversity in our diverse sampled dataset. Dynamic Time Warping (DTW) and Euclidean Distance are utilized for illustration. Classification algorithms such as Decision Tree, eXtreme Gradient Boosting (XGBoost), K-Nearest Neighbors (KNN), and Random Forest are used to classify the various classes obtained based on clustering. This study also demonstrates the efficacy of Mozilla's Deep Speech framework for Arabic speech recognition. The core component of deep speech is its neural network architecture, which consists of multiple layers of Recurrent Neural Networks (RNNs). It strives to comprehend the intricate patterns and interactions between spoken sounds and their corresponding textual representations. The clustered labeled Arabic audio dataset, along with transcripts and Arabic Alphabets, is used as input to Baidu's Deep Speech model for training and testing purposes. PyCharm, in conjunction with Python 3.6, is used to build a Dockerfile. Creating, editing, and managing Dockerfiles within PyCharm's IDE is simplified by its functionality and integrated environment. Deep speech provides an eminent Arabic speech recognition quality with reduced loss, word error rate (WER), and character error rate (CER). Baidu's Deep Speech intends to achieve high performance in both end-to-end and isolated speech recognition with good precision and a low word rate and character error rate in a reasonable amount of time. The suggested strategy yielded a loss of 276.147, a word error rate of 0.3720, and a character error rate of 0.0568. This technique increases the accuracy of Arabic automatic speech recognition (ASR).

KEYWORDS

clustering, language model, acoustic model, Baidus deep speech, RNN, deep learning

1 Introduction

Speech acts as a gateway in communicating our ideas through different vocal sounds and is a powerful tool that shapes our world. The study of speech signals and the techniques used to process them is known as speech processing. Modern automatic speech recognition (ASR) systems replace the conventional human–machine interface in various commercial applications. Through the application of linguistics and computer science, ASR systems can interpret spoken words and translate them into text. This enables voice-activated

device interaction, message dictation, and generation of transcripts from recordings. Recent developments in artificial intelligence (AI), particularly natural language processing (NLP), have focused on using AI applications for ASR. Researchers have investigated morphological analysis, resource building, and machine translation for the Arabic language. Speech and language disorders are a side effect of many diseases, and devices like the Servox Digital Electro-Larynx (EL) can generate quasi-clear voices for people with disorders (Mohammed Ameen and Abdulrahman Kadhim, 2023). The respiratory, phonatory, and articulatory end organs are all involved in the intricate neurological process of speech (Musikic et al., 2025). Acoustic media and background noise can disrupt and interfere with speech communication. Vocalization system damage can affect the efficiency of voice recognition and voice clarity (Liu et al., 2025). ASR is useful in many domains, including the development of accessible applications to transforming human–machine communication. Speech recognition automatically identifies and translates a person's spoken words based on the data available in a speech waveform and its historical data feed. The evolution of deep learning has changed the ASR landscape in conjunction with Recurrent Neural Network (RNNs), deep neural networks (DNNs), and convolutional neural networks (CNNs). Deep neural networks are multilayered artificial intelligence that learns from data. They are inspired by the structure of the human brain, and these layers enable them to handle challenging issues. Deep neural networks, which have been trained on enormous datasets, modify their internal connections to identify patterns and carry out tasks such as speech translation and image recognition. The ability of CNNs to extract intricate patterns from audio input has been inspiring. Baidu's Deep Voice enhances voice recognition precision in noisy situations, as well as in far-field and reverberant conditions (Ahmed and Ghabayen, 2017; Masterson, 2015). MFCCs effectively decipher sound content in speech and audio processing. The MEL scale considers how our ears interpret pitch and frequencies with similar sounds. Applications such as speech recognition systems can interpret speech data by evaluating MFCCs. A clustering algorithm is a specific set of instructions that tells a computer how to automatically group data points into clusters. The study addresses the issue of unlabeled Arabic audio data by applying an unsupervised clustering algorithm to analyze and structure the corpus, uncovering acoustic patterns, speaker variabilities, and environmental conditions. These insights inform effective data handling strategies and the training of Arabic Deep Speech ASR models. These algorithms are used in unsupervised learning, where the data does not have predefined labels. There are many clustering algorithms, but one of the popular popular ones is K-means. Algorithms such as Hierarchical clustering, Mean shift clustering, Gaussian mixture model, Affinity propagation, and K-means clustering are widely available to group different patterns of MFCCs (Al-Anzi and Shalini, 2024).

The primary objective of this study is to develop an ASR system that automatically transcribes spoken utterances into a textual format. Our approach utilized a database consisting of Arabic audio recordings, which encompassed news broadcasts, public speeches, and various general recordings of individuals. The primary objective of our study is to extract the Mel-frequency coefficients necessary for ASR from the unlabeled Arabic audio dataset. We employed a clustering approach, with the clusters organized according to the KNN algorithm to label the collected MFCCs. The retrieved MFCCs are categorized according to their auditory characteristics. We have utilized Baidu's Deep Speech model to transcribe spoken language into text. The input given to the model is our clustered Arabic audio dataset along with its transcribe and alphabet. We also assessed the word error rate (WER) and character error rate (CER) of the transcribed results from the audio datasets. We have labeled the clustered dataset using a speech recognition pretrained model from the klaam library, categorizing it as Modern Standard Arabic (MSA), Egyptian Arabic (EGY), and Gulf Arabic (GLF) based on dialects. We have trained the model using different machine learning algorithms to categorize the dialects and assess accuracy, loss, and evaluation metrics for the clustered results.

The subsequent sections of the article are structured as follows: A concise literature overview encompassing ASR, diverse languages and accents in ASR, end-to-end speech processing, and the deep learning architectures that facilitate speech recognition, concluding with a clearly defined research gap, along with the methodologies and materials. Includes fundamental architecture, data collection, data analysis, MFCC analysis, clustering of MFCC characteristics, classification, performance evaluation, findings, debates, conclusion, and future scope.

2 Literature review

The study by Ahmed and Ghabayen (2017) proposes three methods to improve Arabic automatic speech recognition. They are listed in the following order: utilizing a Decision Tree to generate alternative pronunciations, modifying a native acoustic model with a different native model, and text processing to improve the language model. By employing these methods, the word error rate was reduced. The methodology of the paper showed how deep speech recognition models can integrate over time with long, adjustable windows (Ahmed and Ghabayen, 2017).

2.1 Automatic speech recognition

In the study by Keshishian et al. (2021), ASR aims to enable computers to identify and interpret human speech as accurately as possible. Many techniques can be used to implement speech recognition models. The author utilized one of the newest techniques for speech recognition, which employs neural networks with deep learning. An overview of the research conducted on Arabic voice recognition is given in the paper by Wlgihab et al. It also sheds some light on the facilities and toolkits available for Arabic voice recognition system development (Algihab et al., 2019). A vast array of products has been developed that efficiently leverage ASR to enable communication between humans and machines by Karpagavalli and Chandra et al. Speech recognition applications exhibit reduced performance in the presence of reverberation or minimal background noise (Karpagavalli and Chandra, 2016). Both acoustic and text transcriptions are used during the entire training process of ASR neural network systems.

The study by Belinkov et al. compares phonemes and graphemes along with different articulatory properties to evaluate the representation quality across a range of classification tasks. The study analyzes three datasets and two languages, Arabic and English, and demonstrates how consistently different features are represented across deep neural network covers (Belinkov et al., 2019). The purpose of the study by Abdul et al. is to discuss the applications of the MFCC as well as certain problems with its calculation and how they affect the model's performance (Abdul and Al-Talabani, 2022). An enhanced Mel-frequency cepstral coefficients (MFCC) feature for unsupervised marine target clustering is presented in the research. It exhibits a high success rate for multitarget or depth-target clustering as well as strong anti-interference capabilities (Yang and Zhou, 2018). The Short-Time Fan-Chirp Transform (FChT), a novel technique for time-frequency analysis of speech signals, is presented in this study (Képesi and Weruaga, 2006). It enhances spectral and time-frequency representation, making it appropriate for filtering applications. Taking contextual considerations into account, this method examines speech processing to quantify controllable speech features across a variety of talker populations, noise levels, competing speakers, and the channel through which it is conveyed (Pitton et al., 1996).

The study by Abushariah et al. gave a framework for designing a speaker-independent automatic Arabic speech recognition system using a phonetically rich speech corpus. The system uses Carnegie Mellon University's Sphinx tools and Cambridge HTK tools and uses three-emitting state Hidden Markov Models for tri-phone-based acoustic models. The system achieved word recognition accuracy of 92.67 and 93.88% for similar speakers with different sentences, and a Word Error Rate of 11.27 and 10.07% with and without diacritical marks (Abushariah et al., 2012). A simple word decomposition algorithm presented by Afify et al. requires a text corpus and affix list, improving WER by 10% in Iraqi Arabic ASR. The algorithm also reduces WER by 13% relative (Afify et al., 2006). The research presented by Ali Ahamed et al. shows a novel methodology for assessing ASR in languages lacking a standardized orthographic system. The authors solicited five distinct users to transcribe speech segments, subsequently integrating the alignments from numerous references and presenting a revised WER. The findings indicated an average WER of 71.4 and 80.1%, respectively.

2.2 Different languages, ascent speech recognition

To build high-performing recognizers for two radically different languages, such as Mandarin and English, the authors Amodei et al. looked into a variety of network topologies and found a few helpful techniques, such as look-ahead convolution for unidirectional models, and enhanced numerical optimization using SortaGrad and Batch Normalization (Amodei et al., 2016). In the study by Nahid et al., they investigated the capacity of the DeepSpeech network to recognize unique Bengali speech samples. Recurrent Long Short-Term Memory (LSTM) layers form the foundation of this network, which models internal phoneme representations. At the bottom, convolutional layers are added, which removes the requirement to assume anything about internal phoneme alignment. The model was trained using a connectionist temporal classification (CTC) loss task, and the transcript was generated by casting a beam search decoder. On the Bengali real number speech dataset, the developed method produced a lower word error rate and a character error rate (Nahid et al., 2019).

In the study by Priyank Dubey (2023), they discussed that the transcription of spoken speech can be extracted from the waveform using ASR. Mozilla Deep Speech is among the most recent, according to Baidu's Deep Speech research report. Through end-to-end deep learning, the state-of-the-art deep voice recognition system was developed. A properly optimized RNN is used with several Graphical Processing Units (GPUs). Its generalizability to other English accents is limited because American English accents make up the majority of the datasets used in this training. In this study, researchers used the most recent Deep Voice model, Deep Speech-0.9.3, to create an Indian-English speech recognition system from beginning to end for dialects. In the study by Xu et al. (2020), the focus of the research was on a real-time German speech-to-text system that was constructed using numerous German language datasets. Researchers in this study optimized DeepSpeech for teaching a current German speech-to-text prototype by combining multiple German datasets. Moreover, they achieved strong WER rates. The model discussed in the study by Ai-Zaro et al. produces the WER/PER of 3.11 and 6.18% (Al-Zaro et al., 2025).

Literature (Iakushkin et al., 2018) explains how a voice recognition system for the Russian language is made using DeepSpeech. The foundation was the Mozilla Corporation's DeepSpeech English implementation, which is available as open-source software. The system was trained in a containerized environment using Docker technology. A dataset of Russian literary audio recordings made available on voxforge.com was used, and the best WER was 18%. A study by Messaoudi et al. (2021) proposes an end-to-end method for building Tunisian language communication systems based on deep learning. The paired text-speech dataset in the Tunisian dialect created for this proposal is called "TunSpeech." Furthermore, the current Modern Standard Arabic (MSA) speech data were combined with dialectal Tunisian speech data to lower the Out-of-Vocabulary rate.

2.3 End-to-end speech processing

Research (Kim et al., 2017) offers a novel end-to-end speech recognition method that leverages a hybrid CTC-attention model within a multitask learning framework to boost resilience and accelerate convergence, thereby reducing the alignment issue. An experiment using the WSJ and CHiME-4 tasks demonstrates its superiority over the CTC and attention-based encoder-decoder baselines, yielding 5.4–14.6% relative improvements in CER. The study by Agarwal and Zesch (2020) utilizes a shared task on SwissText/KONVENS for a speech-to-text system. A neural network is trained end to end, using Mozilla DeepSpeech as its foundation. Data augmentation, post-processing, and transfer learning from standard English and German were utilized. The WER generated by the system is 58.9%.

2.4 Speech recognition using deep learning

In the study by Nedal Turab (2014), a neural network technique was used to address phoneme recognition. Gaussian low-pass filtering produced improved voice signal quality and reduced noise, which was then used to train a neural network for system training. Study (Alrumiah and Al-Shargabi, 2023) tackles the important task of identifying classic Arabic speech for the 1.9 billion Muslims who recite the Quran. It proposes a model based on Deep Neural Networks (DNNs). With a 19.43% word error rate and a 3.51% character error rate, RNN-CTC outperformed the other models following its training on a 100-h dataset of Quran recordings. CNN was used to further reduce the word error rate. Paper (Alsayadi et al., 2021) presents Arabic diacritical mark-based ASR systems. To create a trustworthy and accurate Arabic ASR, a study by Alsayadi et al. looks at the application of cutting-edge end-to-end deep learning techniques. The acoustic characteristics used in these methods are the log Mel-Scale Filter Bank energies and the Mel-frequency cepstral coefficients. Enhancing discretized Arabic ASR is possible with CNN-LSTM and a new CTC-based ASR. When it comes to Arabic voice recognition, CNN-LSTM with a consideration basis outperforms both traditional ASR and the Joint CTC-attention ASR context (Alsayadi et al., 2021). The research by Ullah et al. utilized Arabic image datasets that have been gathered, consisting of 2,000 Arabic digit records and 900 Arabic phrase records from 24 native speakers. VGG-19 is a deep convolutional neural network with 19 weight layers and is used in this study to extract visual characteristics. Two different approaches, namely, the batch-normalized VGG-19 base model and the standard VGG-19 base model, are presented in the study. The test dataset produces the accuracy of 93% digit and phrase recognition, 97% phrase recognition, and 94%-digit acknowledgment rates (Ullah et al., 2022).

Nagamine et al. analyze a sigmoid DNN trained for a phoneme recognition task to characterize different aspects of the non-linear changes that occur in hidden layers. The more separable phone instances are handled by deeper layers of the network through a non-linear feature space transformation. The study describes how a deep neural network model learns by transforming the feature space in a non-uniform way through repeated non-linear transformations (Nagamine et al., 2016). In the study by Hori et al. (2018), researchers investigate the impact of word-based RNN philological mockups language models (RNN-LMs) on end-to-end ASR performance. It includes a novel word-based RNN-LM which allows decoding with only word-based. Low WER is achieved by the proposed model for the WSJ Eval'92 test set. In the study by Dendani et al. (2020), the representational characteristics of a DNN trained for phoneme recognition were described. In the first hidden layer, node selectivity to specific articulation styles and locations appeared, and in the deeper layers, this selectivity became more pronounced. In the study by Dendani et al. (2020), ASR is implemented using a Deep Auto Encoder (DAE). The results showed that the enhanced speech's accuracy was about 3.17 times better than the accuracy estimated before. Recent models and algorithms, such as Mozilla Deep Speech, have been developed, but their generalizability is limited due to their use of American–English accent datasets (Priyank Dubey, 2023). The study by Srivathshan et al. proposes a hybrid Active Noise Cancellation (ANC) system that combines Secondary-Path Filtered Active Noise Control (SF-ANC) and a Fuzzy Adaptive Neuro-Fuzzy Inference System (FxANFIS) to improve noise reduction performance (Srivathshan et al., 2025).

2.5 Research gap

We haven't found any specific results from my more targeted searches for studies that directly combine Baidu's Deep Speech with cluster learning for Arabic speech recognition. Research on combining Baidu's Deep Speech and cluster learning for Arabic speech recognition has not yielded specific results, suggesting a lack of extensive exploration. However, studies using Deep Speech and cluster learning techniques have revealed challenges like language complexity and data limitations. This supports the hypothesis that this specific combination may not yet have been thoroughly investigated by researchers.

3 Methods and materials

The unlabeled Arabic audio dataset, along with the alphabet, is applied in the proposed work. The auditory data are converted and then hooked onto a sequence of probabilities spanning the characters in the alphabet. Second, this sequence of possibilities gives rise to a cast of characters. The first and second steps are made possible by a Deep Neural Network and an n-gram language model, respectively. The n-gram language model is trained on a text corpus, and the neural network is trained on corresponding text transcripts and audio files. To predict text from speech and prior text, respectively, both the language model and the neural model receive training. Generating (MFCC, Analog to Digital Conversion, Framing, Windowing, Discrete Fourier Transform conversion, Mel-Filter Banks Wrapping Frequency, Converting Mel Filter Banks to Log, Executing Discrete Cosine Transform, the Resultant MFCC Acoustic Model generation, Language Model creation, and Decoding algorithm with deep speech are the fundamental techniques employed in this system. They are all converted to a WAV setup and given a monaural aural canal with a sampling rate of 16,000 Hz and a depth of 16 bits for each value to allow our deep speech pipeline to read all audio clips.

Our unlabeled Arabic audio dataset was subjected to a clustering technique and was mainly used in the pre-processing and data interpretation phases. Since our original dataset was completely unlabeled, we used clustering to characterize acoustic diversity, which involves identifying distinct acoustic groups. The results obtained are manually tested against the transcribed text data. The clustering algorithm enables us to find hidden structures in the data by grouping the MFCC features. The MFCCs are derived from the available Arabic Audio datasets, which are further clustered based on their similar features using clustering algorithms. Machine learning algorithms are further introduced to classify the clusters. The combination of MFCC extraction, clustering, and classification provides an effective framework for extracting insightful information from Arabic speech data. Speech analysis tasks are a good fit for MFCCs because they capture the aspects of

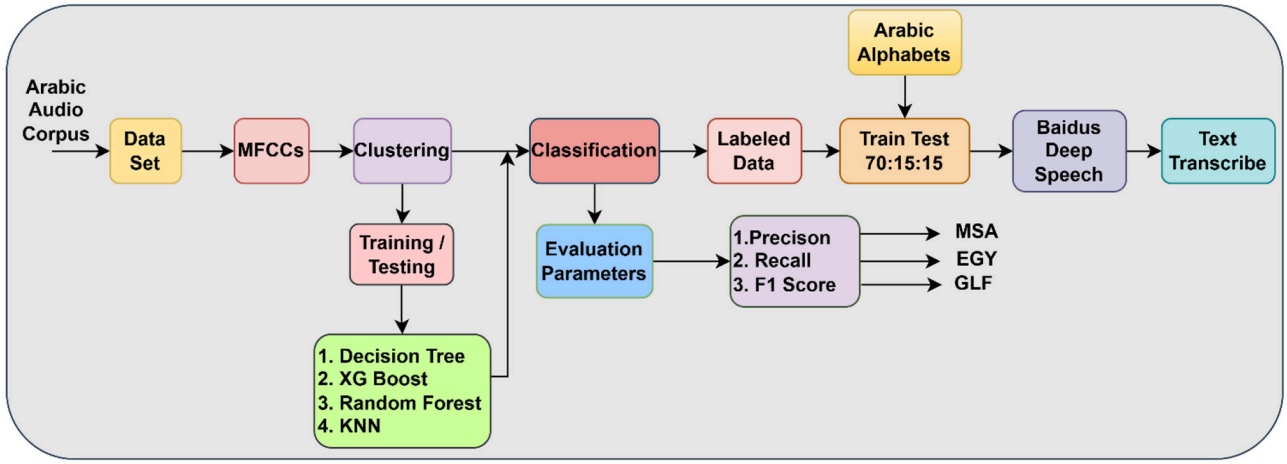

FIGURE 1
Methodology diagram with clustering and Baidu's deep speech.

speech that are perceptible to humans. ASR allows voice-activated computer communication for individuals with physical disabilities. Mozilla's Deep Speech is one of the well-known ASR systems widely accepted and has shown remarkable progress in multiple languages, including Arabic. Baidu's Deep Speech framework is an open-source ASR system that converts spoken words into written language. This speech-to-text technology uses deep learning algorithms to translate spoken language into written text. Acoustic models, language, speech coherence, and performance evaluation are a few components of speech recognition models.

3.1 Methodology

Figure 1 depicts a detailed pipeline for processing Arabic audio data, incorporating both unsupervised and supervised machine learning methods alongside a deep learning model for transcription. The method commences with an Arabic Audio Corpus, which is subsequently input into a dataset preparation phase. MFCCs are recovered from this dataset, functioning as resilient acoustic characteristics. The characteristics subsequently undergo Clustering, wherein an unsupervised algorithm, presumably K-means, categorizes the audio segments according to their acoustic similarities. The speech recognition pretrained model by the klaam library labeled the clustered output as MSA, EGY, and GLF. The efficacy of the classification models is evaluated by metrics such as Precision, Recall, and F1-Score, with distinct results highlighting an emphasis on dialectal performance. The result of this clustering phase initiates a Training/Testing phase for traditional machine learning models, such as Decision Trees, XGBoost, Random Forest, and KNN, employed for a Classification task, presumably aimed at categorizing audio segments based on insights derived from the clustering. The classification outcomes, combined with the "Arabic Alphabets" input, facilitate the generation of labeled data, which is thereafter divided into 70% for training, 15% for testing, and 15% for validation. These annotated data are essential for training Baidu's DeepSpeech model, the fundamental element responsible for the Text Transcribe job, which converts Arabic audio into text. This integrated architecture exemplifies a multifaceted strategy for Arabic speech processing, amalgamating feature engineering, unsupervised learning, conventional classification, and deep learning to provide a holistic solution.

3.2 Architecture of the speech recognition system

Figure 2 shows the architecture of the Speech Recognition System. Deep neural networks are used in speech recognition to translate spoken words into written text. To extract significant acoustic properties, the spoken utterances are first preprocessed. The following steps correspond to the preprocessing, feature extraction phases, decoder, and model creation. The preprocessing block performs various operations on the speech signal, such as noise reduction and silence removal. After the noise reduction, the background noise gets removed. There will not be any background noise in the spoken signal after the preprocessing phase. Scaling the voice signal to a standard magnitude is known as normalization. The speech stream is divided into shorter segments through framing, and these segments typically last 20–30 ms.

The process of extracting information from each voice signal frame is known as feature extraction. The acoustic properties of the voice signal are represented by these features. These characteristics are then applied to a series of models: an audio model forecasts the phoneme sequence, and a dialectal prototypical model uses the analysis of the previous word to predict the next. A decoder transforms the sequence into a string of words, enabling accurate speech-to-text conversion. This process uses a pronunciation dictionary to ensure accurate translation and proper word pronunciation. The retrieved features in the acoustic model, a statistical model, represent a set of phonemes. The language model is a numerical model that forecasts the next verse in a

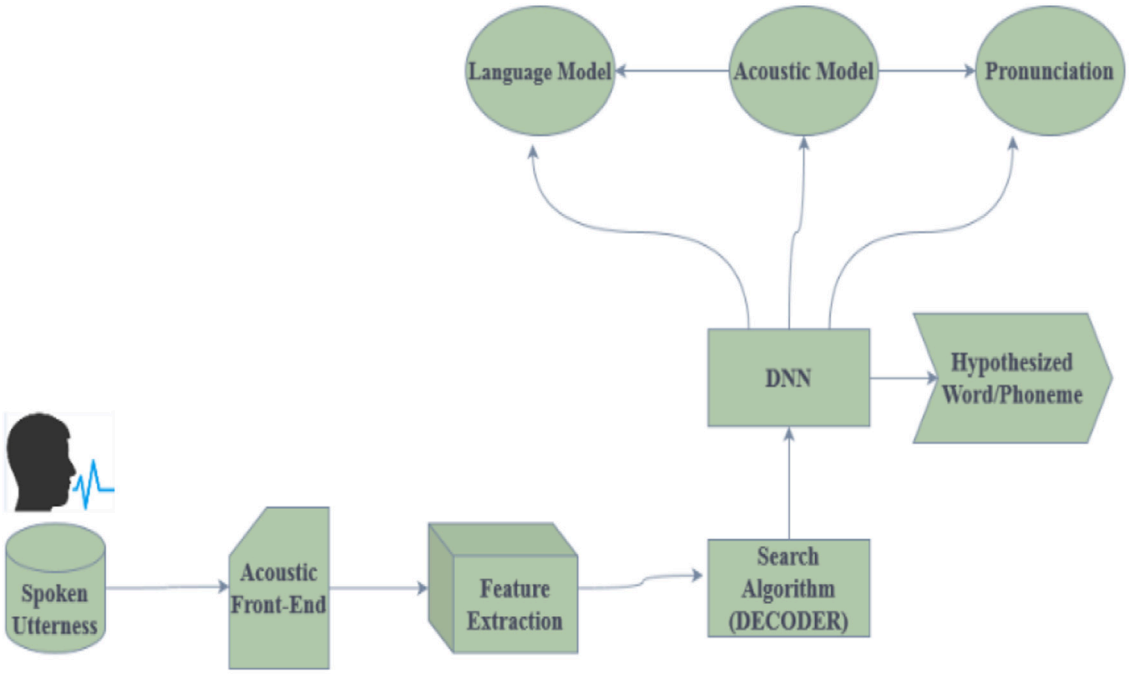

FIGURE 2
Architecture of the speech recognition system.

series based on the verses that have already been spoken. The decoder needs to convert the sequence of phonemes from the acoustic model into a word order. The last block in the diagram represents the word sequences that have been transcribed. A string of words represents spoken speech. Because DNNs can identify complex patterns in data, they are well-suited for voice recognition tasks.

3.2.1 Probability theory for speech recognition

An ASR system's main objective is to infer the acoustic input O in Equation 1, the most likely discrete symbol sequence among all valid sequences in the language L (Rabiner and Juang, 1993).

$$O = o_1, o_2, o_3 \ldots .o_t \tag{1}$$

The symbol sequence to be recognized is N, given in Equation 2:

$$N = n_1, n_2, n_3 \ldots .n_n \tag{2}$$

The fundamental ASR system goal and the probability are given in Equations 3, 4.

$$W = \text{argmax} P(W/O) \tag{3}$$

$$P(W/O) = \frac{P(O/W)}{P(O)} P(W) \tag{4}$$

3.3 Data collection

The Arabic audio dataset is our in-house dataset, which contains 4,071 audio samples from various fields, such as security and justice, Economy, Education, Health, Technology, and Sports. Each heading of data is subdivided into three levels of datasets, such as first, second, and third sets. Deep speech requires mono-channel audio files in WAV format with a sampling rate of 16 kHz and an encoding of 2 bytes per sample for all WAV files, so ensuring consistency in audio quality and format. This collection is categorized by speech type, comprising 733 spontaneous voice files and 588 read speech files, providing a varied representation of natural and controlled verbal expressions. The text linked to these audio recordings has an average length of 93.0 characters, reflecting a moderate complexity and vocabulary range within the collection. Ten to twenty-second passes are available between each voice sample. The more closely we match this, the longer or shorter the model will be. The alphabet.txt file contains a transcription of every character from the given voice clip. From the audio voice clip, all punctuation has been removed, including quotation marks, dashes, and other marks. Three sets of data were separated: test, validation, and training. Diacritical marks are used to show proper pronunciation or to provide phonetic guidance because the standard Arabic script does not provide enough information about pronunciation. Since deep speech operates at the character level, the inclusion of these representations influenced the generation of the acoustic model. Prediction possibilities rise based on the number of letters.

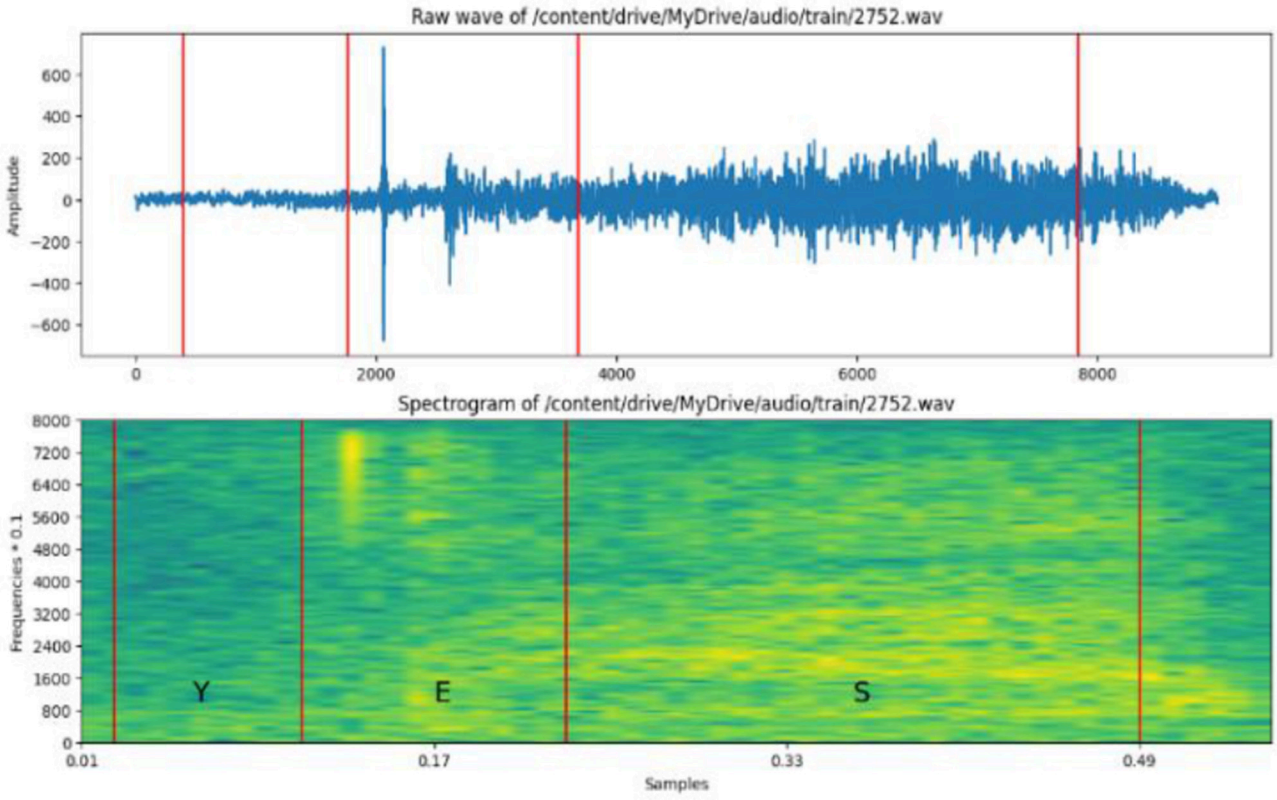

FIGURE 3
Raw and spectrogram of wave signals.

3.4 Data analysis

We have used a sample rate of 1,600 Hz for each audio data. The encoding of each wave file is 2 bytes per sample. Likely, spontaneous speech is used for our analysis. The number of spontaneous speech files is 733, and the number of speech files read is 588. The total number of training files is 1,321. The average text length is 93.0.

3.4.1 Silence removal

Figure 3 shows the signal after noise removal analysis of an Arabic signal. Arabic audio signals must be stripped of silent or low-energy segments by identifying and removing them. The advantages of silence removal include speech analysis for cleared content and improved speech clarity.

3.4.2 Time and frequency analysis of speech

The basic frequency of the vocal cords, which determines whether a voice is perceived as high or low, is referred to as pitch. Rapid alterations in the speech signals linked to consonants and other non-voiced sounds are known as transient features. The time-frequency distribution of the signal is mentioned as the frequency spectrum of the audio signal. The specific characteristics of the spectrum will depend on the speaker's voice, the content of the speech, and the recording conditions. Analyzing spectra gains valuable insights into the acoustic properties of speech signals and is helpful for speech recognition, speaker identification, and language understanding.

3.5 Sampling

Digitalizing the continuous sound wave is necessary for audio signal sampling. We have digitized the sound wave for Arabic audio. To achieve this, the parameters of the sampling rate should be established to determine the frequency of signal measurement. We have used a sampling rate of 44.1 kHz and a bit depth of 16 bits for our Arabic speech for sampling one lengthy audio wave. The overall sampling rate is 16 kHz. Figure 4 shows the sampling frame of the audio signal. Spectra used horizontal and vertical axes to visually represent the energy distribution across time and frequency, respectively. The power of each combination is indicated by the intensity of the color. Common observations include darker areas, which are associated with high energy, and lighter areas, often linked to unvoiced sounds.

3.5.1 Discrete Fourier Transform

The windowed speech signal is subjected to DFT, which yields the signal's phase and magnitude representation. The Fast Fourier Transform (FFT) algorithm transforms time domain analysis to frequency domain analysis Figure 5 shows the FFT spectrum of

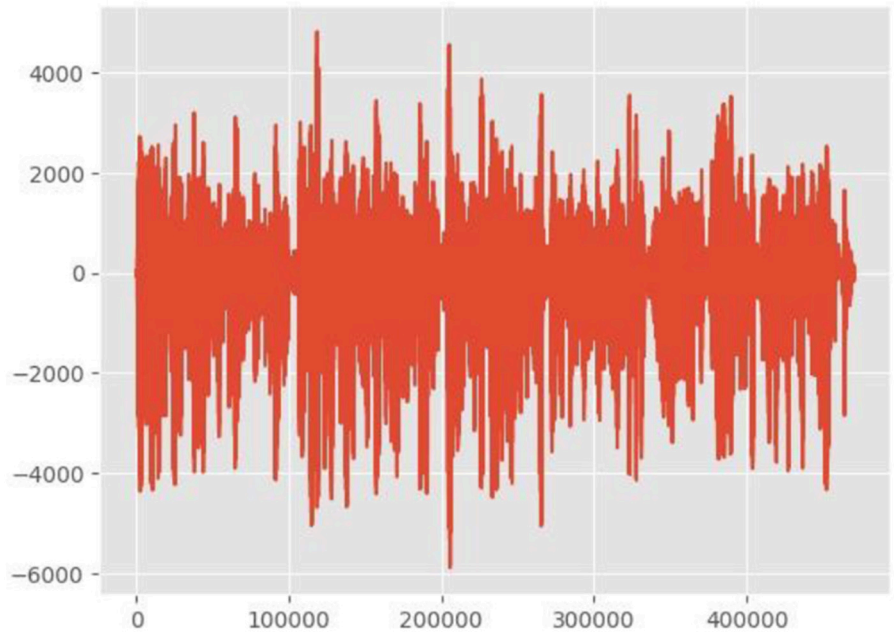

FIGURE 4
Sampling frame of an audio signal.

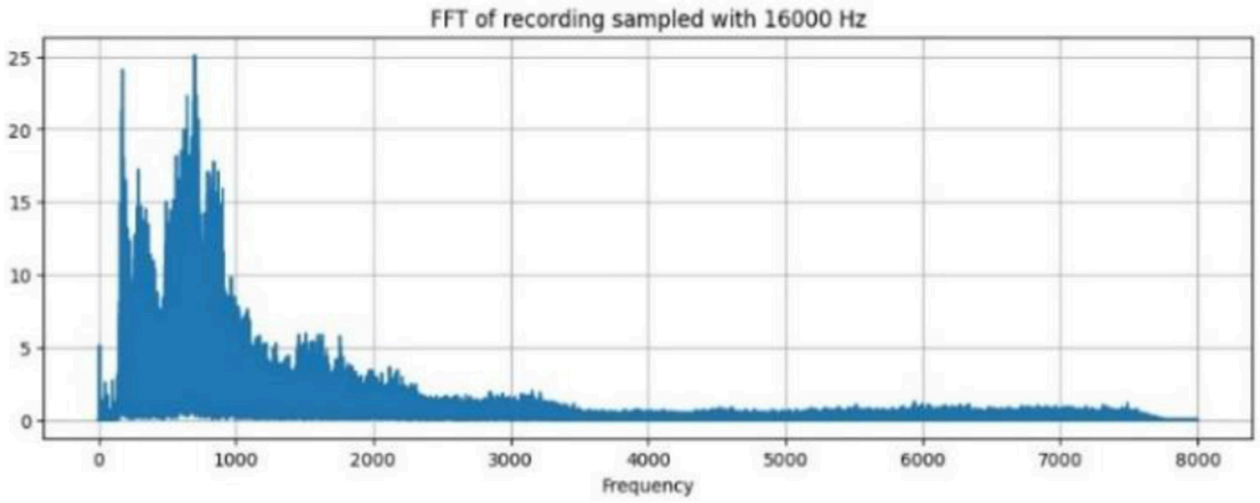

FIGURE 5
FFT recordings of wave.

an audio signal and the distribution of the energy that occurs at different frequencies for each segment. Dominant frequencies are those that indicate prominent tones, such as formants and pitch. The spectral content is used to reveal the presence of various frequency components. The sampling frequency of 1,600 Hz provides basic frequency analysis.

3.5.2 MFCC feature extractions

The process of extracting MFCC features is essential for comprehending speech content, which involves triangular filters. Standard FFTs linearly analyze frequencies of sound, but human hearing operates on a Mel scale. The output of the FFT is passed through triangle-shaped filters. We can capture the portions of the spectrum most pertinent to human hearing by adding the contributions of each filter, each of which focuses on a particular frequency range. The MFCC is the result of this Mel-focused representation. Filters are arranged logarithmically, except above 1,000 Hz, and are equally distributed. The equation used to compute Mel frequency is given in Equation 5 (Gupta et al., 2013).

$$\text{Mel}(f) = 1127\text{In}\left(1 + \frac{f}{700}\right) \tag{5}$$

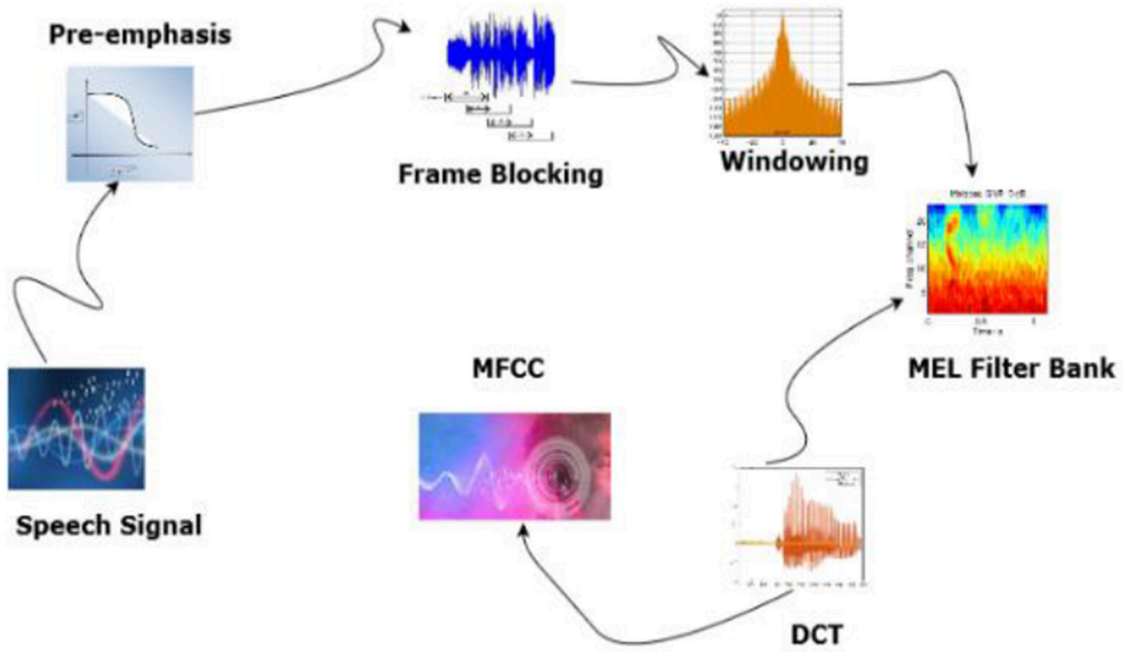

FIGURE 6
MFCC feature extraction.

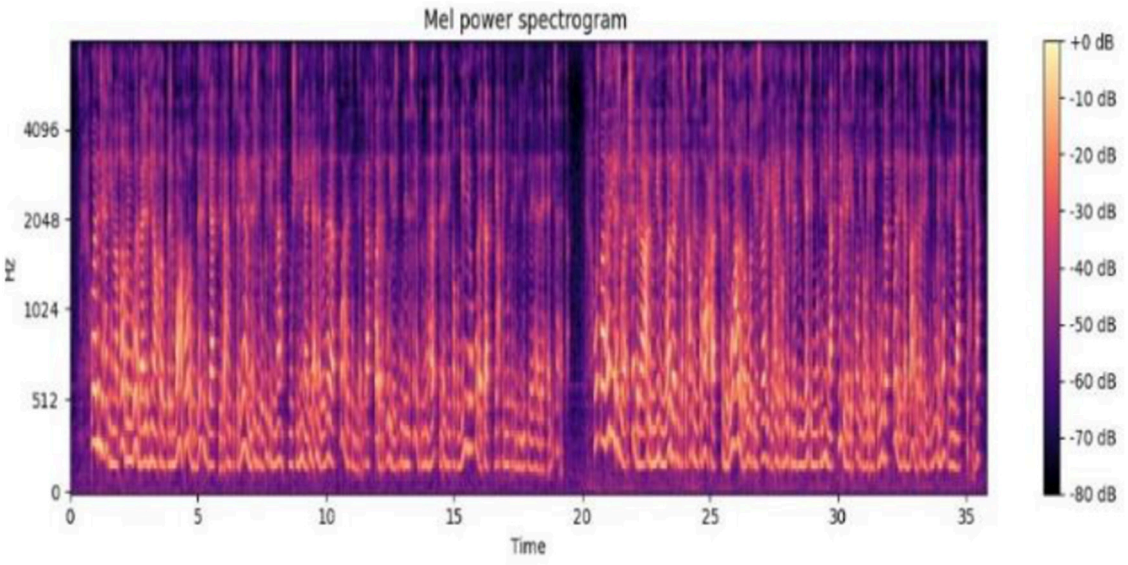

FIGURE 7
Mel spectrogram.

The changes in the speech from frame to frame can be calculated with the first and second MFCC coefficients. Figure 6 shows the block diagram of MFCC feature Extraction.

The audio signal is divided into frames. Windowing and FFT are applied to convert it to the frequency domain. Mel-scale filtering is used in accordance with human auditory perception and logarithmic compression. The discrete Cosine Transform is used to reduce dimensionality, and the resulting MFCCs can provide speaker independence, robustness against noise, and can be processed efficiently. They also capture the fundamental spectral characteristics of speech. Figure 7 shows the Mel power spectrum of the Arabic audio dataset.

TABLE 1 MFCC statistics.

Mean	Standard deviation	Maximum	Minimum
−52.965	8.573	−19.167	−88.341

3.5.3 MFCC statistics

The mean, standard deviation, maximum, and minimum values are represented in Table 1. The mean reveals the average emphasis on the frequency band within the speech. The speech data's standard deviation is a measure of its variability. The maximum and minimum values help in locating anomalies or errors made during the MFCC extraction process. A Discrete Cosine Transform is applied to each MEL filter band to extract MFCCs from the Mel spectrum.

Figure 8 shows the correlation heat map of the different Mel frequency coefficients. The degree of similarity between different MFCCs is shown by their correlation. The various MFCC features are represented by the rows and columns in the heatmap. The correlation between the features that correspond to the row and column is represented by the color of each cell. When two features have a positive correlation, that is, when they tend to rise or fall together, they are colored red. When two features are negatively correlated, one tends to increase while the other decreases, as indicated by blue. When the two features are uncorrelated, the color white is used. Every value on the heatmap's diagonal is 1.0, indicating that every feature has a perfect correlation with every other feature. Higher values indicate stronger correlations. The values of the diagonal range from −1.0 to 1.0. MFCC captures the spectral envelope of audio signals based on the relative prominence of different frequency bands.

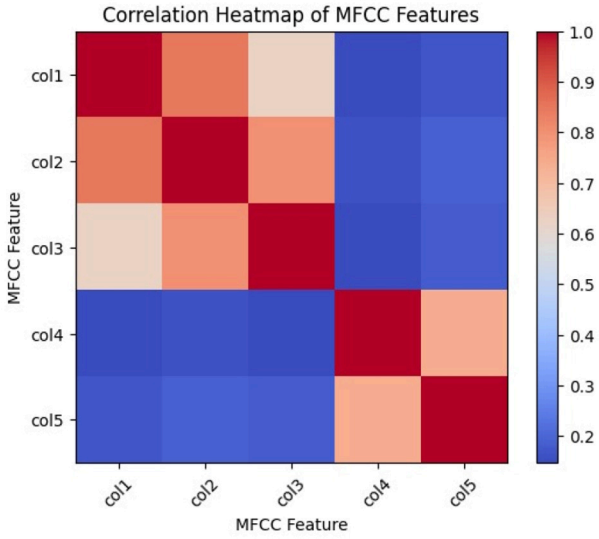

FIGURE 8
Correlation heat map.

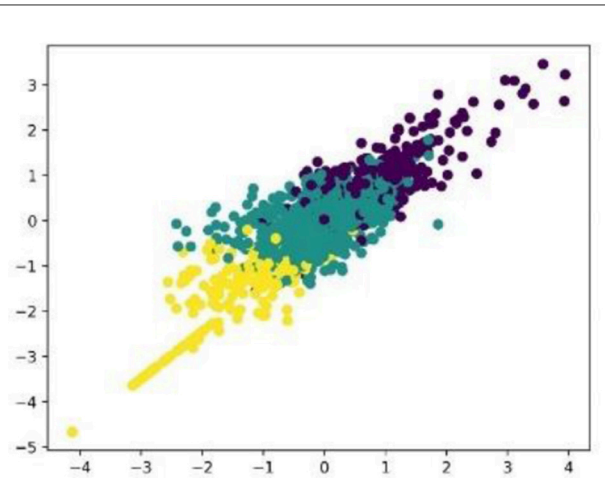

FIGURE 9
Clustering of MFCC features.

4 Clustering and classification

MFCC features are clustered together using a clustering algorithm. As the labels are unknown to us, supervised learning is not a solution to the problem. An unsupervised learning method called K-means clustering will be used for grouping into clusters. The clustering divides data points into a fixed number of groups (K) based on their similarity. The first K data points are chosen at random to serve as the initial cluster centers. The nearest center is determined by averaging these assigned points. Repeating this process until the centers stabilize produces groups in which the data points are unique from those in other clusters and similar to each other within each cluster. Clustering is done based on the Euclidean distance in the MFCC feature space between data points. Three clusters are applied to MFCC features. The clustered data are scaled with a silhouette score. Figure 9 shows the three groups of clusters formed from MFCC correlation features. A silhouette score of 0.6918 was obtained in the clustering. The silhouette score is the metric used to assess the quality of clustering algorithms. It evaluates how well data points are assigned to their clusters. Scores range from −1 to 1, with values closer to 1 indicating improved clustering.

4.1 Grid search

In machine learning, grid search is a technique used to determine a model's optimal settings, also known as hyperparameters. Each hyperparameter has a specific range, and the model is trained using all possible combinations from the different ranges. The performance of each combination is assessed, and the best combination is selected as an ideal set. Grid search CV finds the optimal solution based on the selected metric.

4.2 Classification

For multiclass classification tasks, the support vector machine classifier is used. A hyperparameter tuning method called grid

search is used to maximize the performance of the SVM model. "Linear" and "rbf" for kernel and (Mohammed Ameen and Abdulrahman Kadhim, 2023; Belinkov et al., 2019) for C are the possible values that are explored for the two hyperparameters, "kernel" and "C." The training data are fitted to the SVM model that performs the best. Confusion matrix and classification report metrics are used in performance evaluation.

5 Baidu's deep speech

The state-of-the-art speech recognition system known as Deep Speech was developed using Baidu's end-to-end ASR architecture. A massive amount of speech data is trained using multiple GPUs and an RNN. Baidu's Deep Speech can learn directly from a large set of data, so it does not require speech adaptation or noise filtering. Deep RNN training will be based on supervised learning. From voice samples, mel-frequency cepstral coefficients are extracted, and transcription is output directly. A full voice recognition system powered by deep learning and its structure. The system generates a matrix of character probabilities, which shows that it gives each character in the alphabet a chance at each period step, indicating the likelihood that that particular character will match the audio. Furthermore, the Connectionist Temporal Classification (CTC) loss function increases the probability of accurate transcription. TensorFlow uses Baidu's Deep Speech Architecture to implement Mozilla Deep Speech, enabling the creation of prototypes for any dialect. It is simpler to operate and performs better in noisy environments than other traditional systems. This system's main advantage is that it outperforms traditional speech recognition systems, capable of handling speaker oscillation, echo, and background noise. From audio files, a time series spectrogram is produced, with each time slice representing a vector of audio characteristics. Three of the five unseen layers that comprise the RNN that powers the Deep Speech model are non-recurrent. Figure 10 shows the architecture of Baidu's Deep Speech system.

5.1 Acoustic model and language model

The acoustic archetypal generates a likelihood distribution over the characters of the alphabet in response to audio. The acoustic model takes up the majority of the training time. Typically, three steps are involved in the feature extraction process. The acoustic front end, also known as speech analysis, is the initial phase. It creates raw features by performing a type of temporal analysis of the signal's spectrum. The acoustic model's task is to use the sequence-to-sequence Deep Speech algorithm to identify which acoustic signals correspond to which specific letters. The language model helps translate these probabilities into comprehensible language words, followed by extensive labeled voice training on a large volume of data. The most important things to consider are the data that are rarely or never present in our training sets. We combine our system with one of these n-gram language models since they are readily trained from large unlabeled text datasets. Language models are typically trained by minimizing confusion on training data and by observing word sequences in text corpora that contain millions of word tokens. A variety of toolkits, including SRILM, KENLM, and open-game toolkits, are used to generate language models. It is necessary to train the linguistic model and the audio model with the same alphabet. alphabet.txt is the glue that holds the linguistic model and the acoustic model together. The neural network utilized in the acoustic model was trained on a corpus of voice and transcripts, which was created with TensorFlow. An n-gram model trained with KENLM is the morphological ideal, and the training data are a corpus of text. As inputs are fed into the network for a reference window of size k, the ith unit in a convolutional layer l at a timestamp t delivers M(l,i), as shown in Equation 6, which represents the architecture of a deep RNN using Arabic data.

$$M^{(l,i)} = \sigma\left(\omega^{(l,i)} \cdot M^{l-1}_{t-k:t+k}\right) \tag{6}$$

Here, M(0) denotes the input, and it contains 13 units. $\sigma(.)$ is the activation function as in Equation 7, and the hidden fully connected layers use a Rectified Linear Unit (ReLU) activation function. We always constrain the output of a convolution unit to up to 5 (Wu et al., 2024).

$$\sigma(x) = \min(\max(0, x), 5) \tag{7}$$

At any timestamp t, the units at layer l of the recurrent bidirectional LSTM take updates from both past and future timestamps, as shown in Equations 8, 9.

$$\overrightarrow{M^l_t} = tanh\left(\omega^l \cdot \mathrm{M} + \overrightarrow{U^l} \cdot \overrightarrow{M^l_{t-1}} + b^l\right) \tag{8}$$

$$\overleftarrow{M^l_t} = tanh\left(\omega^l \cdot \mathrm{M} + \overleftarrow{U^l} \cdot \overleftarrow{M^l_{t+1}} + b^l\right) \tag{9}$$

where ω^l is the input hidden weight matrix and U^l is a recurrent weight matrix. The sum of forward and backward directional states yields an "informed state" (hl), which is shaped by the prior transitional probabilities of the phonemes. The activation function tanh(.) acts like a squashing function, as shown in Equation 10 (Morais, 2025).

$$\tan h(x) = \frac{e^x - e^x}{e^x + e^x} \tag{10}$$

The processed cepstral coefficients flow through the recurrent layers, and each upper layer receives this processed information from its immediate lower layer, which is given in Equation 11.

$$M^l_t = f\left(\omega^l \cdot M^{l-1}_t + b^l\right) \tag{11}$$

The output is a softmax layer that gives a probability distribution over phonemes, shown in Equation 12.

$$P\left(o^k_t = k/x\right) = \frac{e^{\omega^L_k \cdot h^{L-1}_t}}{\Sigma_i e^{\omega^L_k \cdot h^{L-1}_t}} \tag{12}$$

The value of the output unit at any timestamp t will indicate the probability of the corresponding phoneme n as predicted by

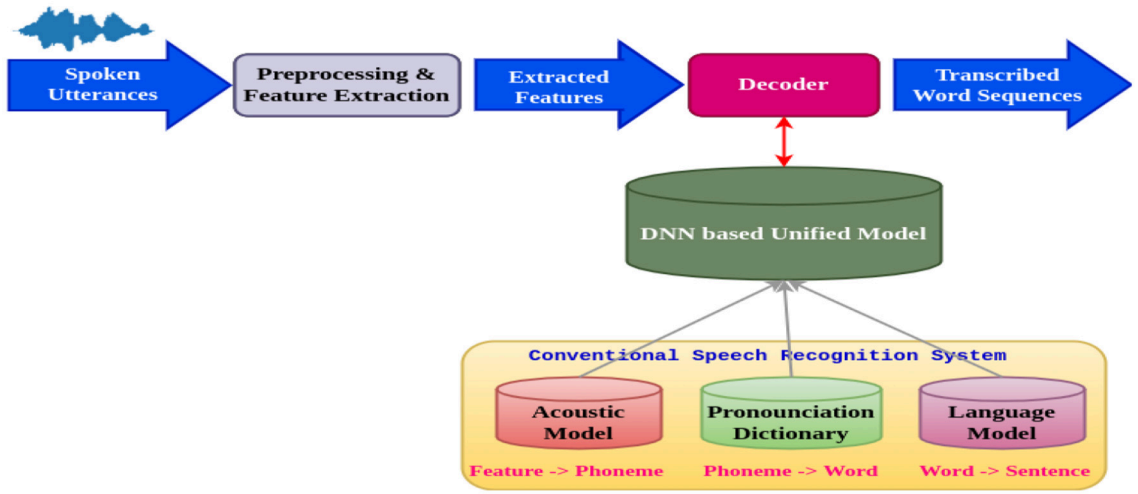

FIGURE 10
Baidu's Deep speech structure.

```
      100     1000     1001     1002     1003     1004     1005     1006  \
0  0.00005  0.00005  0.00005  0.00005  0.00005  0.00005  0.00005  0.00005

      1007     1008  ...        يى        يوان       يو       يه       ينى  \
0  0.00005  0.00005  ...  0.000198  0.040961  0.001785  0.000149  0.000099

        يين      يبع      يبقى      يبد      يبى
0  0.000099  0.00005  0.000099  0.00005  0.000694
```

FIGURE 11
Sample TF-IDF vectorizer data.

the network. The network is then trained using the CTC loss function, and the parameters of the network are updated using the backpropagation through time (BPTT) algorithm. Then 32-bit beam search decoder is used to construct the output from the phoneme distribution. The Term Frequency Inverse Document Frequency (TF-IDF) vectorizer is a useful tool for translating Arabic text data into numerical vectors. When analyzing text at the character level, it considers individual characters, pairs of characters, and triplets of characters. This is an important step for the Arabic script. It learns the vocabulary and term importance from the data and then creates TF-IDF vectors for each document. Based on the frequency of each term in the document and rarity across the dataset, these vectors indicate the relative importance of each term. Then, among other NLP tasks, these vectors can be used to train machine learning models for document classification, hidden topic identification, and document similarity comparison. The two main tasks completed by the vectorizer are stemming/lemmatizing Arabic text and normalizing it. The sample data are shown in Figure 11.

To calculate the probability of each sentence, the function counts the number of sentences (n-grams) that have been viewed so far, divides that count by the total number of sentences, and increases the count for each sentence. This is a basic method to determine the word or words that will appear next in a given sequence and to calculate the probability that a sentence will appear again based on how frequently it appears in the dataset. It separates Arabic text data into words, cleans it up, and calculates the probability that different word combinations (n-grams) will occur together. A sample prediction is shown in Figure 12.

5.2 Augmentation and hyperparameter setup

5.2.1 Baidu's deep speech hyperparameters

The majority of the hyperparameters in the preconfiguration for Mozilla Deep Speech remained unchanged. Nonetheless, the batch size was slightly modified in consideration of the machine's capabilities and the amount of training data. Furthermore, Deep Speech offers the ability to create checkpoints, allowing training to be resumed in the event of an error using the checkpoints. Either we create a checkpoint directory and store the training checkpoints there, or we freight the Deep Speech frontier directory containing the training checkpoints. Prediction accuracy is calculated using

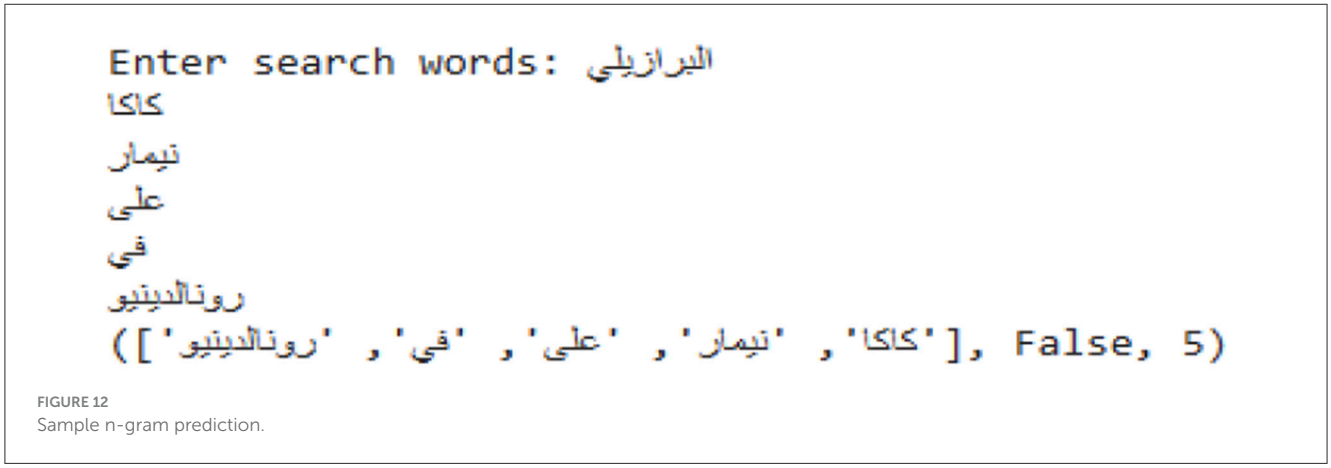

FIGURE 12
Sample n-gram prediction.

the loss. As the loss decreases, the difference between the neural network's predictions and the actual known values becomes smaller. When there is no reduction in loss, the parameter indicates how many training epochs should be considered as a plateau.

- **Hyperparameter optimization:** Optuna is a framework utilized for hyperparameter optimization. It specifically adjusts lm_alpha, which is a language model weight, and lm_beta is a word insertion bonus. To reduce the WER and CER on a designated test set, it systematically assesses several combinations of these parameters, dynamically reinitializing the TensorFlow graph for each iteration and relaying intermediate performance metrics to Optuna, which subsequently directs the search intelligently and eliminates unpromising trials to enhance efficiency. The model ascertains whether to optimize for WER or CER according to the loaded scorer's mode and offers a definitive entry point for users to commence this essential post-training optimization procedure, yielding the optimal parameters and their associated performance.
- **Reduce plateau**: If training does not result in a decrease in loss over time, it is said to have plateaued. It is possible to break through the plateau and keep reducing losses by adjusting the learning rate and other parameters.
- **Early stopping:** If training does not eventually reduce loss, an early termination is an option.
- **Dropout:** When training produces a model with poor generalization, it is referred to as overfitting and has an impact on the model's generalizability. A method called "dropout" enhances the generalizability of the model by arbitrarily eliminating nodes from the neural network to lessen overfitting.
- **Steps and Epochs:** A training set's entire cycle is referred to as an epoch. Batch size affects how much memory is required for processing. Fifteen epochs and a batch size of four are employed for this optimization.
- **Train–test split:** The training loop efficiently manages data loading, preprocessing, and augmentation, while enabling multi-GPU training by distributing computations across "towers" to average gradients for faster updates. Key components, including adaptive learning rate reduction during performance plateaus, early stopping to prevent overfitting, and thorough checkpointing, which entails retaining the best-performing model on a validation set, are integrated to ensure rapid and effective model development. This provides functionalities for autonomous evaluation of models on test datasets and the creation of efficient inference graphs, representing a complete solution for DeepSpeech model training and deployment. We have utilized 70% of the audio data for training 15% for testing, and 15% for validation.

TABLE 2 Hyperparameters of grid search.

Scores	Decision tree	XGBoost	KNN	Random forest
Mean fit time	0.0135	0.0317	0.0234	0.0293
Standard fit time	0.0007	0.0009	0.0020	0.0009
Mean score time	0.0037	0.0112	0.0030	0.0101
Standard score time	1.2655	4.6037	7.41052	1.0215
Mean test score	0.9973	0.9886	0.9980	0.9900
Standard test score	0.0020	0.0028	0.0019	0.0027
Rank test score	2.000	3.000	1.000	3.000

5.2.2 Machine learning hyperparameters

Table 2 shows that the grid search method uses different values of hyperparameters in each run. The first run uses the C values of 73, 79, 50, and 52, while the second run uses the C values of 19, 81, 72, and 89. The fit and score time are mentioned in Table 2.

5.2.3 Computational environment

All experimental methods were performed on a MacBook Pro, specifically configured with a 1.4 GHz Quad-Core Intel Core i5 processor. The system employed Intel Iris Plus Graphics 645 for graphics processing, featuring 1,536 MB of memory. The device was equipped with 8 GB of 2,133 MHz LPDDR3 RAM and ran

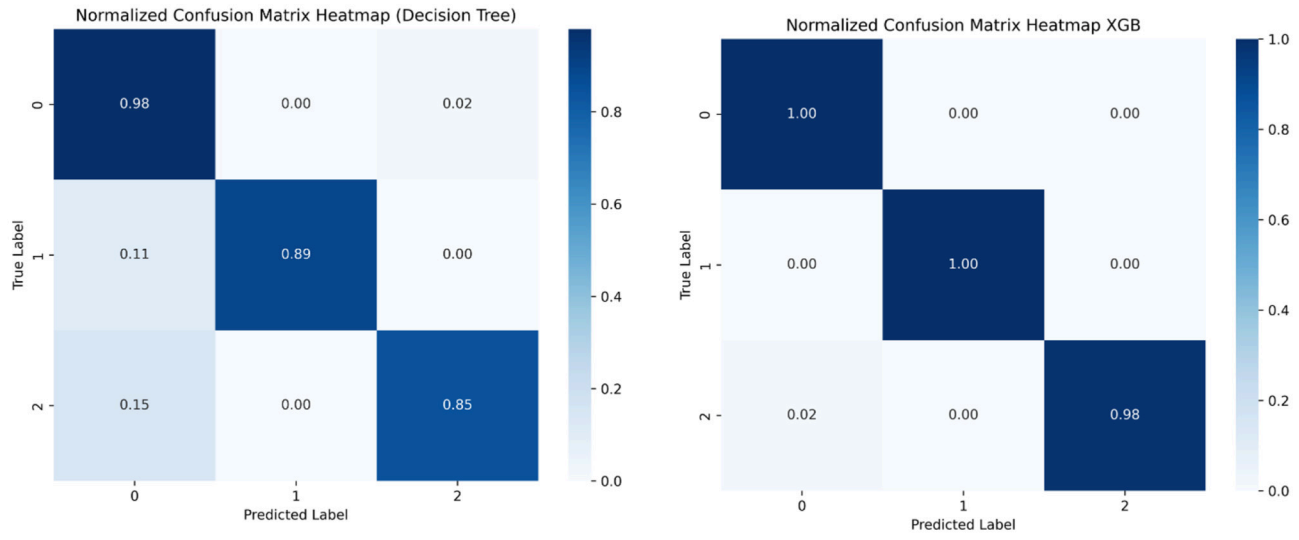

FIGURE 13
Confusion matrix, decision tree, and XGBoost.

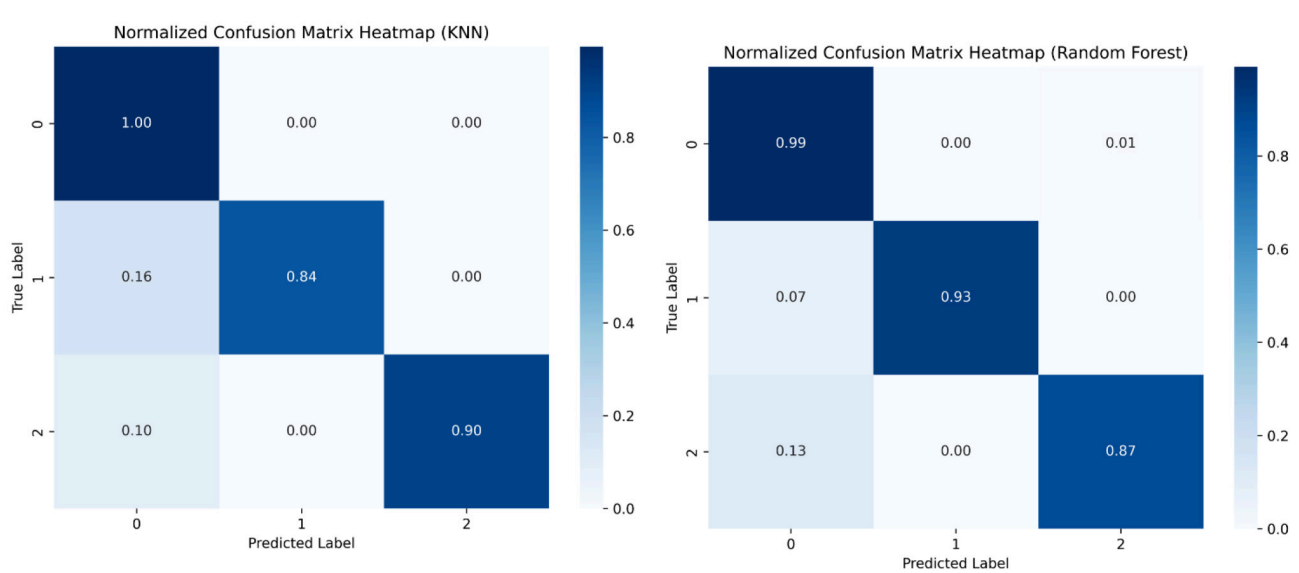

FIGURE 14
Confusion matrix KNN and Random Forest.

macOS Sequoia version 15.5. The dataset and computational outputs were stored on a 250.69 GB Macintosh HD, with 112.16 GB of space available during the experimental phase. This configuration facilitated the computational framework for all data processing, model training, and evaluation activities conducted in this research.

6 Results and discussions

6.1 Confusion matrix

Confusion matrices are specially used to visualize a model's performance in classification problems. They display the frequency of errors, such as false positives and false negatives, as well as the proportion of correctly classified data points, such as true positives and true negatives. The model predicts 1,145 actual instances of class 1 correctly and 55 actual instances of class 2, and 86 out of 87 actual instances of class 3. Figures 13, 14 show the confusion matrices.

6.2 Classification report

Both the confusion matrix and classification report indicate that the model achieved excellent performance with perfect accuracy, precision, recall, and F1-score for each class. Table 3 shows the classification report.

TABLE 3 Classification report.

Classifiers	Class	Precision	Recall	F1-score	Support
Decision tree	0	1.00	0.99	0.99	99
	1	1.00	1.00	1.00	1134
	2	1.00	1.00	1.00	54
XGBoost	0	0.99	0.98	0.98	99
	1	1.0	1.0	1.0	1126
	2	1.00	0.98	0.99	62
KNN	0	0.95	0.87	0.91	95
	1	0.98	1.00	0.99	1137
	2	1.00	0.76	0.87	55
Random Forest	0	0.88	0.90	0.89	78
	1	0.99	0.99	0.99	1153
	2	0.98	0.95	0.96	56

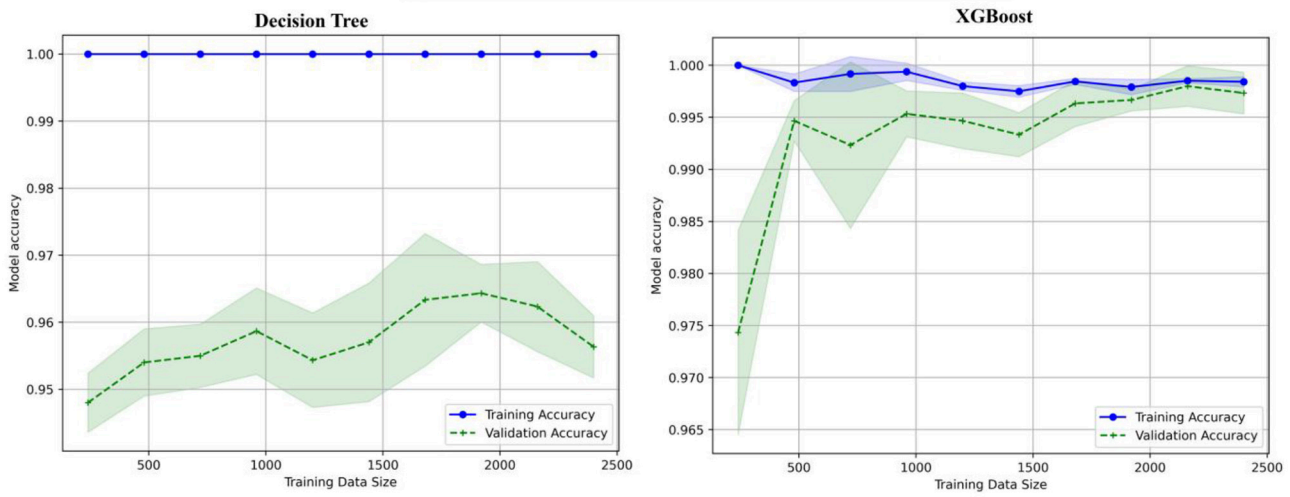

FIGURE 15
Learning curve for decision tree and XGBoost.

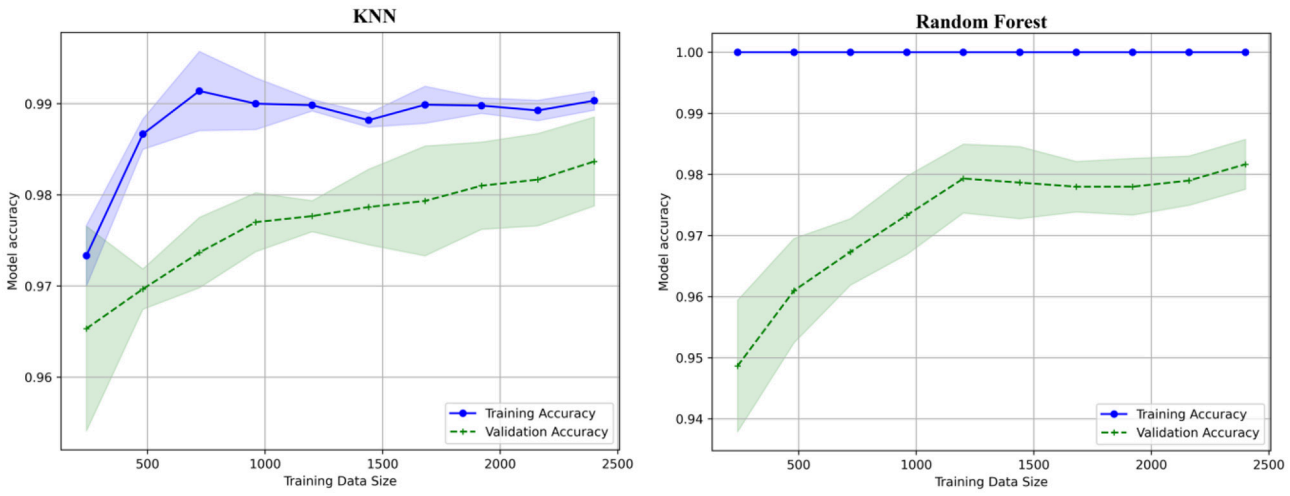

FIGURE 16
Learning curves for KNN and random forest.

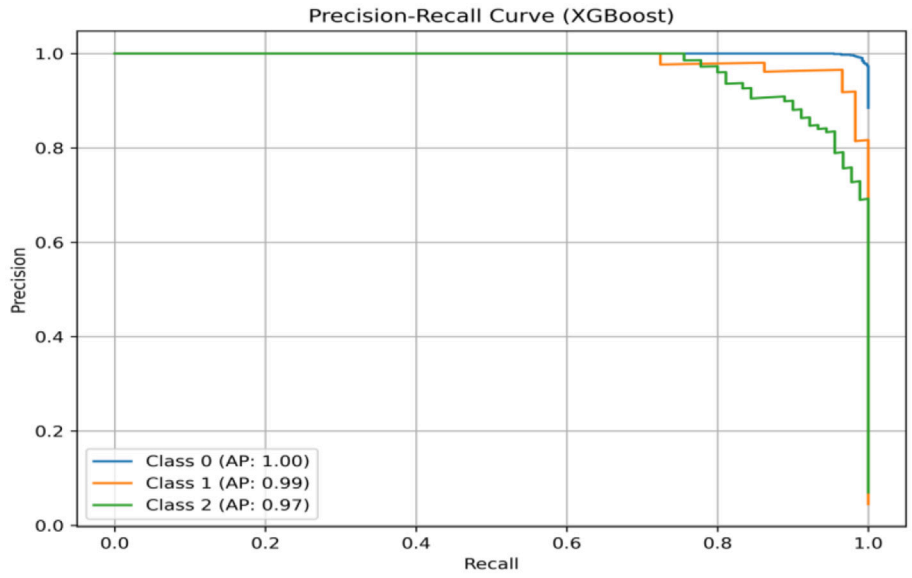

FIGURE 17
Precision–recall curve for XGBoost.

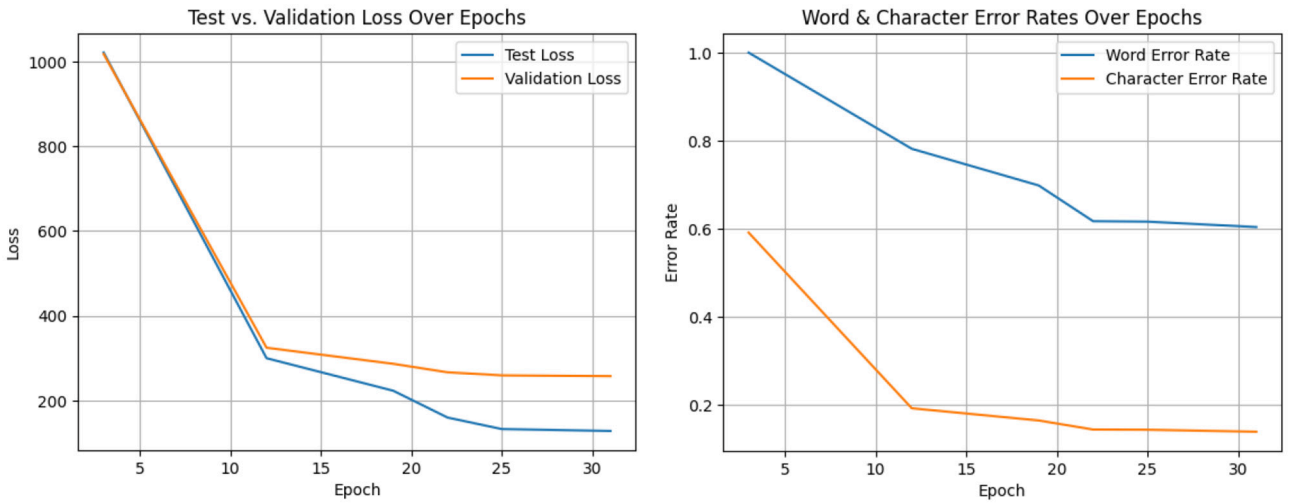

FIGURE 18
Test and validation loss vs. epochs and word and character error rate vs. epochs.

6.3 Learning curve

The learning curve shows the x-axis with values between 500 and 2,500 labeled as training data size, shown in Figure 15. The model accuracy y-axis has a range of 0.95 to 1.0. Two lines are displayed, one green for validation accuracy and one blue for training accuracy. As the size of the training data increases, the validation accuracy also increases, indicating that data are being trained well and validated. The learning curves for the decision tree, XGboost, KNN, and Random Forest are shown in Figures 15, 16.

TABLE 4 Model performance analysis.

Epoch	Test loss	Validation loss	Word error rate	Character error rate
3	1,017.0	1021.4	1.0000	0.59118
12	300.00	324.70	0.7815	0.1920
19	223.27	286.77	0.6982	0.1643
22	160.01	266.72	0.6170	0.1437
25	132.86	259.57	0.6160	0.1432
31	128.33	257.66	0.6037	0.1387

TABLE 5 Model performance analysis—best model.

Epoch	Best WER	Best CER	Loss at best WER/CER	Arabic text	English text
12	0.4687	0.1060	110.289	السَيف قِصَر فِي الأُسبُوعِيّ اجتِمَاعُهُ الوُزَرَاء مَجلِسُ عَقَدَ المحَمَّد نَاصِر الشَّيخ الشَّيخ مَجلِسِ رَئِيسِ سُمُوّ بِرِئَاسَةِ كَشَفَ الهَامَّة المِلَفَّاتِ مِن مَجمُوعَةِ الوُزَرَاءُ تَدَاوَلَ ت حَيثُ رَوضَانِ الالوُزَرَاء مَجلِس لِشُؤُونِ الدَّولَة وَزِيرِ عَنهَا الرَّوضَان	The Cabinet held its weekly meeting at Seif Palace under the chairmanship of His Highness the Prime Minister Sheikh Nasser Al-Mohammed, where the ministers deliberated a set of important files revealed by Minister of State for Cabinet Affairs Roudhan Al-Roudhan
19	0.3720	0.0568	276.147	بَعدَ أَشهُر مُنذُ إضرَابَاتٍ يَشهَدُ اليَمَن أَنَّ إِلَى الإِشَارَةُ تَجدُرُ عَلِيّوَتِلكَ الحَاكِم لِلنِّظَامِ المُؤَيِّدَةِ وَالمَسِيرَاتِ المُظَاهَرَاتِ الرَّئِيس يَتَلَقَّى فِيمَا بِإِسقَاطِه تُطَالِبُ وَالَّتِي لَه المُعَارَضَةِ عَقِبَ السَّعُودِيَّة السَّعُودِيَّة المَملَكَةِ فِي العِلَاجَ صَالِح عَبدالله الشَّهر هَذَا مِن سَابِقِ وَقتٍ فِي الرِّئَاسِيّ القَصر عَلَى هُجُوم	Yemen has been witnessing strikes for months after demonstrations and marches in support of the ruling regime and those opposing it, demanding its ouster, while President Ali Abdullah Saleh is receiving treatment in Saudi Arabia following an attack on the presidential palace earlier this month.

6.4 Precision–recall curve

The graphical tool called a precision–recall curve (PRC) is used to assess how well the classification model performs in multiclass problems, as shown in Figure 17. PRCs offer insight into the tradeoff between precision and recall in contrast with the receiver operating characteristic area under the curve (ROC AUC), which concentrates on binary classification. The ROC AUC score is obtained as 0.99928. The WER is the percentage of words that the system incorrectly recognizes, and the CER is the percentage of characters that the system recognizes incorrectly. This shows that the speaker's ability to speak correctly has improved, as has the speech recognition system's ability to recognize their speech. The graph also shows that the WER continuously outperforms the CER. This is because the speech recognition system finds it easier to identify individual characters.

Figure 18 shows the test and validation loss vs. various epochs and the word and character error rate vs. epochs of the system's WER and CER plotted against time. The WER is the percentage of words that the system incorrectly predicts, and the CER is the percentage of characters that the system incorrectly predicts (Baghdasaryan, 2022). The graph shows that both the WER and CER show a decrease over time, suggesting that the system's speech recognition performance is improving. In contrast, the WER constantly exceeds the CER. The reason for this is that individual characters are recognized by the algorithm more readily than entire words. The graph also shows how the WER and CER start to plateau after a certain number of epochs. The graph shows that the voice recognition system is training effectively. The system's increasing efficiency is demonstrated by the decrease in WER and CER over time. The word error rate is the most popular metric for ASR.

$$\mathrm{WER} = \frac{S_w + D_w + I_w}{N_w} \tag{13}$$

When a word in the reference sequence is transcribed as a different word, it is called a substitute word (Sw). When a word is completely absent from the automatic transcription, it is referred to as a deleted word (Dw). The number of words inserted is Iw. This means the word's appearance in the transcription has no correspondent in the reference word sequence. As it lacks the upper bound, the word error rate only indicates whether one system is superior to another. For this reason, a character error rate is used.

$$\mathrm{CER} = \frac{s + d + i}{N} \tag{14}$$

Table 4 describes the entire model analysis. The size and complexity of the exercise data, along with the system's design, will determine the ideal number of epochs for training a speech recognition system.

Table 5 illustrates the best model analysis and the corresponding transcribed Arabic text.

6.5 Discussion

Upon examining the performance of diverse ASR models, some significant themes and insights arise concerning their efficacy across various languages and architectural methodologies. The data reveals a wide range of WERs, from an exceptional 0.720% for the suggested Arabic DeepSpeech model to a maximum of 58.87% for Kazakh utilizing Kaldi. Recent improvements in deep learning models, especially Transformer-based architectures such as XLSR-Wav2Vec 2.0 for Turkish, exhibit markedly lower word error rates (0.23%) compared to previous or toolkit-based methodologies. DeepSpeech is a widely utilized model for several languages (Bengali, Russian, German, Tunisian, Arabic), although its efficacy fluctuates, indicating a significant impact of linguistic attributes and dataset quality. The incorporation of various languages, including Arabic, Bengali, German, Hindi, Kazakh, Russian, Tunisian, and Turkish, emphasizes the international endeavor in ASR development while revealing persistent challenges in attaining universal high performance, particularly for languages characterized by intricate phonetics or scarce resources. The efficacy of the built Baidu's Deep Speech model was meticulously assessed using an independent test dataset in our proposed work. This dataset, completely omitted from the model's training and validation phases, functioned as a vital assessment of the model's capacity to generalize to novel, previously unencountered data. Our results indicate that the model attained a WER of 0.3720 and a CER of 0.0568 during training and 0.19 WER and 0.02 CER during the testing phase.

TABLE 6 Comparison table with previous works.

Reference	Year	Model	Language	WER
Kazakh speech and recognition methods (Karabaliyev and Kolesnikova, 2024)	2024	Kaldi Mozilla DeepSpeech Google Speech-to-Text API	Kazakh speech	56.87% 55.36% 52.97%
End-to-end Bengali speech recognition (Nahid et al., 2019)	2019	Bidirectional LSTM	Bengali speech	8.20%
Russian-language speech recognition (Iakushkin et al., 2018)	2018	DeepSpeech	Russian speech	18%
German speech recognition (Xu et al., 2020)	2020	DeepSpeech	German speech	12.3%
German end-to-end speech recognition (Agarwal and Zesch, 2019)	2019	DeepSpeech	German speech	15.1%
Tunisian dialectal end-to-end speech recognition (Messaoudi et al., 2021)	2021	DeepSpeech	Tunisian speech	24.4%
Hindi speech recognition (Kumar et al., 2012)	2012	HTK	Hindi speech	12.99%
Transformer-based Turkish automatic speech recognition (Tasar et al., 2024)	2024	XLSR-Wav2Vec 2.0	Turkish Speech	2.3%
Arabic phonic transcription (Elmahdy et al., 2011)	2011	ACA	Arabic	19%
Arabic autoencoder speech recognition (Mohammed Ameen and Abdulrahman Kadhim, 2023)	2023	Deep learning models	Arabic	4%
Convolutional neural networks to facilitate the continuous recognition of Arabic speech (Sayed et al., 2024)	2024	CNN-LSTM	Arabic	3.63%
Arabic speaker-independent continuous automatic speech recognition (Abushariah et al., 2012)	2012	Hidden Markov models	Arabic	11.27%
Proposed study		Baidu's Deep Speech	Arabic Speech	3.7%

The unsupervised clustering of MFCC features, together with traditional machine learning classification, could be applied to enhance speaker diarization, acoustic scene categorization, or, importantly, Arabic dialect identification from various audio sources. This feature is essential for augmenting customer service analytics, expanding accessibility tools, facilitating more efficient content filtering, and enriching language learning systems. Furthermore, the framework's proven effectiveness with unlabeled data provides a means for creating ASR solutions for additional low-resource languages or specialized fields that lack comprehensive annotated corpora, thus expanding its influence within the speech technology sector. Table 6 shows the comparison with previous studies.

7 Conclusion

In this study, we examined the effectiveness of using clustering and classification techniques in conjunction with MEL frequency extraction for Arabic audio data processing. This study also briefs on the effectiveness of Baidu's Deep Speech in Automatic speech recognition of the Arabic dataset. Our results demonstrate that MFCCs efficiently capture important features, facilitating the successful clustering of audio segments using K-means or hierarchical clustering algorithms. Additionally, we obtained a low loss of 128.33 for the training dataset and a validation loss of 257.66 by using Baidu's Deep Speech. The WER for the reference is 0.19, indicating that 19% of the words were misidentified. 2% of the characters in the reference were misidentified, according to the CER of 0.02 in the testing phase. The evaluation's findings are encouraging. The model has a respectable level of accuracy regarding Arabic speech recognition.

7.1 Future studies

Future studies might investigate applying the existing methods to other widely used Arabic dialects. Potential applications such as assistive technologies for the hearing-impaired, voice-enabled services in Arabic-speaking regions, and integration with NLP pipelines are possible. This would entail developing acoustic models tailored to a particular dialect or investigating transfer learning strategies to modify the current model to accommodate new dialectal data. Also, predicting the next word and character from Arabic text for audio-impaired individuals can be possible from the transcribed data.

Data availability statement

The datasets presented in this article are not readily available due to privacy reasons. Requests to access the datasets should be directed to fawaz.alanzi@ku.edu.kw.

Author contributions

FA-A: Conceptualization, Funding acquisition, Investigation, Project administration, Resources, Software, Supervision, Writing – review & editing. BS: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work is funded by the Kuwait University project number EO 02/22.

Acknowledgments

The authors wish to acknowledge the support provided by the Kuwait University (project number: EO 02/22), which contributed to different aspects of this research.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Abdul, Z. K., and Al-Talabani, A. K. (2022). Mel frequency cepstral coefficient and its applications: a review. *IEEE Access* 10, 122136–122158. doi: 10.1109/ACCESS.2022.3223444

Abushariah, M. A-. A. M., Ainon, R. N., Zainuddin, R., Elshafei, M., and Khalifa, O. O. (2012). Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus. *Int. Arab J. Inf. Technol.* 9, 84–93.

Afify, M., Sarikaya, R., Kuo, H-. K. J., Besacier, L., and Gao, Y. (2006). On the use of morphological analysis for dialectal Arabic speech recognition. *Interspeech*. 277–280. doi: 10.21437/Interspeech.2006-87

Agarwal, A., and Zesch, T. (2019). *German End-to-end Speech Recognition Based on DeepSpeech*. Konvens.

Agarwal, A., and Zesch, T. (2020). LTL-UDE at low-resource speech-to-text shared task: investigating mozilla deepspeech in a low-resource setting. *SwissText/KONVENS* 31, 40–47.

Ahmed, B. H. A., and Ghabayen, A. S. (2017). "Arabic automatic speech recognition enhancement," in *2017 Palestinian International Conference on Information and Communication Technology (PICICT)* (Gaza: IEEE), 98–102. doi: 10.1109/PICICT.2017.12

Al-Anzi, F. S., and Shalini, S. T. B. (2024). Revealing the next word and character in Arabic: an effective blend of long short-term memory networks and ARABERT, (in English). *Appl. Sci.* 14:10498. doi: 10.3390/app142210498

Algihab, W., Alawwad, N., Aldawish, A., and AlHumoud, S. (2019). "Arabic speech recognition with deep learning: a review," in *International Conference on Human-Computer Interaction* (Cham: Springer International Publishing), 15–31.

Alrumiah, S. S., and Al-Shargabi, A. A. (2023). A deep diacritics-based recognition model for Arabic speech: quranic verses as case study. *IEEE Access* 11, 81348–81360. doi: 10.1109/ACCESS.2023.3300972

Alsayadi, H. A., Abdelhamid, A. A., Hegazy, I., and Fayed, Z. T. (2021). Arabic speech recognition using end-to-end deep learning. *IET Signal Process.* 15, 521–534. doi: 10.1049/sil2.12057

Al-Zaro, S., Al-Ayyoub, M., and Osama, A-. K. (2025). Speaker-independent phoneme-based automatic quranic speech recognition using deep learning. *IEEE Access*. 99:1. doi: 10.1109/ACCESS.2025.3589252

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., et al. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin, in International conference on machine learning. *PMLR* 48, 173–182.

Baghdasaryan, V. H. (2022). Armenian speech recognition system: acoustic and language models. *Int. J. Sci. Adv*. 3, 719–724. doi: 10.51542/ijscia.v3i5.7

Belinkov, Y., Ali, A., and Glass, J. (2019). Analyzing phonetic and graphemic representations in end-to-end automatic speech recognition. *arXiv*. 81–85. doi: 10.21437/Interspeech.2019-2599

Dendani, B., Bahi, H., and Sari, T. (2020). "Ch. Chapter 24, speech enhancement based on deep autoencoder for remote arabic speech recognition," in *Image and Signal Processing, (Lecture Notes in Computer Science)*, (Cham: Springer), 221–229. doi: 10.1007/978-3-030-51935-3_24

Elmahdy, M., Gruhn, R., Abdennadher, S., and Minker, W. (2011). "Rapid phonetic transcription using everyday life natural chat alphabet orthography for dialectal Arabic speech recognition," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Prague: IEEE), 4936–4939. doi: 10.1109/ICASSP.2011.5947463

Gupta, S., Jaafar, J., Ahmad, W. W., and Bansal, A. (2013). Feature extraction using MFCC. *Signal Image Process. Int. J.* 4, 101–108. doi: 10.5121/sipij.2013.4408

Hori, T., Cho, J., and Watanabe, S. (2018). "End-to-end speech recognition with word-based rnn language models," in *2018 IEEE Spoken Language Technology Workshop (SLT)* (Athens), 389–396. doi: 10.1109/SLT.2018.8639693

Iakushkin, O. O., Fedoseev, G. A., Shaleva, A. S., Degtyarev, A. B., and Sedova, O. S. (2018). "Russian-language speech recognition system based on deepspeech," in *Proceedings of the VIII International Conference Distributed Computing and Grid-technologies in Science and Education (GRID 2018), Dubna, Moscow region, Russia, September 10–14, 2018*.

Karabaliyev, Y., and Kolesnikova, K. (2024). Kazakh speech and recognition methods: error analysis and improvement prospects, (in English). *Sci. J. Astana IT Univ*. 20, 62–75. doi: 10.37943/20DZGH8448

Karpagavalli, S., and Chandra, E. (2016). A review on automatic speech recognition architecture and approaches. *Int. J. Signal Process. Image Process. Pattern Recognit.* 9, 393–404. doi: 10.14257/ijsip.2016.9.4.34

Képesi, M., and Weruaga, L. (2006). Adaptive chirp-based time-frequency analysis of speech signals. *Speech Commun.* 48, 474–492. doi: 10.1016/j.specom.2005.08.004

Keshishian, M., Norman-Haignere, S., and Mesgarani, N. (2021). Understanding adaptive, multiscale temporal integration in deep speech recognition systems. *Adv. Neural Inf. Process. Syst*. 34, 24455–24467.

Kim, S., Hori, T., and Watanabe, S. (2017). "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (New Orleans, LA), 4835–4839. doi: 10.1109/ICASSP.2017.7953075

Kumar, K., Aggarwal, R., and Jain, A. (2012). A Hindi speech recognition system for connected words using HTK. *Int. J. Comp. Syst. Eng.* 1, 25–32. doi: 10.1504/IJCSYSE.2012.044740

Liu, T., Zhang, M., Li, Z., Dou, H., Zhang, W., Yang, J., et al. (2025). Machine learning-assisted wearable sensing systems for speech recognition and interaction. *Nat. Commun*. 16:2363. doi: 10.1038/s41467-025-57629-5

Masterson, M. (2015). Baidu's deep speech recognition bests google, apple, and bing. *Speech Technol. Mag*. 20:12.

Messaoudi, A., Haddad, H., Fourati, C., Hmida, M. B., Elhaj Mabrouk, A. B., Graiet, M., et al. (2021). Tunisian dialectal end to end speech recognation based on deep speech. *Procedia Comput. Sci.* 189, 183–190. doi: 10.1016/j.procs.2021.05.082

Mohammed Ameen, Z. J., and Abdulrahman Kadhim, A. (2023). Deep learning methods for arabic autoencoder speech recognition system for electro-larynx device. *Adv. Hum. Comp. Interact*. 2023:7398538. doi: 10.1155/2023/7398538

Morais, R. (2025). *DeepSpeech model*. Available online at: https://www.geeksforgeeks.org/deep-learning/speech-recognition-with-deepspeech-using-mozilla-s-deepspeech/

Musikic, N., Chepeha, D. B., and Popovic, M. R. (2025). Surface electromyography-based speech detection amid false triggers for artificial voice systems in laryngectomy patients. *IEEE Trans. Med. Robot. Bionics*. 7, 404–415. doi: 10.1109/TMRB.2025.3527685

Nagamine, T., Seltzer, M. L., and Mesgarani, N. (2016). "On the role of nonlinear transformations in deep neural network acoustic models," in *Presented at the Interspeech 2016* (San Francisco, CA). doi: 10.21437/Interspeech.2016-1406

Nahid, M. M. H., Purkaystha, B., and Islam, M. S. (2019). End-to-end Bengali speech recognition using deepspeech. *J. Eng. Res. Innovation Educ* 1, 40–49.

Nedal Turab, K. K. (2014). A novel Arabic speech recognition method using neural networks and Gaussian filtering. *Int. J. Electric. Electron. Comp. Syst*. 19, 40–49.

Pitton, J. W., Wang, K., and Juang, B-. H. (1996). Time-frequency analysis and auditory modeling for automatic recognition of speech. *Proc. IEEE* 84, 1199–1215. doi: 10.1109/5.535241

Priyank Dubey, B. S. (2023). Deep speech based end-to-end automated speech recognition (ASR) for Indian-English accents. *arXiv*. doi: 10.48550/arXiv.2204.00977

Rabiner, L., and Juang, B.- H. (1993). *Fundamentals of Speech Recognition*. Upper Saddle River, NJ: Prentice-Hall, Inc.

Sayed, S. A., Ahmed Abdel Azeem Abul Seoud, R., and Abdel Naby, H. Y. (2024). Convolutional neural networks to facilitate the continuous recognition of arabic speech with independent speakers. *J. Electrical Comp. Eng.* 2024:4976944. doi: 10.1155/2024/4976944

Srivathshan, S., Sree Ramya, G., and Bindu Babu, P. K. (2025). Active noise cancellation system using hybrid SF-ANC and FxANFIS algorithms. *J. Innovative Image Process*. 7, 119–145. doi: 10.36548/jiip.2025.1.006

Tasar, D. E., Koruyan, K., and Cilgin, C. (2024). Transformer-based Turkish automatic speech recognition, (in English). *Acta Infologica* 8, 1–10. doi: 10.26650/acin.1338604

Ullah, I., Zahid, H., Algarni, F., and Asghar Khan, M. (2022). Deep learning-based approach for arabic visual speech recognition. *Comp. Materials Continua* 71, 85–108. doi: 10.32604/cmc.2022.019450

Wu, Q., Wu, J., Chen, Y., and Zhang, Z. (2024). Research on intelligent speech interaction system based on residual neural network and baidu speech platform. *J. Intelligence Knowl. Eng*. 2:71. doi: 10.62517/jike.202404213

Xu, J., Matta, K., Islam, S., and Nürnberger, A. (2020). "German speech recognition system using deepspeech," in *Presented at the Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval* (Association for Computing Machinery). 102–106. doi: 10.1145/3443279.3443313

Yang, K., and Zhou, X. (2018). Unsupervised classification of hydrophone signals with an improved Mel-frequency cepstral coefficient based on measured data analysis. *IEEE Access* 7, 124937–124947. doi: 10.1109/ACCESS.2018.2886802

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 11 September 2025
DOI 10.3389/frai.2025.1653728

OPEN ACCESS

EDITED BY
Shadi Abudalfa,
King Fahd University of Petroleum and Minerals, Saudi Arabia

REVIEWED BY
Vasudevan Nedumpozhimana,
Trinity College Dublin, Ireland
Dhaou Ghoul,
Université Paris-Sorbonne, France

*CORRESPONDENCE
Areej Jaber
✉ a.jabir@ptuk.edu.ps

RECEIVED 25 June 2025
ACCEPTED 21 August 2025
PUBLISHED 11 September 2025

CITATION
Jaber A, Bahati I and Martínez P (2025) Leveraging pre-trained embeddings in an ensemble machine learning approach for Arabic sentiment analysis.
Front. Artif. Intell. 8:1653728.
doi: 10.3389/frai.2025.1653728

Leveraging pre-trained embeddings in an ensemble machine learning approach for Arabic sentiment analysis

Areej Jaber[1]*, Israa Bahati[1] and Paloma Martínez[2]

[1]Computer Science Department, Palestine Technical University - Kadoorie, Tulkarm, Palestine,
[2]Computer Science and Engineering Department, Universidad Carlos III de Madrid, Leganes, Spain

Introduction: Arabic sentiment analysis presents unique challenges due to the linguistic complexity of the language, including its wide range of dialects, orthographic ambiguity, and limited language resources. Addressing these issues is essential to develop robust sentiment classification systems.

Methods: This study investigates the application of ensemble machine learning methods for Arabic sentiment analysis. Several homogeneous ensemble techniques are implemented and evaluated on two datasets: the balanced ArTwitter dataset and the highly imbalanced Syria_Tweets dataset. To mitigate class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is employed. The models incorporate pre-trained word embeddings and unigram features.

Results: Experimental results indicate that individual classifiers using pre-trained embeddings achieve strong performance; however, ensemble models consistently yield superior outcomes. On the ArTwitter dataset, the ensemble of Naive Bayes, Support Vector Machine, and Decision Tree classifiers achieved an accuracy of 90.22% and an F1-score of 92.0%. On the Syria_Tweets dataset, an ensemble combining Stochastic Gradient Descent, k-Nearest Neighbors, and Random Forest attained 83.82% accuracy and an 83.86% F1-score.

Discussion: The findings highlight the effectiveness of ensemble learning in enhancing the robustness and generalizability of Arabic sentiment analysis systems. Incorporating pre-trained embeddings further strengthens performance, demonstrating that ensemble-based approaches can overcome challenges posed by linguistic complexity and dataset imbalance in Arabic natural language processing tasks.

KEYWORDS

ensemble learning, sentiment analysis, machine learning, Arabic language, SMOTE

1 Introduction

With recent advancements in Natural Language Processing (NLP), several text analysis tasks have been successfully automated, including disinformative tweets detection (Jaber and Martínez, 2023), word sense disambiguation (Jaber and Martínez, 2022), and propaganda detection (Duridi et al., 2025). Sentiment analysis, a subtask of text classification, aims to classify a piece of text into binary classes (positive or negative) or multi-class categories (positive, negative, neutral). It has found widespread application across various domains, including politics (Grover et al., 2025), business (Tiwari and Arora, 2025), and social media (Alotaibi et al., 2025).

The performance of sentiment analysis systems largely depends on two core phases: feature engineering and the choice of classification algorithms. Feature engineering refers to transforming raw textual data into numerical representations that capture the semantic and syntactic properties of the text. Traditional approaches such as Term Frequency-

Inverse Document Frequency (TF-IDF) and n-gram models have been effective in handling short texts (Nafis and Awang, 2021). More recent approaches based on word embeddings, including Word2Vec (Church, 2017), GloVe (Pennington et al., 2014), FastText (Joulin et al., 2016), and Large Language modeling (Mansour et al., 2025) provide rich semantic context and reduce the sparsity problem inherent in high-dimensional representations.

Among the classification strategies, ensemble learning has shown great promise in improving NLP task performance. The key idea of ensemble methods is to combine the predictions of multiple base classifiers to offset the weaknesses of individual models while leveraging their strengths. Ensemble learning based on machine learning algorithms has demonstrated its effectiveness across various NLP applications (Rane et al., 2024).

Arabic is one of the six official languages of the United Nations and is the native language of over 300 million people across 22 countries. However, Arabic sentiment analysis poses numerous challenges due to the linguistic complexity of the language. These challenges include morphological richness, the presence of multiple dialects, and the frequent use of figurative language such as ambiguity, sarcasm, and irony (Rahma et al., 2023), which makes sentiment classification more difficult (Alwakid et al., 2017).

The contribution of this work is an model based on a majority voting homogeneous ensemble machine learning approach. Exploring different vector-based feature representations and machine learning algorithms, including TF-IDF with ngrams and pretrained word embeddings. To address the issue of class imbalance during training, the Synthetic Minority Oversampling Technique (SMOTE) is employed Syria_tweet dataset. Optimize the hyperparameters of the proposed model to achieve the highest possible classification performance. The results are compared with the most relevant previous work, which demonstrates its superior performance.

The remainder of this article is organized as follows: Section 2 reviews prior studies on dialectal Arabic sentiment classification. Section 3 presents the proposed research methodology. Section 4 discusses the experimental results and evaluations. Finally, Section 5 concludes the study and outlines directions for future research.

2 Related work

Sentiment analysis has become quite popular in many languages, including Arabic, since social media, product evaluations and opinions, and user-generated content are becoming more and more important. Several comprehensive surveys have traced the evolution of Arabic sentiment analysis and mapped out the key resources in the field. Ghallab et al. (2020) reviewed work published between 2015 and 2019, grouping existing approaches into three main categories: lexicon-based, machine learning-based, and hybrid methods that combine the two. Their review also provided an overview of more than twenty available datasets, ranging from domain-specific corpora to large Twitter-based collections such as ASTD and ArSenTD-Lev, which remain popular because of Twitter's rich mix of short, informal, and often dialectal content.

A more focused perspective was offered by Obiedat et al. (2021), who surveyed research on **Arabic aspect-based sentiment analysis (ABSA). Their study covered early rule-based and lexicon methods, as well as more recent deep learning architectures that integrate pre-trained embeddings and attention mechanisms. They also listed key ABSA resources, including the SemEval Arabic corpora and HARD, and discussed persistent challenges such as handling the diversity of Arabic dialects, the scarcity of large annotated datasets, and the difficulty of building models that generalize well across domains.

Sentiment analysis approaches can be categorized into three categories: lexicon-based approaches, machine learning approaches, and hybrid approaches (Matrane et al., 2023).

In a lexicon-based technique, sentiment analysis operates by giving a polarity score to each token in the text. The ratings are then averaged, with positive, negative, and neutral values tallied individually. The overall polarity of the text is ascertained by identifying the greatest value among the various scores. Elshakankery and Ahmed (2019) introduced HILATSA, a hybrid incremental learning method that combines a lexicon-based approach with machine learning. The system updates its sentiment lexicon incrementally with newly labeled data. On the ArTwitter and Syria_Tweets datasets, it achieved an accuracy of 85% (SVM) and 75.5% (RNN), respectively.

Abdulla et al. (2013) conducted an initial study on Arabic sentiment analysis, comparing lexicon-based and corpus-based methodologies. In the lexicon-based technique, an Arabic sentiment lexicon was manually created by expanding a set of seed words and assigning polarity ratings, thereafter categorizing text based on the aggregate sentiment of its words. Their study used a manually annotated dataset of 2,000 Arabic social media comments and reviews, which underwent preprocessing using light stemming approaches. The lexicon-based technique achieved an accuracy of around 59%, demonstrating the feasibility of rule-based sentiment classification in the absence of huge labeled datasets, while also highlighting its dependence on the comprehensiveness and quality of the lexicon.

Mataoui et al. (2016) focused on vernacular Algerian Arabic, creating three dialect-specific sentiment lexicons and a manually annotated dataset sourced from social media. Their lexicon-based algorithm sorted texts by adding up the polarity of related phrases, which was around 61% accuracy. This shows that rule-driven methods may work well in very dialectal settings, but they also depend on having a complete vocabulary. Assiri et al. (2018) enhanced lexicon-based sentiment analysis for the Saudi Arabic dialect by creating a comprehensive dialectal lexicon and using weighted polarity scoring that accounts for negation and supplication. Their method got around 68% of the answers right on a Saudi social media dataset, which is better than standard lexical baselines.

Machine learning approaches have also been applied to ASA. This approach is based on an annotated corpus, which is fed into ML algorithms in the training phase; then, after the model is trained, unannotated sentences are fed to the model to predict their polarity. Aladeemy et al. (2024) applied a range of traditional machine learning algorithms—namely SVM, Random Forest, Decision Tree, Logistic Regression, and XGBoost—using BoW and TF-IDF representations with unigram and bigram features. The best result was achieved by SVM, with an accuracy of 90.3% using unigram features.

Tubishat et al. (2019) proposed an Improved Whale Optimization Algorithm (IWO for feature selection in Arabic

sentiment analysis. Their method integrates Elite Opposition-Based Learning to improve population diversity and Differential Evolution operators to refine the optimization process. The proposed approach was tested on four datasets and yielded a best average accuracy of 89.68% on the ArTwitter dataset. However, the introduction of pre-trained word embeddings brought a notable shift. For example, Gamal et al. (2019) introduced a Twitter benchmark dataset for ASA and showed that distributed word representations capture semantic context far better than traditional bag-of-words features, even for short and noisy tweets.

A more recent trend has been targeted sentiment analysis (TSA), which focuses on detecting sentiment toward a specific entity within a text. In this area, Sahmoud et al. (2022) released AT-ODTSA, a large-scale dataset of Arabic tweets annotated for open-domain TSA. This dataset spans multiple topics and sentiment targets, making it a valuable resource for fine-grained sentiment studies. However, our work differs in scope: we focus on overall tweet-level sentiment classification, applying and evaluating models on both a balanced dataset (ArTwitter) and a highly imbalanced one (Syria_Tweets).

Lately, transformer-based models have also entered the scene. For example, Alsalem and Abudalfa (2024) fine-tuned AraBERT for Arabic sentiment tasks, achieving impressive results but requiring significant computational resources. Likewise, a recent study Alosaimi et al. (2024) explored hybrid pipelines that combine pre-trained embeddings with traditional classifiers for low-resource languages. While promising, these works did not deeply investigate imbalanced Arabic datasets or compare classical ensemble methods under such conditions.

In contrast, our study combines multiple pre-trained embeddings with a homogeneous hard-voting ensemble of classical classifiers, and evaluates performance on both balanced and imbalanced datasets. We also address imbalance directly using SMOTE and report results using both accuracy and F1-score, allowing for a fairer and more informative comparison with recent state-of-the-art methods.

Ensemble Machine learning was applied by Saleh et al. (2022), which developed a heterogeneous stacking ensemble model that combines RNN, LSTM, and GRU as base learners with meta-learners such as Logistic Regression, Random Forest, and SVM. Using CBOW features, their model attained an accuracy of 83.12% on the ArTwitter dataset. Al-Azani and El-Alfy (2017) employed word2vec embeddings combined with single and ensemble machine learning classifiers to handle highly imbalanced sentiment datasets. They applied SMOTE for data balancing and reported their best result—80% accuracy—using the KNN classifier on the Syria_Tweets dataset.

While previous research has explored a range of lexicon-based, machine learning, deep learning, and ensemble techniques for Arabic sentiment analysis, most studies have either focused on a single dataset, relied heavily on deep neural models with high computational demands, or overlooked the performance implications of dataset imbalance. Our work distinguishes itself by systematically evaluating a homogeneous hard-voting ensemble of classical classifiers in combination with multiple pre-trained Arabic word embeddings. This design leverages the semantic richness of modern embeddings while retaining the efficiency and interpretability of traditional algorithms. Furthermore, by conducting experiments on both a balanced dataset (ArTwitter) and a highly imbalanced dataset (Syria_Tweets), and applying SMOTE to mitigate imbalance, we provide a more comprehensive assessment of model robustness.

3 Materials and methods

An overview of the proposed Arabic Sentiment Analysis Framework is illustrated in Figure 1. The process begins with dataset preprocessing, which includes several text-cleaning steps. The textual data is then transformed into numerical vectors using two feature engineering techniques: the first involves TF-IDF with n-gram representations, and the second leverages the averaged vectors of pre-trained Word2Vec embeddings. A set of individual machine learning classifiers is subsequently trained, with their hyperparameters optimized using Bayesian optimization. Finally, several hard voting ensemble models are constructed by combining different classifiers to improve overall performance. The following subsections provide a detailed explanation of each step in the proposed pipeline.

3.1 Dataset

This study employed two sets of data. The ArTwitter dataset, created by Abdulla et al. (2013), is a balanced corpus focusing on Modern Standard Arabic (MSA). Two thousand tweets of various topics, such as politics and arts, were gathered from Twitter and completely labeled by specialists in the field as either positive or negative. ArTwitter has been commonly used as a standard dataset in Arabic sentiment analysis research since it is balanced and includes high-quality annotations. The second data set is a highly unbalanced data set, which the Twitter API acquired from Syrian tweets in May 2014. Syria_Tweets (Mohammad et al., 2016) composed from 1,798 tweets; 1,350 are annotated as negative tweets and 448 are annotated as positive tweets. Table 1 illustrates the key characteristics of the used data sets.

3.2 Data set preprocessing

An essential phase is the preprocessing of the dataset, which guarantees that the data is clean, standardized, and fit for sentiment analysis. Due to the complexities of the Arabic language, this process employs various tailored methods to improve the dataset's quality and ensure that the text is well-prepared for both machine learning and ensemble learning models. The preprocessing pipeline initially involves the removal of NaN values and duplicates to uphold data integrity. Following this, the text undergoes systematic cleaning to tackle important linguistic challenges such as punctuation and inconsistencies in spelling and writing styles. Standardization techniques, such as removing punctuation and normalizing text, aid in unifying the data, thereby enhancing model accuracy. Further cleaning procedures are

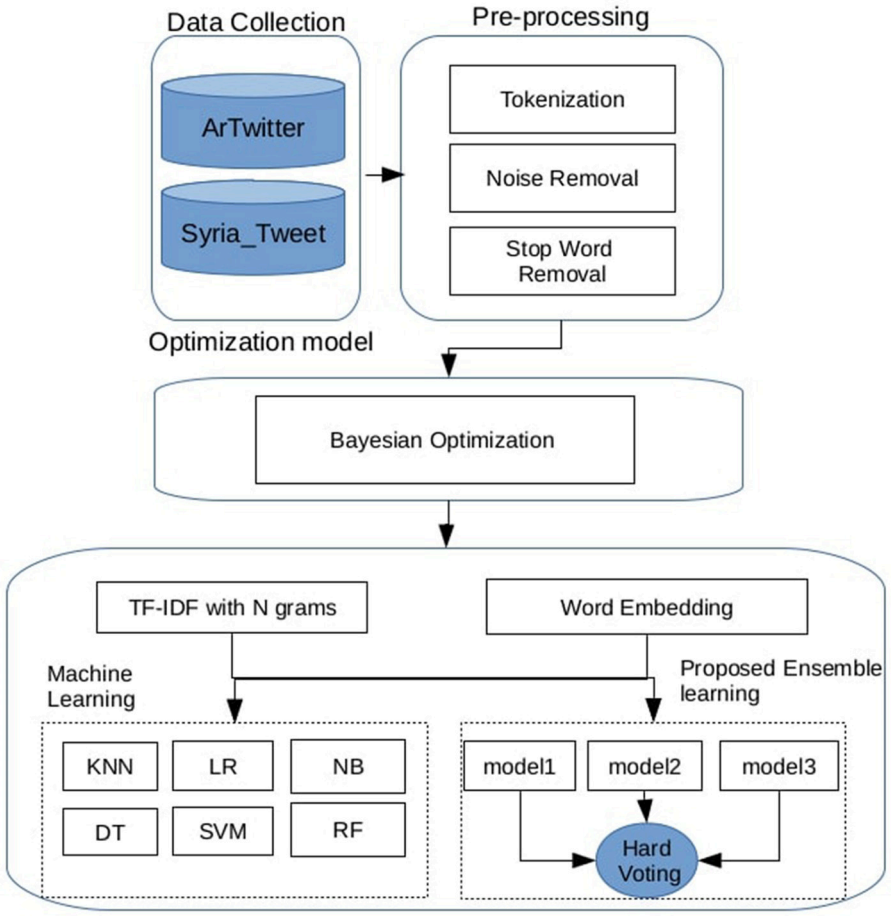

FIGURE 1
Architecture of the proposed arabic sentiment analysis framework.

TABLE 1 Key characteristics of the ArTwitter and Syria_Tweets sentiment analysis datasets.

Feature	ArTwitter	Syria_Tweets
Source	Twitter	Twitter
Language variety	Modern Standard Arabic (MSA)	Levantine dialectal Arabic
Annotation	Manually annotated	Manually annotated
Total tweets	1,951	1,798
Sentiment classes	Positive, negative	Positive, negative
Positive samples	993	1,350
Negative samples	958	448

implemented to remove noise and irrelevant elements, such as non-Arabic characters, emojis, and English words or numbers. These actions ensure that only pertinent information is retained, thus optimizing the dataset for sentiment classification. The preliminary data cleaning operations, which were performed by using the NLTK library (Bird et al., 2009) and the ISRI Arabic stemmer (Taghva et al., 2005), include:

- **Stopword removal:** removing common words like conjunctions (e.g: ثم) and (e.g: الى , من في,), which have little semantic importance and do not meaningfully assist in classification efforts.
- **Punctuation removal:** stripping punctuation from Arabic text to reduce extraneous data and simplify further analysis (e.g:?, !, ...).
- **Hashtag and mention removal:** eliminating hashtags and user tags (like @username, #hashtag) from the text.
- **Emoji removal:** extracting emoji characters using a regular expression pattern to cleanse the dataset by matching and eliminating emojis.
- **English words and numbers removal:** taking out English terms and numerals from the Arabic script using regular expressions to identify and discard typical alphanumeric sequences.
- **Character repetition handling:** reducing sequences of repeated characters to a single character (e.g: ,اللههههههههه).
- **Whitespace cleanup:** compressing multiple spaces into a single space for text uniformity.
- **Tokenization:** this step breaks down the polished text into discrete tokens or units by employing separator characters such

as spaces, commas, or tabs, facilitating separate analysis of each word or element.

3.3 Data balancing technique

An imbalanced dataset is characterized by an unequal distribution of class labels, where the majority class comprises a large number of training samples, and the minority class contains relatively few annotated instances. To address this issue, the *Synthetic Minority Oversampling Technique (SMOTE)* (Chawla et al., 2002) is one of the most widely adopted solutions.

SMOTE improves the representation of the minority class by generating synthetic samples based on the feature space similarities between existing instances. For each minority class instance $x_i \in S_{\min}$, SMOTE identifies its k-nearest neighbors (typically using Euclidean distance), and constructs synthetic examples by linearly interpolating between x_i and one of its neighbors. Specifically, a new sample is generated as:

$$x_{\text{new}} = x_i + \delta \cdot (x_{nn} - x_i) \tag{1}$$

where x_{nn} is one of the k-nearest neighbors of x_i, and $\delta \in [0, 1]$ is a random number. This interpolation ensures that the synthetic instances are consistent with the local topology of the minority class (He and Garcia, 2009). The oversampling process continues until the minority class is balanced or reaches a predefined target size. In our study, we applied SMOTE with $k = 5$ nearest neighbors. SMOTE technique was applied only to the training set, while the testing sets remained unbalanced, to maintain the original class distribution.

3.4 Feature representation methods

Transforming text into numerical values while representing the semantic meaning of the text is the nex step after the cleaning of the data. In this work, several forms of N-grams with TF-IDF representations were implemented, in addition to pre-trained word embedding with word2vec was leveraged to improve the performance of the proposed models. In the following subsections a brief descriptions for the data representation methods that were used in the study.

3.4.1 TF-IDF with n-grams

Term Frequency-Inverse Document Frequency (TF-IDF) is a common way to weight words and phrases in text classification. It looks at how important a word or phrase is in a document compared to a group of documents. It balances out two things: word Frequency (TF), which counts how many times a word appears in a text, and Inverse Document Frequency (IDF), which makes common words less important and puts greater emphasis on unique phrases. The TF-IDF score is calculated as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{\text{DF}(t)}\right) \tag{2}$$

TABLE 2 N-gram generation examples for feature extraction.

N-gram	Results
Original Arabic Sentence	[رائع جد مميز انت عمر]
Unigram	[رائع ,][جد],[مميز],[انت],عمر]
Bigram	[رائع [جد],جد [مميز],مميز [انت],انت عمر]
Trigram	[رائع جد [مميز ,]جد مميز [انت ,]مميز انت عمر]

where t is the term, d is the document, N is the total number of documents, and DF(t) is the number of documents containing term t. To capture local context and word co-occurrence patterns, we applied TF-IDF weighting over n-gram features.

N-grams (Jurafsky and Martin, 2009) represent one of the simplest and most widely used approaches to language modeling in natural language processing. They are used to represent textual data by capturing contiguous sequences of words. A single word forms a unigram, a sequence of two consecutive words is referred to as a bigram, and a sequence of three successive words is known as a trigram. Despite their simplicity, n-gram models effectively capture local context and are commonly used in various tasks such as text classification, sentiment analysis, and machine translation. Table 2 shows an example of how the sentence is tokenized based on the chosen type of n-grams.

In our study, we examined the effectiveness of three types of n-gram features—unigram, bigram, and trigram—in combination with machine learning and ensemble learning approaches.

3.4.2 Pre-trained word embeddings

ArWordVec (Fouad et al., 2020) is a huge set of pretrained models that is built from 55 million tweets with different topics, including social affairs, politics, and health care. The embeddings are trained by word2vec and Glove methods with different approaches, window size, and vector size.

In our experiments, we used the Word2Vec architecture with the Skip-Gram (SG) approach, a window size of 3, and an embedding dimension of 300. The Skip-Gram model was chosen because it tends to perform better with infrequent words and is more effective at capturing detailed semantic relationships than the Continuous Bag-of-Words (CBOW) method (Mikolov et al., 2013a). A relatively small window size of 3 was selected to emphasize local contextual dependencies, which suits the characteristics of the used dataset, while limiting the influence of less relevant, distant words. The choice of a 300-dimensional vector is consistent with common practice in earlier studies (Mikolov et al., 2013b; Pennington et al., 2014), as it offers a practical balance between the ability to represent nuanced meaning and the need to keep training time and memory use manageable.

To leverage the strengths of the model, we compute the average of the word embedding vectors across the entire sentence, as defined in Equation 3.

$$\text{AVG}(E(S)) = \frac{1}{n}\sum_{i=1}^{n} \text{Emb}(S(i)) \tag{3}$$

Where AVG(E(S)) is the average embedding of the sentence S, S(i) is the i-th word in the sentence, Emb(S(i)) is the embedding of word i, and n is the total number of words in the sentence.

3.5 Individual machine learning models

Several individual Machine learning classifiers were implemented. A brief definition of the selected algorithms is provided below:

- Naïve Bayes (NB) (Duda et al., 2001): is a probabilistic classifier that uses Bayes' theorem and assumes that features are very independent of each other. Even though it's simple, it does an amazing job at classifying text because it's fast and works well with data that has a lot of dimensions.
- Support Vector Machine (SVM) (Cortes, 1995): builds the best hyperplane to divide classes with the most space between them. This makes it work well in spaces with a lot of dimensions. It is considered powerful due to its kernel functions that work well for non-linear decision boundaries.
- Stochastic Gradient Descent (SGD) (Bottou, 2010): it is a good choice for sparse datasets, it updates its model parameters in an iterative optimization process for linear classifiers.
- Logistic Regression (LR) (Cox, 1958): logistic functions are used to model of the probability of binary results.
- Random Forest (RF) (Breiman, 2001): builds multiple decision trees and combines their results to enhance generalization and decrease overfitting.

3.6 Ensemble learning models

Ensemble learning aims to optimize the classification task by fusing multiple base classifiers, which reduces the variance of the predictions of the individual classifiers (Kumar et al., 2020). Thus, several ensemble techniques are designed to achieve this goal, such as bagging (Yang et al., 2020), boosting (Deng et al., 2023), and voting (Onan et al., 2016).

The use of heterogeneous base classifiers is utilized in the Voting technique for the production of concurrent ensemble networks. Voting is categorized into two types: weighted averaging and majority voting, which this study uses.

In majority voting, each model "votes" for a class label; the most voted label is chosen for the final predictions. This happens by combining several individual classifiers, which are known as base learners, and the majority vote makes the final decision. In this study, combinations of sets of individual machine learning classifiers were tested, it is named v with numbers from 1 to 11.

3.7 Evaluation metrics

To measure the performance of the proposed approaches, two datasets were used with different setups. We performed an 80/20 train-test split using stratified sampling, ensuring that both subsets maintained the original class imbalance of approximately 75% negative and 25% positive tweets. SMOTE was applied only to the training set, while the test set remained untouched to evaluate model performance on real-world imbalanced data. The vectorized training and test datasets were input into the Machine learning classifiers in addition to ensemble learning.

The machine learning classifiers were trained to determine the sentiment polarity of the reviews as either positive or negative. To evaluate model performance, we used four standard classification metrics: precision, recall, F-measure, and accuracy. These are defined in Equations 4–7.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$F\text{-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

where *TP*, *TN*, *FP*, and *FN* represent true positives, true negatives, false positives, and false negatives, respectively.

4 Experiments results and discussion

4.1 Experiments setup

All experiments were performed on the Google Colab platform, utilizing a Tesla T4 GPU for accelerated computation mainly for faster processing of the embedding and hyperparameter tuning. After data set preprocessing, the data was split into 80% training and 20% testing data sets. Then, the SMOTE technique was applied to the Syria_tweet dataset to solve the imbalanced dataset problem. SMOTE techniques were applied to the training dataset to make sure the learned model would be tested on real test data.

4.1.1 Hyperparameter optimization

For optimizing the performance of the proposed models, Bayesian Hyperparameter optimization techniques (Snoek et al., 2012) were applied to both TF-IDF with n-grams and word embeddings feature extractions.The optimization techniques were applied via the Gaussian Process-based. This method models the objective function using a Gaussian Process, which provides uncertainty estimates that guide the search efficiently through the hyperparameter space. We set the number of iterations to 32 and employed three-fold cross-validation. As shown in Table 3, the optimal hyperparameter values vary between the two datasets. For example, the `alpha` parameter in Naive Bayes is smaller for the Syria_Tweets dataset compared to ArTwitter. Additionally, the SVM model uses a linear kernel for ArTwitter, while an RBF kernel is preferred for Syria_Tweets.

Table 4 shows the optimal values of the hyperparameters for different sets of machine learning algorithms after applying Bayesian optimization.

It's important to note that the tuning parameters are very different between the two datasets. For example, SGD

TABLE 3 Best hyperparameters for ArTwitter and Syria_Tweets datasets across TF-IDF with N-gram models.

Classifier	Hyperparameter	Unigram		Bigram		Trigram	
		ArTwitter	Syria	ArTwitter	Syria	ArTwitter	Syria
Naive Bayes (NB)	Alpha	0.0340	0.0010	0.0275	0.0010	0.1896	0.0010
SVM	C	0.9635	3.6975	0.4667	105.7621	0.6839	105.7621
	Gamma	0.0015	0.0271	0.0570	0.0447	0.1	0.0447
	Kernel	Linear	Linear	Linear	Rbf	Linear	Rbf
KNN	Metric	Minkowski	manhattan	Minkowski	Manhattan	Euclidean	Manhattan
	n_neighbors	12	2	14	2	4	2
	Weights	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform
Decision Tree (DT)	MAX_depth	39	35	50	21	50	32
	Min_samples_leaf	1	1	1	1	1	1
	Min_samples_split	20	2	19	2	15	3

TABLE 4 Best hyperparameters using Word2Vec for ArTwitter and Syria_Tweets datasets.

Classifier	Hyper-parameter	ArTwitter value	Syria_Tweets value
SGD	Alpha	1e-06	0.000563
	eta0	1.0225	0.0174
	Learning_rate	Invscaling	Adaptive
	Loss	Log_loss	Log_loss
	Max_iter	3251	1000
	Penalty	Elasticnet	l1
	Tol	0.01	1.41e-05
Logistic regression (LR)	C	0.5023	11185.625
	Penalty	l2	l2
	Solver	Liblinear	Liblinear
Support vector machine (SVM)	C	25.8455	30.0
	Gamma	0.1877	0.15
	Kernel	rbf	rbf
K-Nearest Neighbors (KNN)	Metric	Minkowski	Manhattan
	n_neighbors	6	2
	Weights	Uniform	Uniform
Random Forest (RF)	Bootstrap	False	False
	Max_depth	50	45
	Max_features	Log2	Sqrt
	Min_samples_leaf	1	1
	Min_samples_split	2	2
	n_estimators	500	500

hyperparameters optimized for ArTiwtter data set in a much smaller learning rate initialization (`eta0`) and used a "invscaling" learning schedule with a `elasticnet` penalty. While Syria_Tweets hyperparameters optimized to an "adaptive" schedule and an "l1" penalty,An adaptive learning rate helped keep the model's training on a stable and efficient path. At the same time, the L1 penalty was great at promoting feature sparsity, which let the model focus on the most important predictors and tune out the noise in the data, preventing it from just memorizing the training examples. . However, the SVM classifier shared the same RBF kernel across both datasets. The KNN classifier revealed greater variation: ArTwitter favored six neighbors and the Minkowski distance, while Syria_Tweets performed best with just two neighbors and the Manhattan distance, indicating that Syria_Tweets required tighter local decision boundaries.

4.2 Results

Table 5 presents the performance of both individual and ensemble learning models using TF-IDF with unigram, bigram, and trigram representations on the ArTwitter dataset. The results demonstrate that unigram features consistently outperform both bigram and trigram configurations. Among the individual classifiers, Naive Bayes (NB) achieved the highest accuracy of 89.27 and 89.00% F1-score with unigrams, followed closely by SVM with 88.01% accuracy and 88.0% F1-score. Notably, all ensemble models outperformed the individual classifiers across the different n-gram representations. The V1 ensemble model (comprising NB, SVM, and DT) achieved the highest accuracy of 90.22 and 90.00% F1-score with unigram features, highlighting the effectiveness of combining diverse classifiers.

For the balanced Syria_Tweets dataset, Table 6 reveals more consistent performance across all n-gram representations. Both NB and SVM classifiers showed strong results, achieving 81.47 and 81.76% accuracy, respectively, using unigram features and 80.69% and 81.25 F1-score. However, ensemble models again demonstrated superior performance. In particular, the V4 ensemble (SVM, DT, and KNN) achieved the highest accuracy of 83.82 and 83.33% F1-score with bigram features, indicating that ensemble learning can

TABLE 5 Performance across unigram, bigram, and trigram features on the ArTwitter dataset.

Classifier	Unigram				Bigram				Trigram			
	Acc()	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
NB	89.27	89.00	89.00	89.00	87.70	88.00	88.00	88.00	86.75	87.00	87.00	87.00
SVM	88.01	88.00	88.00	88.00	86.75	87.00	87.00	87.00	84.54	85.00	85.00	85.00
K-NN	83.91	84.00	84.00	84.00	81.39	82.00	81.00	81.00	80.44	80.00	80.00	80.00
DT	79.18	80.00	79.00	79.00	81.70	82.00	82.00	82.00	81.70	82.00	82.00	82.00
V1 (NB, SVM, DT)	**90.22**	90.00	90.00	90.00	**89.27**	89.00	89.00	89.00	**88.96**	89.00	89.00	89.00
V2 (NB, SVM, K-NN)	89.91	90.00	90.00	90.00	87.38	87.00	87.00	87.00	83.60	84.00	84.00	83.00
V3 (NB, DT, K-NN)	88.01	88.00	88.00	88.00	87.70	88.00	88.00	88.00	86.75	87.00	87.00	87.00
V4 (SVM, DT, K-NN)	88.01	88.00	88.00	88.00	87.38	88.00	87.00	87.00	85.17	0.85	85.00	85.00

Bold values indicate the best performance of each model.

TABLE 6 Performance across unigram, bigram, and trigram features on the Syria_Tweets dataset.

Classifier	Unigram				Bigram				Trigram			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
NB	81.47	80.43	81.47	80.69	81.18	80.06	81.18	80.33	81.76	80.79	81.76	81.05
SVM	81.76	80.99	81.76	81.25	80.88	79.88	80.88	80.19	81.18	80.15	81.18	80.44
K-NN	80.29	79.36	80.29	79.69	79.71	79.00	79.71	79.29	78.82	80.18	78.82	79.36
DT	79.41	77.99	79.41	78.37	79.71	77.94	79.71	78.07	80.00	79.38	80.00	79.64
V1 (NB, SVM, DT)	**83.53**	82.64	83.53	82.14	82.65	81.54	82.65	81.24	**83.24**	82.26	83.24	81.88
V2 (NB, SVM, K-NN)	82.65	81.66	82.65	81.82	82.06	81.08	82.06	81.31	81.18	80.15	81.18	80.44
V3 (NB, DT, K-NN)	82.35	81.44	82.35	81.66	82.94	82.44	82.94	82.63	80.88	81.13	80.88	81.00
V4 (SVM, DT, K-NN)	82.65	81.66	82.65	81.82	**83.82**	83.14	83.82	83.33	82.35	82.21	82.35	82.28

Bold values indicate the best performance of each model.

TABLE 7 Individual classifiers and ensemble performance using word embeddings on ArTwitter and balanced Syria_Tweets datasets.

Classifier	ArTwitter				Syria_Tweets			
	Accuracy (%)	Precision	Recall	F1-score	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
SGD	89.27	89.00	89.00	89.00	79.12	81.69	79.12	79.97
LR	**90.54**	91.00	91.00	91.00	75.59	78.89	75.59	76.70
SVM	**90.54**	91.00	91.00	91.00	80.85	81.00	80.85	80.90
K-NN	84.20	86.00	83.00	84.50	76.18	81.33	76.18	77.57
RF	88.96	89.00	89.00	89.00	81.76	80.49	81.76	80.48
V1 (SGD, LR, SVM)	92.11	92.00	92.00	92.00	82.50	83.00	82.50	82.60
V2 (SGD, LR, K-NN)	91.10	91.80	91.10	91.30	79.41	81.85	79.41	80.22
V3 (SGD, LR, RF)	91.17	91.00	91.00	91.00	79.12	80.57	79.12	79.68
V4 (SGD, SVM, RF)	**92.43**	92.00	92.00	**92.00**	82.10	82.40	82.10	82.20
V5 (SGD, K-NN, RF)	91.85	91.70	91.60	91.65	**83.82**	83.89	83.82	**83.86**
V6 (LR, SVM, RF)	91.48	92.00	91.00	91.00	82.60	82.90	82.60	82.70
V7 (LR, K-NN, RF)	91.00	91.30	91.00	91.10	83.24	83.17	83.24	83.20

Bold values indicate the best performance of each model.

TABLE 8 Comparison of accuracy between previous and our study on ArTwitter Dataset.

Reference	Approach	Accuracy	F1 score
Al-Saqqa et al. (2018)	Ensemble machine learning (voting of KNN, SVM, DT, NB)	84.4% (SVM individually)	84.0%
Saleh et al. (2022)	Stacked deep learning (RNN, LSTM, GRU + SVM meta-learner)	83.12%	82.8%
Aladeemy et al. (2024)	Machine learning (SVM with BoW Unigram)	90.3%	90.3%
Our approach	Ensemble machine learning (voting of SGD, SVM, RF)	**92.43%**	92.0%

Bold values indicate the best performance of each model.

TABLE 9 Comparison of F1 score between previous and our study on Syria_Tweets Dataset.

Reference	Approach	F1-score
Al-Azani and El-Alfy (2017)	Ensemble machine learning (stacking)	63.95%
El-Alfy and Al-Azani (2020)	Machine learning (SGD classifier)	70.7%
Our approach	Ensemble machine learning (voting of SGD, K-NN, RF)	**83.86%**

Bold values indicate the best performance of each model.

capture richer contextual information and provide more robust classification in complex datasets.

Finally, Table 7 presents the results of individual and ensemble models using word embeddings on both datasets. Across the board, word embeddings improved the performance of all models compared to the TF-IDF-based representations. Ensemble models significantly outperformed individual classifiers in both datasets. On the ArTwitter dataset, the V4 ensemble (SGD, SVM, RF) achieved the highest accuracy of 92.43% 92.00% F1-score. On the Syria_Tweets dataset, the best performance was obtained by the V5 ensemble (SGD, KNN, RF), which reached an accuracy of 83.82% 83.86% F1-score. These findings confirm the effectiveness of combining rich semantic features with ensemble strategies to enhance classification accuracy in Arabic social media text.

4.3 Error analysis

To gain a clearer picture of where our model falls short, we looked closely at tweets it misclassified in both datasets. Three main patterns stood out.

First, sarcasm and irony often tripped the model. Tweet التعدد المال من كثيييييييير إلى يحتاج ولكن جداً جميل which means in English "Polygamy is very beautiful, but it requires a lot of money." used positive wording to express criticism, usually labeled incorrectly because the model lacked any mechanism to detect sarcasm. Second, dialectal variation posed a challenge. Like tweet " وايد تتحمس لا سهل مب الشي لأن (العدل) بموضوع" which means in English "Don't get too excited about the topic of it's not easy." The tweet contained regional expressions, particularly from Gulf "مب,وايد," that were not well captured in the embeddings. Words that carried a negative tone in one dialect could be interpreted as neutral in another, leading to incorrect predictions.

Finally, mixed sentiment such as دايما المرأة حقوق مع انا نسوية نفسي عن أقول مستحيل لكن which means in English "I am always for women's rights, but it is impossible for me to call myself a feminist." The tweet conveyed both positive and negative feelings about different entities were often reduced to a single overall sentiment, which meant losing important nuances. A more fine-grained, aspect-based approach would likely handle such cases better.

4.4 Comparison of the proposed model with existing work

To compare the proposed approach with the most relevant previous studies, Table 8 presents the results of selected works. Al-Saqqa et al. (2018) applied ensemble learning using traditional machine learning classifiers and achieved an accuracy of 84.4%. Saleh et al. (2022) employed a stacking ensemble method that integrated deep learning architectures such as RNN, LSTM, and GRU, with an SVM meta-classifier, achieving 83.12% accuracy. The most recent work by Aladeemy et al. (2024) attained 90.3% accuracy using a standalone SVM classifier with unigram features. In contrast, our proposed approach—based on hard voting ensemble learning that combines SGD, SVM, and Random Forest classifiers with pre-trained word embeddings—achieved the highest accuracy of 92.43%, demonstrating its superior performance in Arabic sentiment classification.

However, related to the Syria_Tweet data set, the F1-score is used because the accuracy isn't available. Table 9 compares our results with the most related previous work. As shown, our approach with ensemble voting (SGD, K-NN, RF) improved the performance of analyzing the sentiment of the dataset. The ensemble stacking approach was applied on the same data set by Al-Azani and El-Alfy (2017), and the F1-score achieved is 63.95%. While a traditional ML algorithm, which is SGD, was applied by El-Alfy and Al-Azani (2020) and achieved a 70.7% F1-score.

5 Conclusion and future direction

The objective of this study was to investigate multiple methodologies for feature extraction specifically tailored for Arabic sentiment analysis. Our focus was directed toward analyzing three distinct types of n-gram features—namely, unigram, bigram, and trigram—alongside leveraging a pre-trained Word2Vec word embedding model. A diverse machine learning algorithms was

employed in our analysis, including Support Vector Machines (SVM), k-Nearest Neighbors (K-NN), Stochastic Gradient Descent (SGD), Logistic Regression (LR), and Random Forest (RF). Additionally, we implemented ensemble techniques based on hard voting.

The experimental investigations were conducted utilizing two distinct datasets: the balanced ArTwitter dataset and the significantly imbalanced Syria_Tweets dataset. To address the issue of class imbalance present in the Syria_Tweets dataset, the Synthetic Minority Oversampling Technique (SMOTE) was applied during the training phase.

Our results indicated that Naïve Bayes (NB) achieved the highest accuracy rate of 89.79 and 89% F1-score on the ArTwitter dataset when unigram features were employed. Conversely, the Support Vector Machine (SVM) achieved an accuracy rate of 81.76 and 81.25% F1-score on the Syria_Tweets dataset, with SVM excelling with unigram features and NB performing optimally with trigram features. Notably, the hard voting ensemble containing Naïve Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT) utilizing unigram features outperformed others on the ArTwitter dataset, achieving an accuracy of 90.22% and 90% F1-score. Meanwhile, the hard voting ensemble combining SVM, DT, and K-Nearest Neighbors (K-NN) attained superior results on the Syria_Tweets dataset with an accuracy of 83.82% and 83.33% F1-score when employing bigram features. However, average weighted pretrained word embedding achieved superior results on both datasets with the ensemble approach; hard voting (SGD, SVM, and RF) achieved 92.43% accuracy and 92% F1-score on ArTwitter Dataset. While hard voting (SGD, KNN, and RF) achieved 83.82% accuracy and 83.86% F1-score on Syris_tweet dataset.

The outcomes of this research suggest that leverage pretrained word embedding in representing the data can significantly enhance model performance and that ensemble approaches contribute to a more robust overall system. Looking ahead, there is potential for employing transformer-based models, which provide deep contextualized embeddings, thereby further optimizing performance. The exploration of novel data balancing methodologies could advance the efficacy of model operation.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Ethics statement

Ethical approval was not required for the study involving human data in accordance with the local legislation and institutional requirements. Written informed consent was not required for either participation in the study or for the publication of potentially/indirectly identifying information, in accordance with the local legislation and institutional requirements. The social media data was accessed and analyzed in accordance with the platform's terms of use and all relevant institutional/national regulations.

Author contributions

AJ: Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing. IB: Investigation, Software, Writing – original draft, Writing – review & editing. PM: Data curation, Funding acquisition, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work was supported by grant PID2023-148577OB-C21 (Human-Centered AI: User-Driven Adapted Language Models-HUMAN AI) by MICIU/AEI/10.13039/501100011033 and by FEDER/UE.

Acknowledgments

Thanks to Palestine Technical University - Kadoorie support and grant PID2023-148577OB-C21 (Human-Centered AI: User-Driven Adapted Language Models-HUMAN AI) by MICIU/AEI/10.13039/501100011033 and by FEDER/UE.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that Gen AI was used in the creation of this manuscript. ChatGPT was used for language refinement and write-proofing purposes only.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Abdulla, N. A., Ahmed, N. A., Shehab, M. A., and Al-Ayyoub, M. (2013). "Arabic sentiment analysis: Lexicon-based and corpus-based," in *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)* (Amman: IEEE), 1–6. doi: 10.1109/AEECT.2013.6716448

Aladeemy, A. A., Aldhyani, T. H., Alzahrani, A., Alzahrani, E. M., Khalaf, O. I., Alsubari, S. N., et al. (2024). Machine learning algorithms for predicting and analyzing arabic sentiment. *SN Comput. Sci.* 5, 1–10. doi: 10.1007/s42979-024-03494-w

Al-Azani, S., and El-Alfy, E.-S. M. (2017). Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text. *Procedia Comput. Sci.* 109, 359–366. doi: 10.1016/j.procs.2017.05.365

Alosaimi, W., Saleh, H., Hamzah, A. A., El-Rashidy, N., Alharb, A., Elaraby, A., et al. (2024). Arabbert-lstm: improving arabic sentiment analysis based on transformer model and long short-term memory. *Front. Artif. Intell.* 7:1408845. doi: 10.3389/frai.2024.1408845

Alotaibi, A., Nadeem, F., and Hamdy, M. (2025). Weakly supervised deep learning for Arabic tweet sentiment analysis on education reforms: leveraging pre-trained models and LLMS with snorkel. *IEEE Access* 13, 30523–30542. doi: 10.1109/ACCESS.2025.3541154

Alsalem, A. Y., and Abudalfa, S. I. (2024). "Empirical analysis for arabic target-dependent sentiment classification using LLMS," in *2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)* (Sakhir: IEEE), 170–176. doi: 10.1109/3ict64318.2024.10824564

Al-Saqqa, S., Obeid, N., and Awajan, A. (2018). "Sentiment analysis for arabic text using ensemble learning," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)* (Aqaba: IEEE), 1–7. doi: 10.1109/AICCSA.2018.8612804

Alwakid, G., Osman, T., and Hughes-Roberts, T. (2017). Challenges in sentiment analysis for arabic social networks. *Procedia Comput. Sci.* 117, 89–100. doi: 10.1016/j.procs.2017.10.097

Assiri, A., Emam, A., and Al-Dossari, H. (2018). Towards enhancement of a lexicon-based approach for saudi dialect sentiment analysis. *J. Inf. Sci.* 44, 184–202. doi: 10.1177/0165551516688143

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Cambridge, MA: O'Reilly Media, Inc.

Bottou, L. (2010). "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers* (Cham: Springer), 177–186. doi: 10.1007/978-3-7908-2604-3_16

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. doi: 10.1023/A:1010933404324

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* 16, 321–357. doi: 10.1613/jair.953

Church, K. W. (2017). Word2vec. *Nat. Lang. Eng.* 23, 155–162. doi: 10.1017/S1351324916000334

Cortes, C. (1995). Support-vector networks. *Mach. Learn.* 20, 273–297. doi: 10.1023/A:1022627411411

Cox, D. R. (1958). The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B: Stat. Methodol.* 20, 215–232. doi: 10.1111/j.2517-6161.1958.tb00292.x

Deng, S., Huang, X., Zhu, Y., Su, Z., Fu, Z., Shimada, T., et al. (2023). Stock index direction forecasting using an explainable extreme gradient boosting and investor sentiments. *N. Am. J. Econ. Financ.* 64:101848. doi: 10.1016/j.najef.2022.101848

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). Pattern *Classification*, 2nd Edn. Hoboken, NJ: Wiley.

Duridi, T., Atwe, L., Jaber, A., Daraghmi, E., and Martínez, P. (2025). "Detection of propaganda and bias in social media: a case study of the Israel-Gaza war (2023)," in *2025 International Conference on New Trends in Computing Sciences (ICTCS)* (Amman: IEEE), 204–210. doi: 10.1109/ICTCS65341.2025.10989479

El-Alfy, E.-S. M., and Al-Azani, S. (2020). Empirical study on imbalanced learning of arabic sentiment polarity with neural word embedding. *J. Intell. Fuzzy Syst.* 38, 6211–6222. doi: 10.3233/JIFS-179703

Elshakankery, K., and Ahmed, M. F. (2019). Hilatsa: a hybrid incremental learning approach for arabic tweets sentiment analysis. *Egypt. Inf. J.* 20, 163–171. doi: 10.1016/j.eij.2019.03.002

Fouad, M. M., Mahany, A., Aljohani, N., Abbasi, R. A., and Hassan, S.-U. (2020). Arwordvec: efficient word embedding models for arabic tweets. *Soft Comput.* 24, 8061–8068. doi: 10.1007/s00500-019-04153-6

Gamal, D., Alfonse, M., El-Horbaty, E.-S. M., and Salem, A.-B. M. (2019). Twitter benchmark dataset for arabic sentiment analysis. *Int. J. Mod. Educ. Comput. Sci.* 11:33. doi: 10.5815/ijmecs.2019.01.04

Ghallab, A., Mohsen, A., and Ali, Y. (2020). Arabic sentiment analysis: a systematic literature review. *Appl. Comput. Intell. Soft Comput.* 2020:7403128. doi: 10.1155/2020/7403128

Grover, P., Gohil, V., Goel, B., Veeramani, H., Shah, S. B., Jain, S. R., et al. (2025). "Polimeme: exploring offensive meme propagation in the Israel-Palestine conflict," in *Companion Proceedings of the ACM on Web Conference 2025, WWW '25* (New York, NY: Association for Computing Machinery), 1969–1978. doi: 10.1145/3701716.3718387

He, H., and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* 21, 1263–1284. doi: 10.1109/TKDE.2008.239

Jaber, A., and Martínez, P. (2022). Disambiguating clinical abbreviations using a one-fits-all classifier based on deep learning techniques. *Methods Inf. Med.* 61(S 01), e28-e34. doi: 10.1055/s-0042-1742388

Jaber, A., and Martínez, P. (2023). "Ptuk-hulat at araieval shared task fine-tuned distilbert to predict disinformative tweets," in *Proceedings of ArabicNLP 2023* (Singapore: IEEE), 525–529. doi: 10.18653/v1/2023.arabicnlp-1.50

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext. zip: compressing text classification models. *arXiv [preprint]*. arXiv:1612.03651. doi: 10.48550/arXiv.1612.03651

Jurafsky, D., and Martin, J. H. (2009). *Speech and Language Processing*, 2nd Edn. Upper Saddle River, NJ: Pearson Prentice Hall.

Kumar, V., Recupero, D. R., Riboni, D., and Helaoui, R. (2020). Ensembling classical machine learning and deep learning approaches for morbidity identification from clinical notes. *IEEE Access* 9, 7107–7126. doi: 10.1109/ACCESS.2020.3043221

Mansour, O., Aboelela, E., Talaat, R., and Bustami, M. (2025). Transformer-based ensemble model for dialectal arabic sentiment classification. *PeerJ Comput. Sci.* 11:e2644. doi: 10.7717/peerj-cs.2644

Mataoui, M., Zelmati, O., and Boumechache, M. (2016). A proposed lexicon-based sentiment analysis approach for the vernacular algerian arabic. *Res. Comput. Sci.* 110, 55–70. doi: 10.13053/rcs-110-1-5

Matrane, Y., Benabbou, F., and Sael, N. (2023). A systematic literature review of arabic dialect sentiment analysis. *J. King Saud Univ.-Comput. Inf. Sci.* 35:101570. doi: 10.1016/j.jksuci.2023.101570

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv [preprint]*. arXiv:1301.3781. doi: 10.48550/arXiv.1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 26.

Mohammad, S. M., Salameh, M., and Kiritchenko, S. (2016). How translation alters sentiment. *J. Artif. Intell. Res.* 55, 95–130. doi: 10.1613/jair.4787

Nafis, N. S. M., and Awang, S. (2021). An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification. *IEEE Access* 9, 52177–52192. doi: 10.1109/ACCESS.2021.3069001

Obiedat, R., Al-Darras, D., Alzaghoul, E., and Harfoushi, O. (2021). Arabic aspect-based sentiment analysis: a systematic literature review. *IEEE Access* 9, 152628–152645. doi: 10.1109/ACCESS.2021.3127140

Onan, A., Korukoğlu, S., and Bulut, H. (2016). A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Syst. Appl.* 62, 1–16. doi: 10.1016/j.eswa.2016.06.005

Pennington, J., Socher, R., and Manning, C. D. (2014). "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha: IEEE), 1532–1543. doi: 10.3115/v1/D14-1162

Rahma, A., Azab, S. S., and Mohammed, A. (2023). A comprehensive survey on arabic sarcasm detection: approaches, challenges and future trends. *IEEE Access* 11, 18261–18280. doi: 10.1109/ACCESS.2023.3247427

Rane, N., Choudhary, S. P., and Rane, J. (2024). Ensemble deep learning and machine learning: applications, opportunities, challenges, and future directions. *Stud. Med. Health Sci.* 1, 18–41. doi: 10.48185/smhs.v1i2.1225

Sahmoud, S., Abudalfa, S., and Elmasry, W. (2022). *At-Odtsa: A Dataset of Arabic Tweets for Open Domain Targeted Sentiment Analysis*. doi: 10.12785/ijcds/1101105

Saleh, H., Mostafa, S., Alharbi, A., El-Sappagh, S., and Alkhalifah, T. (2022). Heterogeneous ensemble deep learning model for enhanced arabic sentiment analysis. *Sensors* 22:3707. doi: 10.3390/s22103707

Snoek, J., Larochelle, H., and Adams, R. P. (2012). "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems, Vol.* 25, 2951–2959.

Taghva, K., Elkhoury, R., and Coombs, J. (2005). "Arabic stemming without a root dictionary," in *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II* (Washington, DC: IEEE), 152–157. doi: 10.1109/ITCC.2005.90

Tiwari, N. K., and Arora, H. (2025). "Sentiment analysis and forecasting for improved business performance in E-commerce using machine learning algorithms," in *2025 International Conference on Electronics and Renewable Systems (ICEARS)* (Tuticorin), 14871491. doi: 10.1109/ICEARS64219.2025.10940994

Tubishat, M., Abushariah, M. A., Idris, N., and Aljarah, I. (2019). Improved whale optimization algorithm for feature selection in arabic sentiment analysis. *Appl. Intell.* 49, 1688–1707. doi: 10.1007/s10489-018-1334-8

Yang, W., Yuan, T., and Wang, L. (2020). Micro-blog sentiment classification method based on the personality and bagging algorithm. *Future Internet* 12:75. doi: 10.3390/fi12040075

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 18 September 2025
DOI 10.3389/frai.2025.1661789

OPEN ACCESS

EDITED BY
Shadi Abudalfa,
King Fahd University of Petroleum and Minerals, Saudi Arabia

REVIEWED BY
Anas Alkhofi,
King Faisal University, Saudi Arabia
Baligh Babaali,
University of Medea, Algeria
Fatima Zahra El Idrysy,
Sidi Mohamed Ben Abdellah University, Morocco

*CORRESPONDENCE
Sa'Ed Abed
✉ s.abed@ku.edu.kw

RECEIVED 08 July 2025
ACCEPTED 29 August 2025
PUBLISHED 18 September 2025

CITATION
Beidas A, Mohi K, Ghaddar F, Ahmad I and Abed S (2025) Cross-dialectal Arabic translation: comparative analysis on large language models.
Front. Artif. Intell. 8:1661789.
doi: 10.3389/frai.2025.1661789

Cross-dialectal Arabic translation: comparative analysis on large language models

Ayah Beidas, Kousar Mohi, Fatme Ghaddar, Imtiaz Ahmad and Sa'Ed Abed*

Computer Engineering Department, College of Engineering and Petroleum, Kuwait University, Kuwait City, Kuwait

Introduction: Exploring Arabic dialects in Natural Language Processing (NLP) is essential to understand linguistic variation and meet regional communication demands. Recent advances in Large Language Models (LLMs) have opened up new vistas for multilingual communication and text generation.

Methods: This paper investigates the performance of GPT-3.5, GPT-4, and Bard (Gemini) on the QADI and MADAR datasets, while GPT-5 was evaluated exclusively on MADAR encompassing over 15 different countries. Several metrics have been used in the evaluation, such as cosine similarity, universal similarity encoder, sentence BERT, TER, ROUGE, and BLEU. In this study, different prompting techniques were used: zero-shot and few-shot. Zero-shot was employed for all dialects, and few-shot was employed only for the least translation performance dialect, Tunisian.

Results: Analysis revealed that in the QADI dataset, GPT-4 significantly outperformed others in translating MSA to DA, with ANOVA tests showing strong significance ($p < 0.05$) in most metrics, except for BLEU and TER where it does not show significance, indicating comparable translation performance among models. Furthermore, GPT-4 was highest in semantic similarity compared to GPT-3.5 and Bard (Gemini), 0.66, 0.61, and 0.63, respectively. GPT-4 was the best in identifying overlapping sentences (i.e., those where the source and target are identical) with a combined average of 0.41 in BLEU and ROUGE-L. All LLMs scored TER values between 6% and 25%, indicating generally good translation quality. However, GPT models, especially GPT-5, responded better to prompting and translation to Levant countries compared to Bard (Gemini). For the MADAR dataset, no significant translation differences were observed in sentence-BERT, ROUGE-L, and TER, while differences are identified in cosine similarity, BLEU, and universal similarity encoder metrics. Therefore, GPT-5 is the top performer in identifying sentence overlaps measured by BLEU and ROUGE-L (combined average 0.37).

Discussion: The few-shot approach did not show a significant improvement in translation performance, especially for GPT-4 and Bard (Gemini), while GPT-3.5 performed consistently. Zero-shot prompts were effective across dialects, while few-shot prompting, applied to the weakest-performing dialect (Tunisian), did not yield improvement. GPT-4 and Bard performed worse under this set-up, while GPT-3.5 remained consistent.

KEYWORDS

language models, GPT 3.5, GPT 4, GPT 5, Bard (Gemini), Arabic language, dialects

1 Introduction

In recent years, new horizons for multilingual communication, translation tasks, and text generation have been widely witnessed due to the advances made in large language models (LLMs) (Shaikh et al., 2023). Models such as GPT, developed by OpenAI and Google Bard (Gemini), have shown promising developments in this field (Kasneci et al., 2023). Such models have demonstrated outstanding skills in handling diverse languages and dialects with the influential role of deep learning techniques and the processing of massive volumes of textual data. According to studies conducted in 2019 by Ethnologue (Eberhard et al., 2019), the total number of dialects spoken around the globe is expected to be 7,111, where a majority of these dialects are found on the Internet through platforms such as Facebook, X, and blog posts through user interactions (Salloum and Habash, 2012). Therefore, with the availability of systems that deal with different languages and dialects, a major shift in focus has been witnessed in literature to bring dialects together by enhancing proper machine learning translation systems (Sghaier and Zrigui, 2020).

Arabic is one of the languages known for its diversity in linguistics, which includes various dialects from different countries all over the Arab world. Notably, Dialectal Arabic (DA) consists of different Arabic dialects. It is an informal language that is used in daily life and social media platforms in contrast with Modern Standard Arabic (MSA), also known as "Fushaa," which is used in formal communications (Harrat et al., 2019). Hence, making the comprehension of different dialects presents a greater challenge compared to MSA, due to its regional variability, especially in the applications of cross-dialect communications, and in sectors such as education and content localization (Sghaier and Zrigui, 2020).

Large language models (LLMs) are a vital approach to understand and enhance the language intelligence of devices (Hadi et al., 2023). LLMs can react to free-text queries without being specifically trained in the activity at hand, which has sparked both excitement and skepticism among researchers regarding their application (Hadi et al., 2023). Models such as OpenAI GPT and Google Bard (Gemini) are examples of LLMs, where they are trained on enormous volumes of text data and can generate human-like prose, answer questions, and perform other language-related tasks with great accuracy (Kasneci et al., 2023). To begin with, OpenAI GPT is a decoder-based, generative pre-trained LLM. It employs an auto-regressive language model that allows sequential text generation. Among many of the advantages present in GPT, one main advantage is that it is a multilingual model, including the Arabic language (Alyafeai et al., 2023). However, it is not an open-access model and is not free of cost. Therefore, developers and researchers have to pay a certain amount based on the number of tokens used per request and the type of model to be used for fine-tuning (Steele, 2023). As for Bard (Gemini), it is developed by Google and is also multilingual; in total, it contains 41 languages (Kadaoui et al., 2023). Similar to GPT, Bard (Gemini) has a certain cost based on the number of tokens used per request and the type of model to be used (Kadaoui et al., 2023). Hence, by analyzing their differences and similarities, a comparison between both models is performed to assist systems in easily translating dialects and achieve human-like reading and writing, building on the comprehensive overview of LLM capabilities by Hadi et al. (2023).

Researchers have been using these models in analyzing various NLP tasks, such as psychological studies of sentiments using GPT (Kheiri and Karimi, 2023). In addition, comparisons with other models such as Bidirectional Encoder Representations from Transformers (BERT) (Zhang et al., 2020) and Bidirectional Long-Form Overlap for Optimizing Multilingual and zero-shot (BLOOMZ) (Yong et al., 2022) have been made in contexts such as translation efficiencies using different languages (Bhat et al., 2023). On the other hand, comparisons between GPT 3.5, GPT 4, and Bard (Gemini) have been made regarding their machine translation (MT) proficiency across 10 varieties of Arabic (Kadaoui et al., 2023). Their analysis shows that LLMs may encounter challenges with dialects for which minimal public datasets exist, but on average, they are better translators of dialects than existing commercial systems. In a similar vein, GPT 4 outperformed Bard (Gemini) in dialect-based commercial systems and different supervised baselines employing zero-shot prompts.

Originally, researchers' main focus was to address the translation of English to Arabic and vice versa (Khoshafah, 2023). However, more recently, researchers have been studying the influence of MSA on the similarity between dialects spoken, as was done by Abu-Haidar (2011) in Baghdad, and vice versa, where researchers study the translation from DA to MSA. For instance, Sghaier and Zrigui (2020) performed a similar study in 2020 where an MT system that translates Tunisian dialect text to MSA using a rule-based approach showed promising results for their proposed solution. Since OpenAI GPT released different models with different versions, researchers have focused on having a comparison between these different versions, where Alyafeai et al. (2023) have compared some of these models, such as GPT 3.5 and GPT 4, on seven distinct Arabic NLP tasks and found that GPT 4 outperforms GPT 3.5 on five NLP tasks. GPT 3.5 and GPT 4 performances were also studied using the Tunisian, Jordanian, and English languages, and the study results highlight a critical dialectical performance gap in GPT, underlining the need to enhance linguistic and cultural diversity in AI models' development, particularly for health-related content (Sallam and Mousa, 2024).

The purpose of this study is to compare the performance of four language models, GPT (versions 3.5, 4, and 5) and Bard (Gemini), in translating a wide corpus of MSA to DA. This novel study bridges a significant gap in understanding model performance across diverse linguistic situations by including a wide corpus of dialects, consisting of over 15 Arabic dialects, in the analysis while evaluating several metrics. Furthermore, two different datasets will be used to further strengthen the analysis using different prompting techniques (zero-shot and few-shot). To explore whether these techniques enhance the quality of dialect translation, zero-shot will be applied to all countries, whereas few-shot will be applied to the weakest country.

This study sheds light on the adaptability and efficiency of these models through careful metric assessments, which is critical for expanding NLP applications in various Arabic-speaking regions. Two datasets are used in this study the first is the Qatar Computing Research Institute (QCRI) Arabic

Dialects Identification (QADI) dataset, which contains 18 different countries with their own dialects. QADI contains over 500,000 tweets from social media platforms, spanning 18 different Arabic dialects (Abdelali et al., 2020). Second, the Multi-Arabic Dialect Applications and Resources (MADAR) corpus dataset is used, which includes a large parallel corpus of 25 Arabic city dialects in the travel domain. These are the most popular datasets adapted for studies with Arabic dialects.

This research study aims to answer the following questions:

- How efficient are GPT 3.5, GPT 4, GPT 5, and Bard (Gemini) in translating MSA to different DA in terms of different performance metrics, such as cosine similarity, semantic universal encoder, sentence BERT, similarity encoder, translation error rate (TER), recall-oriented understudy for gisting evaluation (ROUGE), bilingual evaluation understudy (BLEU), and analysis of variance (ANOVA)?
- How consistent is the LLM performance in the MSA translation to different DAs? (e.g., Levantine vs. Gulf vs. Maghrebi)
- How do prompting techniques (zero-shot vs. few-shot) and external factors like sentence length impact the translation accuracy of LLMs?

The main contribution of this study could be summarized as follows:

- It sheds light on the strengths and drawbacks of the GPT 3.5, GPT 4, GPT 5, and Bard (Gemini) models in dealing with DA differences by analyzing their translation quality and accuracy (measured by metrics) and consistency/reliability, across various dialects from MSA. Hence, exploring how LLMs handle dialectal diversity in Arabic.
- It employs various prompt analysis techniques to evaluate the performance of GPT 3.5, GPT 4, GPT 5, and Bard (Gemini), aiming to understand the specific conditions under which each model excels.
- The study's findings fill in a significant gap in research on MSA to dialect translation using LLMs by using a wide corpus of Arabic dialect translations and analyzing GPT 3.5/4/5, and Bard (Gemini) in translating various dialects using different prompting techniques (zero-shot and few-shot).

Therefore, the study relies on it being the first to offer a comprehensive evaluation of LLMs in translating MSA to a wide range of dialects using QADI and MADAR datasets. Moreover, the evaluation of GPT 3.5, GPT 4, GPT 5, and Bard (Gemini) contributes to fine-tuning and developing inclusive NLP tools to serve a larger Arabic-speaking population with diverse dialects. It identifies the strengths and weaknesses of LLMs in different DAs by translation from MSA. Such insights are essential for the development of inclusive NLP tools that can effectively utilize MSA and different DAs in spoken Arabic to enhance digital accessibility and communication. To the best of our knowledge, we are the first study comparing prominent LLMs specially GPT 5 on MT task from MSA to DA over 15 countries.

The remainder of this study is organized as follows: The related work is described in Section 2, and the proposed methodology is detailed in Section 3. Experimental results are reported and analyzed in Section 4. Finally, the concluding remarks and future research directions are described in Section 5.

2 Related work

This section highlights the challenges of processing the Arabic language and its dialects in Section 2.1, followed by Section 2.2, which explains and explores different LLMs and Section 2.3 describes various MT approaches.

2.1 Challenges for processing Arabic and its dialects

Contemporary Arabic consists of different varieties such as MSA, the official language of the Arab world that is used in formal settings, and dialects of different countries that are commonly used in different informal contexts. In general, Arabic is a complex language with a rich inflectional morphology expressed both templatically and affixationally, as well as various attachable clitic classes (Wright and Caspari, 2011). The dialects of different countries differ from MSA in terms of phonology, morphology, and, to some extent, syntactically, where the differences are based on the presence of clitics and affixes, unlike MSA, are widely used (Salloum and Habash, 2012). Dialects are considered to share all of MSA's problems when it comes to NLP (e.g., optional diacritics and spelling inconsistencies). However, adding to these problems, the absence of standard orthographies for the dialects and their diverse variants, which in turn pose additional issues (Guellil et al., 2021). In addition, there are very few Arabic dialects of English corpora and even fewer dialects of MSA parallel corpora, which makes the number of morphological analyses and tools for these dialects constrained (Salloum and Habash, 2012).

These linguistic challenges pose different difficulties for LLMs in MT. Unlike the English language, which dominates the training of most LLMs, different Arabic dialects are widely underrepresented (Alyafeai et al., 2023; Khondaker et al., 2023). Research papers comparing LLM performance between different languages such as English and Arabic address this gap and confirm it by showing that LLMs achieve better scores in English translation than in Arabic (Peng et al., 2023). Furthermore, within Arabic itself, MSA is better handled in LLMs than in different dialects (Kadaoui et al., 2023). These demonstrate that the wide variation of dialects in the Arabic language and their complexities pose a challenge in MT. Hence, understanding of LLMs ability to translate MSA to different dialects along with the strengths and weaknesses of LLMs in different DAs needs to be addressed as it is critical in the development of NLP tools.

2.2 Large language models

LLMs have exhibited a remarkable transformation throughout the years, where they have evolved from generating only natural texts to understanding them through AI (Jiang et al., 2020). LLMs are trained to predict the next token in a sequence based on the

context, making the generated outputs coherent. They are able to capture long-range dependencies and perform complex tasks such as translation, summarization, and question answering. Moreover, LLMs can generalize across different domains and diverse dialects through prompting techniques (Alabdullah et al., 2025). Research studies vary in terms of whether to include prompts in the analysis or not. For example, Lilli (2023) has studied ChatGPT 4 using Italian dialects; however, the analysis was done using zero-shot analysis only, and the results showed that the model exhibits a significant gap in analytical skills and struggles with text production and interactive tasks, suggesting superior passive linguistic capabilities compared to active ones. Similarly, GPT 4, GPT 3.5, and Bard (Gemini) were compared in terms of Inductive, Mathematical, and Multi-hop Reasoning Tasks using zero-shot, and GPT 4 was found to be better in all of them compared to GPT 3.5 and Bard (Gemini) (López Espejel et al., 2023). Currently, LLMs are widely used in evaluating the performance of NLP tasks in different languages (Kadaoui et al., 2023). However, LLMs are known to have some issues with rare or unseen words, the problem of overfitting, and the difficulty in capturing complex linguistic phenomena.

Researchers have been evaluating different LLM techniques to shed light on future research in the domain (Chang et al., 2023). Other multilingual models such as XGLM (De Varda and Marelli, 2023) have also been studied and were shown to improve significantly in terms of translation performance. It was found that the model performs best if the answer is estimated based on the probability of the first token in the generated answer. However, these models are yet to be studied further (Zhu et al., 2023). Models such as BERT (Devlin et al., 2018) have also been analyzed in terms of language analysis, such as the Arabic language. However, due to its weakness in Arabic dialects, researchers (Baert et al., 2020) created an enhanced language model (BAERT) that showed better performance than BERT in sentiment analysis. LLM research remains a prominent topic across multiple disciplines, including the development and customization of LLMs tailored to specific languages, dialects, or tasks (Mashaabi et al., 2024). There are various LLMs that support the Arabic language, with GPT being the most prominent. Some researchers suggest that ArabianGPT, specifically designed for Arabic, aligns better with Arabic language and rules (Koubaa et al., 2024).

2.3 Machine translation approaches

Machine translation (MT) is an example of an NLP task that addresses grammatical, semantic, and morphological elements between the source and output languages. Importantly, it becomes a challenging task when those elements are significantly different (Joshi et al., 2024). The need for MT systems has been increasing due to the large dialects available on the Internet and their usage in various fields (Sghaier and Zrigui, 2020). Researchers have been studying LLM MT capabilities around the world for different languages. For instance, English to Japanese MT was tested on mBART50, m2m100, Google Translation, Multilingual T5, GPT-3, ChatGPT, and GPT 4 using BLEU, Character Error Rate (CER), WER, Metric for Evaluation of Translation with Explicit ORdering (METEOR), and BERT score, as well as qualitative evaluations by four experts. The analysis showed that GPT 4 outperformed all other models in MT from English to Japanese (Chan and Tang, 2024). Due to their grammatical structure, DA forms a challenge for MT systems (Baniata et al., 2022). MT is an example of an NLP task that addresses grammatical, semantic, and morphological elements between the source and output languages. Importantly, it becomes a challenging task when those elements are significantly different (Joshi et al., 2024). Several approaches and tools are available to perform MT, such as rule-based approaches, hybrid approaches, and sequence-to-sequence (seq2seq) models as well as LLMs (Okpor, 2014). For instance, Salloum and Habash (2012) created a rule-based approach system to translate DA to MSA, which depends on a morphological analyzer, transfer rules, and dictionaries to generate sentences and choose the best matches.

Several researchers have widely used the rule-based approach to translate Arabic dialects to MSA (Al-Gaphari and Al-Yadoumi, 2010; Hamada and Marzouk, 2018; Bouamor et al., 2014). Another study created a hybrid approach to translate the Egyptian dialect to MSA and achieved 90% performance through tokenization (Bakr et al., 2008). Beyond these, Hamed et al. (2025) developed Lahjawi, a customized model specialized in cross-dialectal translation (DA to MSA) that supports 15 dialects. Lahjawi was trained on 7 well-known datasets, including MADAR and Parallel Arabic Dialectal Corpus (PADIC), and fine-tuned above a small language model - Kuwain 1.5B. The model achieved adequate BLEU scores and an accuracy of 58% based on human evaluation. Moreover, Alimi et al. (2024) developed MT model to translate DA to MSA. The model was trained on MADAR and PADIC datasets and fine-tuning transformers such as T5X and AraT5 and some existing tools. The best translation results revealed were for Levantine and Maghrebi region dialects. Some authors also adapted a hybrid approach to translate the Moroccan dialect to MSA using processing tools for MSA (Ridouane and Bouzoubaa, 2014; Hamada and Marzouk, 2018), whereas other studies focused on Neural Machine Translation (NMT) for Arabic dialects (Baniata et al., 2018; Guellil et al., 2017). For example, Baniata et al. (2022) developed an NMT model to translate DA to MSA through multi-head attention with reverse positional encoding and sub-word units. The model achieved high BLEU scores, proving their encoding method across several datasets. In addition, other researchers expand the Dial2MSA dataset through seq2seq datasets in different domains, including social media covering different regions. Leaving a reliable NMT training, the authors conducted a performance evaluation, and it was found that AraT5 achieved the highest performance (Khered et al., 2025). Moreover, researchers Alabdullah et al. (2025) evaluated six LLMs on DA to MSA translation, including Levantine, Egyptian, and Gulf Dialects using different prompting techniques. They demonstrated that GPT 4o achieved the highest score in translation performance, while a fine-tuned version of Gemma2-9B achieved a higher CHrF++ score compared to GPT 4o in zero-show prompting.

Furthermore, researchers utilized LLMs to perform MT tasks. For instance, Zhu et al. (2023) evaluated the multilingual translation of four LLMs, namely, GPT, XGLM, OPT, and BLOOMZ. Interestingly, the researchers found that such models

adapt new patterns to translate. GPT proved excellent capability in MT and outperformed Google Translate according to Peng et al. (2023). In addition, the AraFinNLP shared tasks highlight critical challenges and discussions for cross-dialect translation in preservation of intents using the known ArbBanking77 dataset. The findings highlight that accurate MSA to DA (Moroccan, Tunisian, and Palestinian) translation is possible yet challenging. They demonstrated that fine-tuned BERT models and data augmentation achieve high performance in handling Arabic dialects for financial applications (Malaysha et al., 2024). Moreover, SHAMI-MT developed bidirectional MT models built on the AraT5v2 model and fine-tuned on the Nbra corpus. They evaluated the translation between MSA and the Syrian dialect and used MADAR for benchmark (Sibaee et al., 2025). Similarly, Mohamed et al. (2012) presented a method to convert MSA to Egyptian dialect, applied on part-of-speech (POS). They showed that such MT task improves tagging and is considered as valuable training data for underrepresented dialects.

Prior research studies addressed the translation from MSA to different dialects. A study conducted empirical analysis focusing on Arabic-based LLMs to assess their ability to translate DA to MSA, utilizing four datasets with English-based LLMs as a baseline (Jibrin et al., 2025). They highlighted that AceGPT and Jais performed the best BLEU scores across all data sets, establishing their reliability in Arabic formality. In another study, GPT was evaluated on various NLP tasks. It was revealed that GPT, in comparison with BLOOMZ, struggles on some Arabic tasks yet comparable to human judgment (Khondaker et al., 2023). Several studies explored this field with more precision in relation to the Nuance Arabic Dialect Identification (NADI) 2023 competition. Demidova et al. (2024) performed sentence-based translation from DA to MSA across four dialects through Jais, No Language Left Behind (NLLB), GPT 3.5, and GPT 4 LLMs. They found that Jais outperforms the other models consistently, achieving high BLEU scores whereas NLLB was the least performer. Similarly, other researchers mainly focused on fine-tuning LLama-3 with 8B parameters through Parameter Efficient Fine-Tuning (PEFT) and Low Rank Adaptation (LoRA) methods. The task was also DA-MSA translation across four datasets. LLama fine-tuned model exhibits strong performance related to BLEU metric. Moreover, the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) showed interesting findings through different studies specifically for Dialect to MSA MT task including 5 dialects. Atwany et al. (2024) evaluated AraT5, NLLB, and GPT 3.5. The results show that fine-tuning Arat5 and NLLB on the MADAR dataset demonstrates low BLEU scores, whereas prompting GPT 3.5 achieved high BLEU scores. Moreover, other researchers used GPT 3.5 for dataset generation (Abdelaziz et al., 2024). They used the Saudi Audio Dataset for Arabic (SADA) to translate the audio dialects to MSA texts, leading to notable performance in machine translation achieving high BLEU scores between 25.5 and 31.5. Alahmari et al. (2024) fine-tuned four versions of AraT5 model highlighting that AraT5v2-base-1024 model achieved the highest BLEU score of 21.0. Various researchers have utilized MT with a special focus on the context of Arabic dialects. Table 1 summarizes the MT approaches proposed by the researchers.

3 Proposed methodology

This section discusses the chosen dataset in Section 3.1, followed by Section 3.2, which describes the prompting techniques. Model selection is mentioned in Section 3.3, and the chosen performance metrics are detailed in Section 3.4.

3.1 Dataset

Translating Arabic dialects has been a wide area of research (Harrat et al., 2019). In our research, we aim to use the QADI dataset and the MADAR corpus dataset. QADI dataset is a pre-processed dataset collected through X media platform, and it includes 18 dialects from different Arab countries, the dataset is already cleaned and has no hashtags, emojis, or such symbols which might affect the translation quality (Abdelali et al., 2020). The dataset has 540k training tweets and 3,303 test tweets in total. The rationale for choosing the QADI dataset is the large number of dialects it has which will help us address our research questions and compare the performance evaluation of LLMs. However, in the current study, 50K samples will be used from all countries for the analysis due to computational resource restrictions. We applied random sampling, the QADI dataset was balanced across dialects, our random selection ensured that the selected 50K tweets have no bias and ensure equal selection among the sentences. Table 2 shows different country codes using ISO-3166-1 with corresponding users and tweet count of QADI dataset.

Similarly, the MADAR corpus dataset (Bouamor et al., 2019) contains 25 cities representing 15 countries, each with a unique dialect where some countries feature multiple cities (e.g., Egypt has Aswan, Cairo, and Alexandria) with 2K samples from each dialect. The advantage of using the MADAR dataset is that it includes MSA baseline translation for the sentences present inside the dialects of each country. Hence, making the evaluation of GPT and Bard (Gemini) stronger by comparing the results of these models with the baseline given within the dataset. This study will analyze 15 countries from the MADAR dataset primarily focusing on the capitals of countries that are also included in QADI. Table 3 shows all the city dialects from the MADAR dataset, showing the different cities with their dialects from various Arabic countries.

3.2 Prompting techniques

Prompting strategies have been developed to optimize LLMs' performance and outcomes. The most frequent of these tactics are zero-shot and few-shot. The zero-shot prompt plainly describes the task and provides information without examples (Allingham et al., 2023). Figures 1, 2 show an example of the prompts used to perform the translation task. Unlike zero-shot prompts, few-shot prompts include data examples and sample responses (Jiang et al., 2022). On the other hand, a few-shot prompting technique is established by providing an example within the prompt itself, where one-shot includes a single example, two-shot includes 2 examples, etc. We will include both zero-shot and few-shot prompts. As well as a few

TABLE 1 Summary of machine translation (MT) approaches for Arabic dialects.

Research	Dialect(s)	Approach
Bakr et al., 2008	Egyptian → MSA	Hybrid
Al-Gaphari and Al-Yadoumi, 2010	Sana'ani → MSA	Rule-based
Salloum and Habash, 2012	Arabic Dialects → MSA	Rule-based
Mohamed et al., 2012	MSA → Egyptian	Rule-based
Bouamor et al., 2014	Mainly Egyptian	Rule-based, Corpus of 2,000 sentences
Ridouane and Bouzoubaa, 2014	Moroccan → MSA	Hybrid
Guellil et al., 2017	Algerian	NMT
Hamada and Marzouk, 2018	Egyptian → MSA	Hybrid/Rule-based
Baniata et al., 2018	Arabic dialects → MSA	Neural MT (NMT)
Hamed et al., 2025	15 Dialects → MSA	Custom cross-dialectal model
Alimi et al., 2024	Levantine, Maghrebi → MSA	Transformer-based MT (AraT5, T5X)
Alabdullah et al., 2025	Levantine, Egyptian, Gulf → MSA	LLM-based MT (GPT 4o, Gemma2-9B)
Zhu et al., 2023	Multilingual/Arabic	LLM-based MT (GPT, XGLM, OPT, BLOOMZ)
Malaysha et al., 2024	Moroccan, Tunisian, Palestinian → MSA	LLM + fine-tuned BERT
Sibaee et al., 2025	Syrian → MSA	AraT5v2-based bidirectional MT
Khered et al., 2025	Arabic Dialects → MSA	Seq2seq / Transformer (AraT5)
Jibrin et al., 2025	Arabic Dialects → MSA	LLM-based MT (AceGPT, Jais)
Khondaker et al., 2023	Arabic Dialects → MSA	LLM-based MT (GPT, BLOOMZ)
Demidova et al., 2024	Egyptian, Emirati, Jordanian, and Palestinian → MSA	LLM-based MT (Jais, NLLB, GPT 3.5, GPT 4)
Atwany et al., 2024	Gulf, Egyptian, Levantine, Iraqi and Maghrebi → MSA	LLM-based MT (AraT5, NLLB, GPT 3.5)
Abdelaziz et al., 2024	Saudi Dialect → MSA	LLM-based MT (GPT 3.5)
Alahmari et al., 2024	Arabic dialects → MSA	Transformer MT (AraT5v2)

shot prompts (one-shot) for the country with the weakest dialect translation given by the models to check whether including an example within the prompt would enhance the overall accuracy of the translation. An example of a prompt is shown in Figure 3 to test whether the models would provide a better translation as compared to zero-shot approaches.

3.3 Model selection

This research paper will be using OpenAI's most recent model GPT 5 along with GPT 3.5, GPT 4, and Google's Bard (Gemini) "text-bison" model due to their exceptional performance in research (Zhu et al., 2023; Peng et al., 2023; Khondaker et al., 2023; Kadaoui et al., 2023). LLMs are widely used to evaluate the performance of Arabic NLP tasks such as GPT 3.5, GPT 4, Bard (Gemini), XGLM, and OPT (Zhu et al., 2023). To save computational cost and time, GPT 5 will only be ran on MADAR dataset, whereas QADI will include all remaining models. This study's selection criteria for the models aim to balance between budget and computing resources. In addition, LLM languages that do not include the Arabic language, such as Falcon-7b (Penedo et al., 2023), were initially excluded from the search scope of suitable LLMs. A brief summarization of both models is shown in Table 4.

Figure 4 shows the experiment pipeline implemented for GPT and Bard (Gemini). The experiment starts using the data in the dataset as a prompt for each LLM. Initially, all prompts will be applied with zero-shot techniques, meaning that no example will be included within the prompt. However, after performing the analysis, the country with the least translation performance will be analyzed again but with the few-shot prompting technique. In the QADI dataset, to have a baseline to compare the LLM results with, the back translation process is used (Behr, 2017), where dialects are translated to MSA; then, the resulting MSA is translated back to the corresponding dialect to compare the final resulting dialect with the original dialect from the dataset. However, MADAR offers a baseline for dialects and MSA; therefore, no back-translation will be needed.

For LLM inference, we used the code provided on the Application Programming Interface (API) websites with some correction techniques; rerunning the prompt if the model returns an error to ensure a correct response. After doing so, the error rate in the resulting samples has dropped sufficiently. Cost optimization technique has also been adapted by running 10 translations per API request, which reduced the cost. A threshold of 10 requests was set as the maximum accumulation; as the threshold increases, the error

TABLE 2 QADI dataset: users and tweet counts by country using ISO-3166-1 codes.

Country	Users	Training tweets (k)	Test tweets
Iraq (IQ)	142	18.4	178
Bahrain (BH)	169	28.3	184
Kuwait (KW)	160	49.9	190
Saudi Arabia (SA)	149	35.4	199
United Arab Emirates (AE)	172	27.8	192
Oman (OM)	176	24.8	169
Qatar (QA)	139	36.7	198
Yemen (YE)	138	11.6	193
Syria (SY)	139	18.3	194
Jordan (JO)	146	34.1	180
Palestine (PL)	145	48.6	173
Lebanon (LB)	141	38.4	194
Egypt (EG)	150	67.8	200
Sudan (SD)	139	16.3	188
Libya (LY)	149	40.9	169
Tunisia (TN)	68	12.9	154
Algeria (DZ)	130	17.6	170
Morocco (MA)	73	12.8	178

TABLE 3 All the city dialects and regions that were included in the building of the MADAR dataset.

Region	Sub-region	Cities	Codes
Maghreb	Morocco	Rabat, Fes	RAB, FES
	Algeria	Algiers	ALG
	Tunisia	Tunis, Sfax	TUN, SFX
	Libya	Tripoli, Benghazi	TRI, BEN
Nile Basin	Egypt	Cairo, Alexandria, Aswan	CAI, ALX, ASW
	Sudan	Khartoum	KHA
Levant	South Levant	Jerusalem, Amman, Salt	JER, AMM, SAL
	North Levant	Beirut, Damascus, Aleppo	BEI, DAM, ALE
Gulf	Iraq	Mosul, Baghdad, Basra	MOS, BAG, BAS
	Gulf	Doha, Muscat, Riyadh, Jeddah	DOH, MUS, RIY, JED
Yemen	Yemen	Sana'a	SAN

rate also increases. Finally, the experiment results will be evaluated by calculating the selected performance metrics described in the upcoming section.

The following array has multiple JSON objects, each object has 3 keys, DS which is the Dialectal Arabic sentence, MSA is the translation of DS to Modern Standard Arabic and BT is the translation of MSA to Yemeni Arabic. Complete Both MSA and BT in the array: ${JSON.stringify({ tmpArray })}

FIGURE 1
Zero-shot prompt - QADI.

The following array has multiple JSON objects, each containing two keys: MSA, representing the Modern Standard Arabic sentence, and MSAtoD, which is the translation of MSA to Tunisian Arabic. Complete the MSAtoD field in the array with the appropriate translations. Here's the array: ${JSON.stringify({ tmpArray })}

FIGURE 2
Zero-shot prompt - MADAR.

3.4 Performance metrics

We aim to quantify the differences in performance between GPT 3.5, GPT 4, GPT 5, and Bard (Gemini) and to determine how these models can perform the translation task given the complexity of the Arabic language. There are various common evaluation metrics for comparison. The present study will use 7 evaluation metrics (i.e., cosine similarity, sentence BERT, semantic universal encoder, TER, BLEU, ROUGE, and ANOVA test). These metrics were chosen based on their strengths and popularity in analyzing Arabic sentences. To attest for normality, the Shapiro–Wilk test was used for ANOVA (Alabdullah et al., 2025).

One of the common MT metrics is the universal similarity encoder, which is a neural network architecture for learning similarity-preserving embeddings that uses pre-trained embeddings (e.g., Word2Vec, GloVe, or BERT embeddings) to compare two sentences, rather than having a specific calculation formula. Its range varies from −1 to 1, where results closer to 1 are indicative of high semantic similarity.

However, cosine similarity calculates the cosine of the angle formed by two vectors that represent phrases in several dimensions that represent a word or contextual information. Equation 1 below shows the cosine similarity, where A and B are vectors.

$$\text{Cosine similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{1}$$

High positive values in cosine similarity (close to 1) indicate that there is great similarity between the two vectors.

Sentence BERT is a transformer that adapts cosine similarity by using Tensorflow. The general process involves encoding sentences into fixed-size vectors using pre-trained BERT embedding and then calculating a similarity score between these vectors (Mrinalini et al., 2022). Since sentence BERT adapts cosine similarity, it follows the

The following array has multiple JSON objects, each containing two keys: MSA, representing the Modern Standard Arabic sentence, and MSAtoD, which is the translation of MSA to Tunisian Arabic. Your task is to complete the MSAtoD field in the array with appropriate translations. Below are examples of MSA sentences and their translations into Tunisian Arabic:

1. MSA: "مرحبا، كيف حالك؟"

   MSAtoD: "مرحبا، شحالك؟"

2. MSA: "كم الساعة الآن؟"

   MSAtoD: "شحال فالعداد؟"

3. MSA: "هل يمكنك مساعدتي من فضلك؟"

   MSAtoD: "تقدر تعاونني من فضلك؟"

Complete the MSAtoD field in the array with the appropriate translations. Here's the array:

${JSON.stringify({ tmpArray })}

FIGURE 3
Few-shot prompt - MADAR.

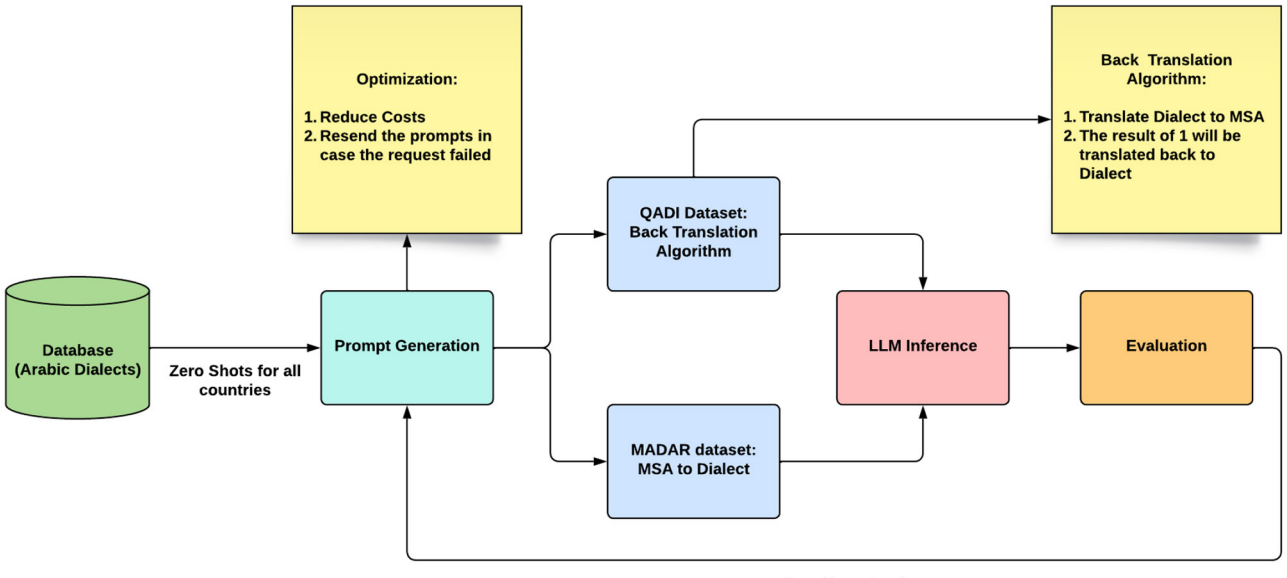

FIGURE 4
Experiment pipeline.

same metric measures of −1 to 1, where close values to −1 mean that the two vectors are completely dissimilar, and values close to 1 mean that there is a high similarity between the vectors. The universal sentence encoder finds the similarity between sentences based on semantics, where it is used to convert phrases into dense vector representations.

Finally, the TER metric is specifically used for MT tasks by comparing the MT outputs against human-generated translation to assess the quality of MT outputs, as shown in Equation 2.

$$\text{TER} = \frac{\text{Total edits}}{\text{Total words in reference translation}} \tag{2}$$

A lower TER score indicates a better translation quality as it means that fewer edits are needed to align the machine-generated translation.

TABLE 4 Tabular comparison between GPT and Bard.

Aspect	GPT 3.5	GPT 4	GPT 5	Bard
Source	OpenAI	OpenAI	OpenAI	Google
Language model	`GPT 3.5-turbo-16k`	`'GPT 4-0125-preview'`	`'GPT 5'`	'text-bison'
Model architecture	Transformer decoder based	Transformer decoder based	Transformer decoder based	Transformer based
Availability	Limited free access	Paid	Paid	Limited free access
Languages	Multilingual	Multilingual	Multilingual	Multilingual
Parameter Size	175 Billion	1.76 Trillion	Not Announced	137 Billion

Moreover, the BLEU metric is a widely popular metric used in research (Sallam and Mousa, 2024) where individual translated segments, usually sentences, are scored by comparing them with a collection of high-quality reference translations. These scores are then averaged throughout the entire corpus to provide an approximation of the translation's overall quality (Papineni et al., 2002). It aims to find the similarity between the translated text and the reference sentence by employing n-grams; contiguous group of n-words that are similar. The metric values range from 0 to 1, and typically a higher value means that more words are overlapping between the machine-translated sentence and the referenced sentence, as shown in Equation 3 (Papineni et al., 2002).

$$\mathrm{BLEU}_w(\hat{S}; S) := \mathrm{BP}(\hat{S}; S) \cdot \exp\left(\sum_{n=1}^{\infty} w_n \log p_n(\hat{S}; S)\right) \quad (3)$$

where BP is the brevity penalty, w is the weights for each n-gram, and p is the precision of n-grams.

Furthermore, ROUGE is a collection of metrics and software packages for assessing automatic summarization and MT software in natural language processing. The metrics assess an automatically generated summary or translation to a reference or a collection of references (human-created summary or translation). ROUGE measures range from 0 to 1, with higher scores indicating a stronger resemblance between the automatically generated summary and the reference (Lin and Hovy, 2003).

ANOVA is a statistical approach for comparing the means of three or more samples to determine whether one of them is substantially different from the others (Keselman et al., 1998). It accomplishes this by analyzing the variance in the data and categorizing it as the variance between groups and the variance within groups. The *p*-value is calculated using the ANOVA test statistic, also known as the F-statistic, as shown in Equation 4.

$$\text{F-statistic (ANOVA Coefficient)} = \frac{\text{Mean Sum of Squares due to Treatment (MST)}}{\text{Mean Sum of Squares due to Error (MSE)}} \quad (4)$$

The *p*-value indicates whether the differences in group means are statistically significant (Keselman et al., 1998). In this study, since we are performing various analyses and tests, it became important to employ ANOVA to determine the statistical significance of the results.

4 Experimental results

This section discusses the model responsiveness in Section 4.1, followed by the metric performance and dialect variations in Section 4.2. Finally, Section 4.3 discusses the impact of sentence length on the model accuracy.

4.1 Model responsiveness

In general, in terms of responsiveness, the models were responsive when given a prompt with input. However, there were differences in the output details of both models. GPT gave a direct response where Gemini explained each word in a row.

When running APIs, Bard (Gemini) has shown varying error rates when translating ranging from 5% up to 71%. This error rate was varying based on the load on the network at the execution time and length of the dataset being analyzed. Hence, to reduce the error rate, we ran Bard (Gemini) when the network was not preoccupied with many other tasks and ran the dataset in smaller batches to reduce the chances of error. There were several cases where Bard (Gemini) has either returned the same input as output, empty output, or a message that says that it is unable to handle a given task.

The rate of failing to give an output is most noticeable when performing the back translation from MSA to a certain dialect in QADI dataset. For example, for the back translation for IQ dialect, Bard (Gemini) failed to give an output with the rate of 37.5%, whereas GPT 3.5 has only failed to do so with a 5.6% rate, and GPT 4 had 0.2% error rate. Therefore, a correction technique was added in the code, where the response was checked, if it included an error, resend the same prompt. After doing so, the error rate in the resulting samples has dropped considerably.

4.2 Performance metrics and dialect variations

4.2.1 Similarity metrics

This section discusses the similarity metrics and the performance of the LLMs on the MADAR and QADI datasets in terms of universal similarity encoder, cosine similarity, sentence BERT, BLEU, and ROUGE F1 scores. The metrics aimed to assess the efficiency and accuracy of the translation process of different dialects. The analysis explained below is further demonstrated in Tables 5 – 11. To address the research questions, both GPT 3.5/4

10.3389/frai.2025.1661789

TABLE 5 Bard metric similarities mean among 18 dialects from QADI dataset.

Dialect	Univ. Sim. Enc.	Cosine Sim.	Sent. BERT	BLEU	ROUGE-L
JO	0.68	0.43	0.92	0.07	0.43
AE	0.65	0.38	0.92	0.35	0.38
LB	0.67	0.40	0.87	0.38	0.40
IQ	0.64	0.40	0.91	0.39	0.41
BH	0.67	0.46	0.88	0.07	0.46
DZ	0.64	0.41	0.89	0.39	0.41
EG	0.72	0.47	0.89	0.45	0.47
KW	0.67	0.46	0.94	0.43	0.45
LY	0.70	0.48	0.90	0.45	0.47
MA	0.63	0.38	0.94	0.04	0.38
OM	0.64	0.45	0.94	0.43	0.45
PL	0.64	0.42	0.94	0.40	0.42
QA	0.67	0.42	0.94	0.05	0.42
SA	0.65	0.39	0.93	0.37	0.39
SD	0.68	0.44	0.90	0.06	0.43
SY	0.66	0.46	0.90	0.43	0.45
TN	0.65	0.42	0.89	0.39	0.41
YE	0.68	0.47	0.93	0.44	0.47

TABLE 6 Bard metric similarities mean among 15 dialects from MADAR dataset.

Dialect	Univ. Sim. Enc.	Cosine Sim.	Sent. BERT	BLEU	ROUGE-L
JO	0.56	0.34	0.93	0.37	0.32
LB	0.53	0.35	0.93	0.34	0.28
IQ	0.50	0.33	0.93	0.32	0.26
DZ	0.52	0.31	0.93	0.29	0.23
EG	0.57	0.38	0.93	0.37	0.32
LY	0.53	0.32	0.93	0.31	0.25
MA	0.50	0.31	0.93	0.29	0.23
OM	0.58	0.40	0.93	0.38	0.33
PL	0.56	0.39	0.92	0.37	0.32
QA	0.53	0.36	0.93	0.34	0.28
SA	0.53	0.35	0.93	0.33	0.27
SD	0.56	0.38	0.94	0.37	0.32
SY	0.55	0.39	0.93	0.37	0.32
TN	0.48	0.26	0.93	0.25	0.17
YE	0.50	0.28	0.93	0.27	0.20

TABLE 7 GPT 3.5 metric similarities mean among 18 dialects from QADI dataset.

Dialect	Univ. Sim. Enc.	Cosine Sim.	Sent. BERT	BLEU	ROUGE-L
JO	0.66	0.38	0.89	0.43	0.46
AE	0.66	0.37	0.88	0.39	0.43
LB	0.65	0.40	0.94	0.48	0.50
IQ	0.62	0.33	0.84	0.38	0.40
BH	0.67	0.40	0.87	0.44	0.47
DZ	0.59	0.29	0.91	0.28	0.31
EG	0.65	0.35	0.86	0.32	0.35
KW	0.65	0.39	0.90	0.45	0.48
LY	0.63	0.34	0.85	0.32	0.36
MA	0.64	0.34	0.89	0.37	0.40
OM	0.64	0.39	0.84	0.46	0.49
PL	0.67	0.43	0.84	0.53	0.55
QA	0.63	0.35	0.87	0.25	0.40
SA	0.63	0.33	0.89	0.32	0.36
SD	0.65	0.37	0.85	0.35	0.46
SY	0.65	0.39	0.90	0.43	0.46
TN	0.66	0.41	0.83	0.46	0.49
YE	0.63	0.39	0.85	0.43	0.45

TABLE 8 GPT 3.5 metric similarities mean among 15 dialects from MADAR dataset.

Dialect	Univ. Sim. Enc.	Cosine Sim.	Sent. BERT	BLEU	ROUGE-L
JO	0.55	0.35	0.92	0.34	0.30
LB	0.52	0.32	0.91	0.32	0.25
IQ	0.51	0.29	0.93	0.28	0.22
DZ	0.50	0.28	0.93	0.26	0.20
EG	0.54	0.34	0.93	0.33	0.28
LY	0.51	0.27	0.93	0.27	0.20
MA	0.50	0.27	0.93	0.26	0.20
OM	0.53	0.31	0.92	0.29	0.24
PL	0.54	0.34	0.92	0.33	0.28
QA	0.53	0.31	0.93	0.30	0.24
SA	0.55	0.34	0.93	0.34	0.28
SD	0.53	0.31	0.92	0.29	0.24
SY	0.55	0.36	0.92	0.35	0.30
TN	0.48	0.24	0.93	0.23	0.16
YE	0.50	0.26	0.93	0.25	0.19

TABLE 9 GPT 4 metric similarities mean among 18 dialects from QADI dataset.

Dialect	Univ. Sim. Enc.	Cosine Sim.	Sent. BERT	BLEU	ROUGE-L
JO	0.73	0.50	0.82	0.49	0.51
AE	0.71	0.45	0.91	0.44	0.46
LB	0.74	0.50	0.94	0.49	0.51
IQ	0.70	0.43	0.88	0.43	0.45
BH	0.72	0.48	0.91	0.48	0.49
DZ	0.75	0.53	0.91	0.55	0.57
EG	0.77	0.55	0.90	0.55	0.57
KW	0.68	0.45	0.88	0.45	0.47
LY	0.70	0.43	0.87	0.42	0.44
MA	0.70	0.41	0.89	0.40	0.41
OM	0.65	0.39	0.77	0.38	0.39
PL	0.71	0.49	0.88	0.48	0.50
QA	0.66	0.37	0.87	0.36	0.37
SA	0.69	0.38	0.89	0.36	0.38
SD	0.74	0.50	0.93	0.51	0.53
SY	0.72	0.48	0.92	0.46	0.49
TN	0.71	0.44	0.88	0.44	0.45
YE	0.69	0.43	0.91	0.41	0.43

TABLE 10 GPT 4 metric similarities mean among 15 dialects from MADAR dataset.

Dialect	Univ. Sim. Enc.	Cosine Sim.	Sent. BERT	BLEU	ROUGE-L
JO	0.60	0.42	0.93	0.41	0.37
LB	0.54	0.34	0.43	0.36	0.28
IQ	0.54	0.34	0.93	0.33	0.27
DZ	0.51	0.30	0.93	0.29	0.23
EG	0.56	0.38	0.93	0.38	0.33
LY	0.52	0.31	0.93	0.30	0.24
MA	0.47	0.26	0.93	0.25	0.18
OM	0.53	0.33	0.93	0.32	0.26
PL	0.59	0.41	0.92	0.41	0.36
QA	0.57	0.39	0.93	0.38	0.33
SA	0.58	0.41	0.93	0.40	0.35
SD	0.54	0.33	0.93	0.32	0.26
SY	0.59	0.41	0.92	0.41	0.36
TN	0.48	0.26	0.93	0.25	0.18
YE	0.52	0.30	0.92	0.29	0.22

and Bard (Gemini) exhibited similar performance levels across the metrics among dialects in both datasets.

The BLEU score values for GPT 3.5/4 are similar among the LLMs and countries for QADI, whereas GPT 5 slightly outperformed its prior models in MADAR dataset. Figures 5, 6 visualize the BLEU scores labeled by each country where the LLMs showed consistent results in MADAR. Bard (Gemini) in the QADI dataset achieved a low score for some countries. These numbers explain that a few words were overlapping between the input and the translated dialect.

TABLE 11 GPT 5 metric similarities mean among 15 dialects from MADAR dataset.

Dialect	Univ. Sim. Enc.	Cosine Sim.	Sent. BERT	BLEU	ROUGE-L
JO	0.62	0.46	0.93	0.47	0.43
LB	0.58	0.39	0.92	0.39	0.34
IQ	0.55	0.37	0.92	0.37	0.31
DZ	0.50	0.28	0.93	0.26	0.20
EG	0.59	0.44	0.92	0.44	0.40
LY	0.54	0.37	0.92	0.36	0.30
MA	0.56	0.40	0.92	0.39	0.34
OM	0.52	0.34	0.93	0.37	0.28
PL	0.61	0.46	0.92	0.47	0.42
QA	0.59	0.43	0.92	0.44	0.38
SA	0.58	0.42	0.92	0.43	0.38
SD	0.54	0.38	0.92	0.37	0.32
SY	0.62	0.47	0.92	0.49	0.44
TN	0.53	0.34	0.92	0.33	0.27
YE	0.55	0.35	0.93	0.34	0.28

Furthermore, when employing a universal similarity encoder and cosine similarity in QADI as shown in Table 12, GPT 4 outperforms the models, which makes it the dominant, followed by Bard (Gemini) and then GPT 3.5. The mean universal similarity encoder score is 71% for GPT 4, 64% for GPT 3.5, and 66% for Bard (Gemini) among all countries. For the MADAR dataset in Table 13, GPT 5 outperforms all models by having a 57% average, whereas GPT 4 has a mean of 54%, GPT 3.5 mean is 52%, whereas Bard (Gemini) has a mean of 53%. This suggests that Bard (Gemini) has shown comparable skill to older GPT models in understanding and conveying the semantic connections among the translated sentences in the MADAR dataset, whereas GPT 5 stands out overall. Whereas for the QADI dataset, GPT 4 had a higher mean, which indicates that it has the best skill in conveying the semantic connections with the existence of the back translation algorithm.

In Table 12 for QADI, the cosine similarity showed a mean of 46% for GPT 4, 43% for Bard (Gemini), and 37% for GPT 3.5. Table 13 exhibits a similar performance of 35% for GPT 4, 39% for Bard (Gemini), and 31% for GPT 3.5 on MADAR. This shows that GPT 4 is the best performer which aligns with the results of Alyafeai et al. (2023) and Peng et al. (2023). GPT 5 outperforms other models with a mean of 39% in MADAR. Noticeably, GPT 3.5 encountered the most struggles in translating to dialects from MSA which exhibits to a similar behavior in the conclusion drawn by Kadaoui et al. (2023).

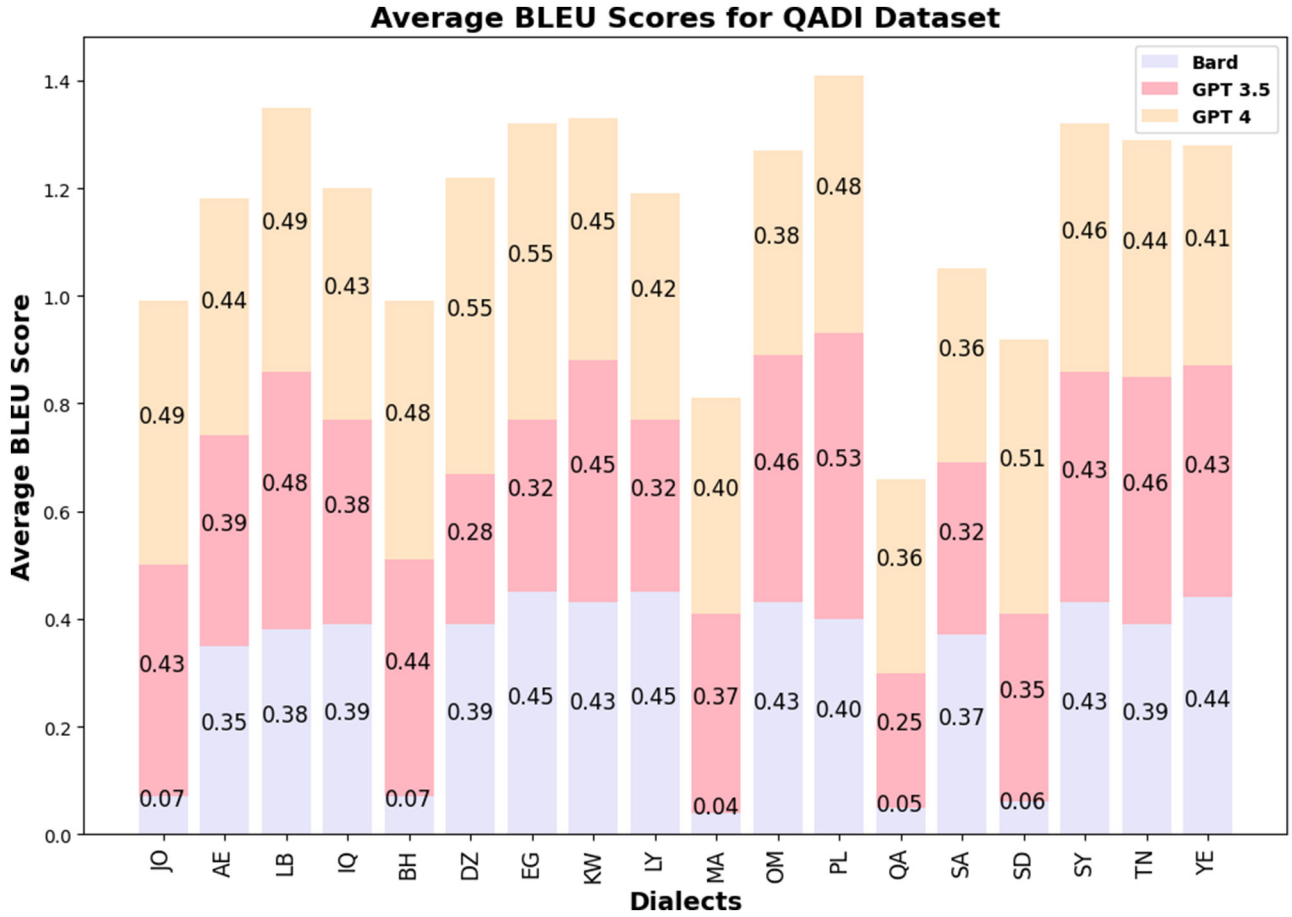

FIGURE 5
Average BLEU scores QADI.

On the other hand, sentence BERT shows the highest mean among all metrics as it uses a transformer model which makes it most accurate in finding similarities between the input dialect and the back-translated dialect. In addition, it showed consistent results for all LLMs across the two datasets. In Table 12 for QADI, Bard (Gemini) shows an average efficiency of 91%, hence outperforming GPT 4 and GPT 3.5 which shows an average efficiency of 89% and 87% consecutively. Similarly for MADAR in Table 13, Bard (Gemini) shows a total mean value of 93%, tying with GPT 3.5 whereas GPT 5 shows 92%, GPT 4 shows 90%. GPT 4 has witnessed a drop in accuracy due to poorer performance in LB dialect because of an outlier compared to other countries as its individual score shows 43% score, whereas others scored approximately 93%. This is due to an error occurred when running the data where sentences were translated to English instead of Arabic which drops the accuracy rate of the overall translation. Given that the error was only observed in the Lebanese dialect, it could be attributed that the model had unresolved difficulties in the background which was also passed down to the updated GPT 5 model as well.

In QADI dataset in Table 12, GPT 3.5 and Bard (Gemini) have an average score of 43% for ROUGE-L where GPT 4 scored an average of 47%. The analysis note that at least one Maghrebi dialect was of the highest ROUGE-L values observed for all models. However, GPT 3.5 achieved the top score for Palestine. This indicates a greater number of sentences overlap. These results indicate that GPT 4 was specifically well trained and consistent in at least one Magherbi dialect (e.g., Moroccon, Algerian, or Tunisian Arabic), whereas GPT 3.5 was a better fit in Palestinian dialect (i.e., Levantine Arabic).

In the same vein for the MADAR in Table 13, ROUGE-L scores were similar showing an average of 27%, 24%, 28% for Bard (Gemini), GPT 3.5/4, respectively, whereas GPT 5 outperforms other models showing 34%. Figures 7, 8 show the averages for each model to further illustrate the scores.

Overall, all three models among different datasets demonstrated a decently high average score for ROUGE-1 and ROUGE-L but lower scores for ROUGE-2. These results indicate that GPT 3.5, GPT 4, and Bard (Gemini) all had higher overlap between single words and long sequences between the compared text with GPT 4 being the highest in Figure 7, whereas GPT 5 clearly outperforms all other models in MADAR as demonstrated in Figure 8.

Overall, the results show that GPT 5 followed by GPT 4, Bard (Gemini), and GPT 3.5 are efficient in translating MSA to different

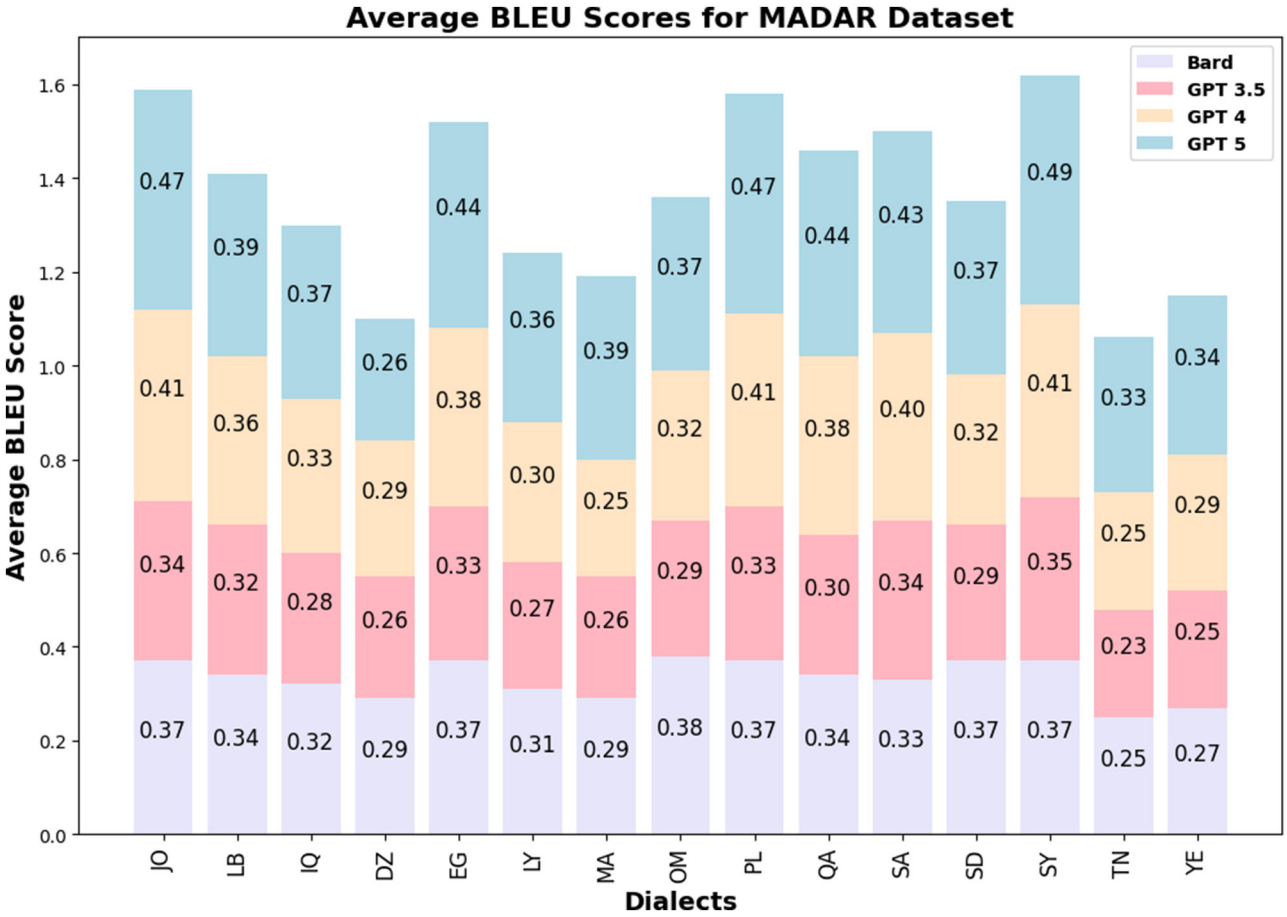

FIGURE 6
Average BLEU scores MADAR.

TABLE 12 Average similarity metrics for QADI dataset.

Metric	GPT 3.5	GPT 4	Bard (Gemini)
Universal similarity encoder	0.64	0.71	0.66
Cosine similarity	0.37	0.46	0.43
Sentence BERT	0.87	0.89	0.91
BLEU	0.39	0.45	0.31
ROUGE-L	0.43	0.47	0.43
TER	15.62%	15.75%	16.55%

Lower error rates are denoted by green.

TABLE 13 Average similarity metrics for MADAR dataset.

Metric	GPT 3.5	GPT 4	GPT 5	Bard (Gemini)
Universal similarity encoder	0.52	0.54	0.57	0.53
Cosine similarity	0.31	0.35	0.39	0.34
Sentence BERT	0.93	0.90	0.92	0.93
BLEU	0.30	0.34	0.39	0.33
ROUGE-L	0.24	0.28	0.34	0.27
TER	6.76%	6.74%	6.61%	6.90%

Lower error rates are denoted by green.

DA, with slight difference and weaknesses noted in some of the dialects and models.

4.2.2 TER

Table 14 shows the TER for all the countries for QADI dataset for GPT 3.5, GPT 4, and MADAR, whereas the Figures 9, 10 visualize some dialects' results from QADI representing the average TER as a red line. The ranges of error demonstrated by TER range from approximately 10% up to 25% for all LLMs. Furthermore, the models have the lowest TER rate of approximately 11% for the OM dialect, whereas Bard (Gemini) has the highest worst TER rate in EG of 25.6%. Comparing the Gulf region countries (AE, BH, KW, OM, QA, and SA) specifically on GPT 3.5, OM showed the lowest TER of approximately 10%, whereas the other countries from the region showed an average ranging from 14% to 18%.

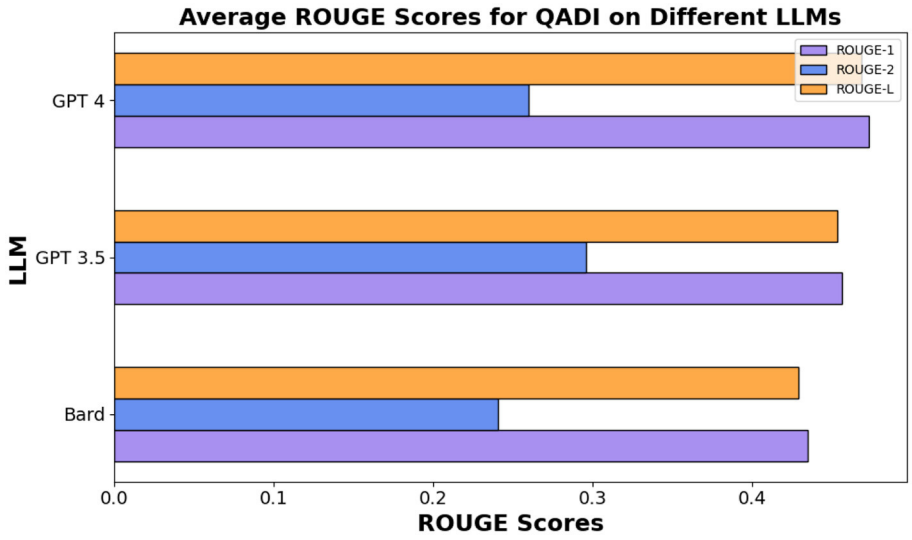

FIGURE 7
Average ROUGE scores for QADI dataset.

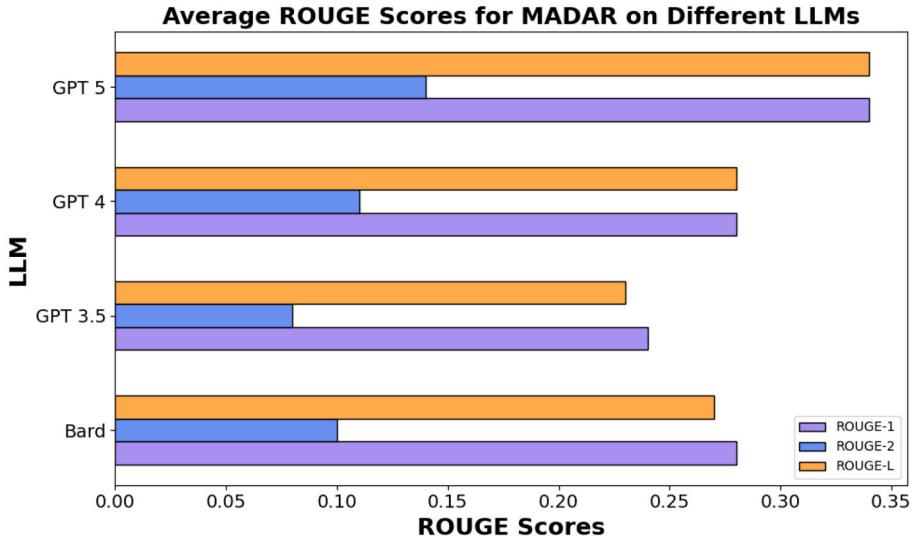

FIGURE 8
Average ROUGE scores for MADAR dataset.

On the other hand, Table 15 and Figure 11 specifically showing GPT 4 illustrate the TER values of each country employing MADAR dataset as an example. In comparison with QADI dataset, the TER rates are closer together and have an overall lower value ranging from 6% to 7%, with JO being the highest and QA, SY, and OM being the lowest in the MADAR and QADI datasets. This may be explained by the fact that the MADAR dataset gathers sentences from a single source as a CORPUS, unlike the QADI dataset, which gathers sentences from X platform (Twitter) which is more prone to errors due to difficulty in filtering the sentences as tweets.

Overall, in terms of efficiency and consistency combined, all models show competitive results and proved capable of translating multiple dialects regardless of the region as they all had approximately close values across the Middle East such as PL, LB, SY, and JO, the Gulf region such as KW, AE, SA, BH, OM, and QA, the Arab Maghreb region such as MA, LY, DZ, and TN and the African and Asian countries such as EG, SD, YE, and IQ. In QADI, GPT 4 outperforms the other LLMs in all similarity metrics and TER, Bard (Gemini) comes in the second place and then GPT 3.5 as shown in Table 12 whereas GPT 5 outperforms GPT 4 and other models in MADAR in Table 13 proving it being a more reliable model in translating from MSA to DA. This is further demonstrated in Figures 12, 13 which further demonstrate LLM performance upon the metrics used in this study. Models exhibited consistent scores among all metrics with GPT 5 being the highest and most appropriate LLM to deal with Arabic dialects.

TABLE 14 TER for comparison for Bard, GPT 3.5, and GPT 4 for each dialect in the QADI dataset, where lower TER means higher performance.

Dialect	Bard	GPT 3.5	GPT 4
JO	18.08%	17.51%	18.02%
AE	17.02%	16.94%	17.75%
LB	18.16%	16.56%	17.34%
IQ	15.17%	15.06%	15.86%
BH	15.87%	14.97%	13.70%
DZ	16.64%	14.90%	13.37%
EG	25.60%	21.54%	22.91%
KW	14.81%	13.52%	12.47%
LY	18.65%	17.53%	17.66%
MA	14.80%	15.14%	17.23%
OM	11.43%	11.02%	10.82%
PL	11.82%	11.62%	11.38%
QA	17.98%	16.14%	14.83%
SA	15.89%	15.93%	16.75%
SD	19.10%	17.85%	16.89%
SY	14.59%	14.38%	14.42%
TN	16.28%	15.62%	16.69%
YE	16.04%	14.92%	15.35%

High error rates are colored by red, lower rates are denoted by green.

4.2.3 ANOVA

ANOVA test is a common test used to check whether the data and mean difference are significant based on different conditions and factors. In previous sections, we found that the average translation performance among similarity metrics and TER are quite similar. To better understand the significance difference, one-way ANOVA is applied to all countries and models with alpha 0.05 threshold. We have applied Shapiro–Wilk test diagnostic to verify the residuals normality and applicable for ANOVA. This is a similar approach adapted by Alabdullah et al. (2025). The ANOVA results are shown in Table 16 for QADI and Table 17 for the MADAR dataset. The models GPT and Gemini are the independent variables and the performance metrics including similarity metrics, BLEU, and ROUGE were considered dependent variables. In reference to Table 16, ANOVA test is applied among all similarity metrics, and there is a significant difference between the model translation performance with a p-value close to 0 in universal similarity encoder, cosine similarity, and sentence BERT, which indicates that the probability of the average similarities are different is approximately 99.96%. Metrics such as BLEU, ROUGE-L, and TER show insignificant difference among the models meaning that all models have similar scores/error rates in translation. Moreover, the f-value <1 suggested that there is no variance across the means.

As for MADAR, Table 17 shows that there is no difference between the means and all models exhibited similar translation performance on sentence BERT, ROUGE-L, and TER. However, the other metrics show significant differences between the LLMs' scores.

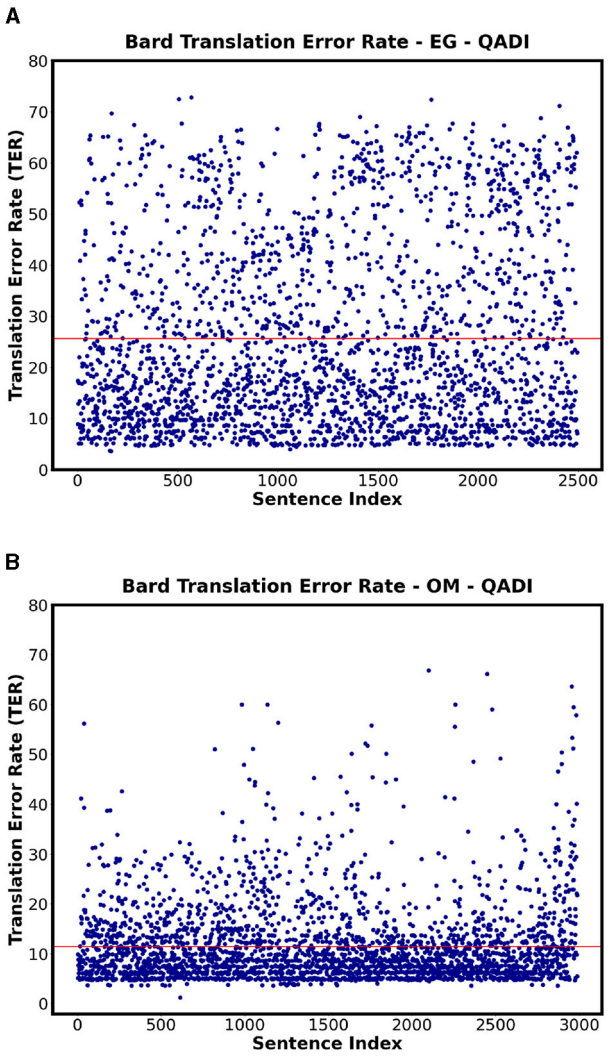

FIGURE 9
Scatter plots showing the TER for QADI datasets on Bard for highest and lowest countries. **(A)** Bard - EG Highest TER. **(B)** Bard - OM Lowest TER.

4.2.4 Evaluation divergence (lexical vs. semantic metrics)

Upon evaluating different models with different performance metrics, some conflicts between the metrics were noted. To strengthen our analysis, we have chosen different metrics, each evaluating a certain category of the LLMs ability. BLEU and ROUGE rely on lexical overlap with the reference translation (the original dialect in our case) and count the n-gram overlap. On the other hand, universal similarity encoder and sentence-BERT are semantic measures that focus on meaning equivalence regardless of literal word matching. TER is concerned with the number of edits to match the generated dialect with the base dialect reference. As we are evaluating the 15 dialects, this variation often involves synonym choice, morphological difference, and substitutions. A model can semantically translate to the correct dialect yet not the exact word matching which leads to lower BLEU and ROUGE scores. Conversely, high lexical overlap does not always guarantee

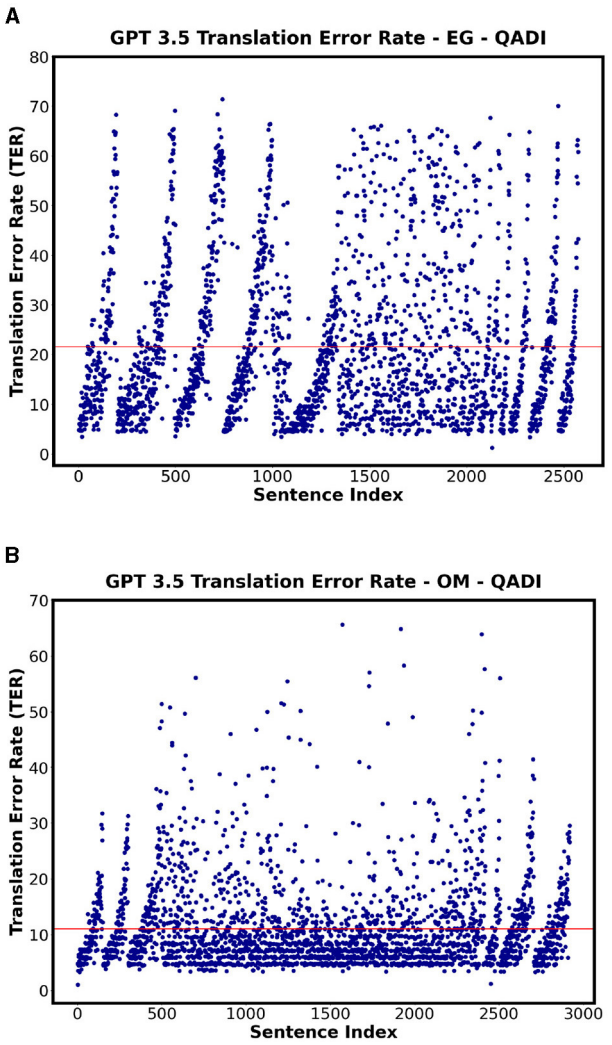

FIGURE 10
Scatter plots showing the TER for QADI datasets on GPT 3.5 for highest and lowest countries. **(A)** GPT 3.5 - EG Highest TER. **(B)** GPT 3.5 - OM Lowest TER.

TABLE 15 TER Comparison for Bard, GPT 3.5, GPT 4, and GPT 5 for each dialect in the MADAR dataset, where lower TER means higher performance.

Dialect	Bard	GPT 3.5	GPT 4	GPT 5
JO	7.32%	7.11%	7.10%	6.95%
LB	6.54%	6.37%	6.36%	6.27%
IQ	6.66%	6.53%	6.49%	6.35%
DZ	7.14%	6.95%	6.93%	6.95%
EG	7.16%	7.02%	7.00%	6.88%
LY	7.06%	6.90%	6.89%	6.71%
MA	7.17%	7.10%	7.02%	6.88%
OM	7.20%	7.10%	7.04%	6.88%
PL	6.73%	6.57%	6.57%	6.41%
QA	6.49%	6.40%	6.35%	6.23%
SA	6.75%	6.61%	6.60%	6.50%
SD	7.14%	7.02%	7.03%	6.83%
SY	6.56%	6.38%	6.42%	6.30%
TN	6.71%	6.52%	6.53%	6.37%
YE	6.93%	6.75%	6.78%	6.61%

High error rates are colored by red, lower rates are denoted by green.

semantic accuracy if the matched words are used in a different sense. The is noted in Table 9, and some dialects such as DZ and EG scored low BLEU/ROUGE scores while achieving high values in the semantic evaluation perspective. These findings support our approach and analysis, highlighting the need to adapt different metric scores, as each captures different aspects of LLM translation quality.

4.3 Effects of model accuracy

4.3.1 Few-shots analysis

In this section, we will explore the opportunity to check whether increasing the prompt size from zero-shot to few-shot would enhance the translation quality of each LLM. We used the MADAR dataset as it has more consistency in results with TN having the lowest similarity scores in Table 18 and a high TER rate as shown in Table 19, indicating a need to enhance the translation quality of this dialect. In both datasets, the models showed the least translation performance for the Tunisian dialect, and this is correspondence to Sallam and Mousa (2024) research as well. QADI showed inconsistency in similarity scores. Which could be attributed to the fact that QADI gathers its sentences from X platform, which means that although the sentences are gathered from the same geolocation, this does not mean that they all belong to the same dialect.

Although adding a few-shot approach provides models with additional examples and reference points, most models exhibited a decline performance in compared to zero-shot. This is illustrated in Tables 20, 21. In particular, GPT 3.5 showed consistency, with no significant differences between the zero-shot and few-shot approach. Suddenly, GPT 4 translated almost 35% of the input sentences into English despite clear instructions. This might be explained by the model's biases or training to adapt English translations in unclear contexts for the model. Given that the few-shot prompt is considered as a long prompt and has several examples and details, GPT 4 might find the prompt ambiguous and refer to the default language setting, which is "English".

4.3.2 Impact of sentence length on model accuracy

This subsection analyzes the impact of sentence length on translation accuracy, hence addressing the third research question. Since the universal similarity encoder is used to compare two sentences, it enabled us to explore the correlation.

For QADI dataset, the highest correlation was 0.42 in MA for GPT 4. The highest correlation for Bard was 0.39 in QA. GPT 3.5 showed a low correlation between the sentence length and the translation accuracy (i.e., similarity between input and output).

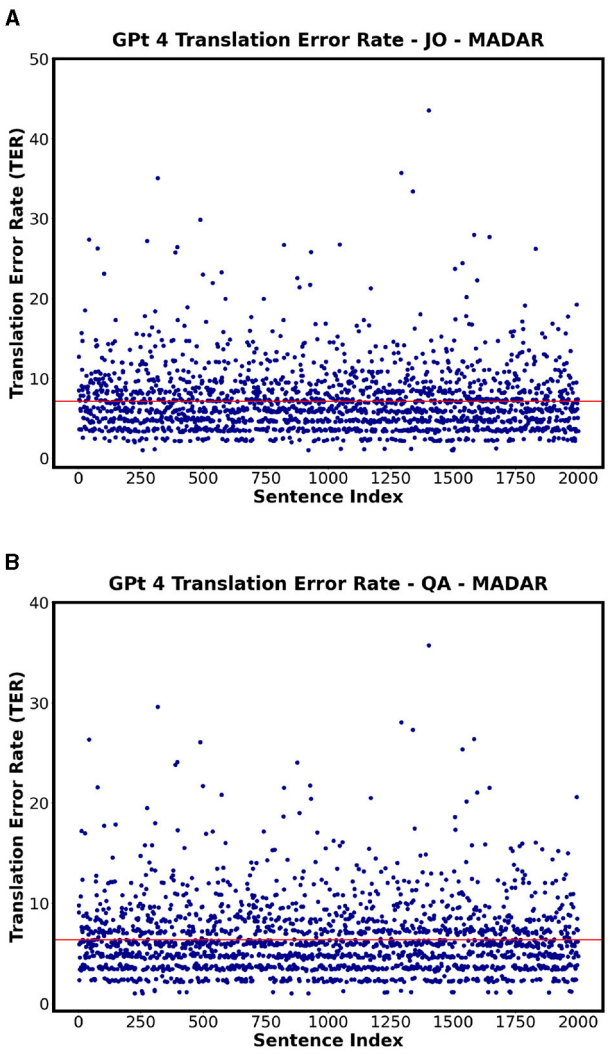

FIGURE 11
Scatter plots showing the TER for MADAR dataset on GPT 4 for each corresponding country. **(A)** GPT 4 - JO Highest TER. **(B)** GPT 4 - QA Lowest TER.

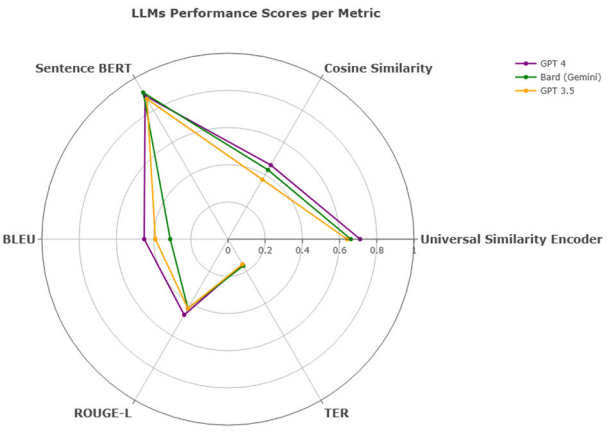

FIGURE 12
LLMs performance scores per metric - QADI dataset.

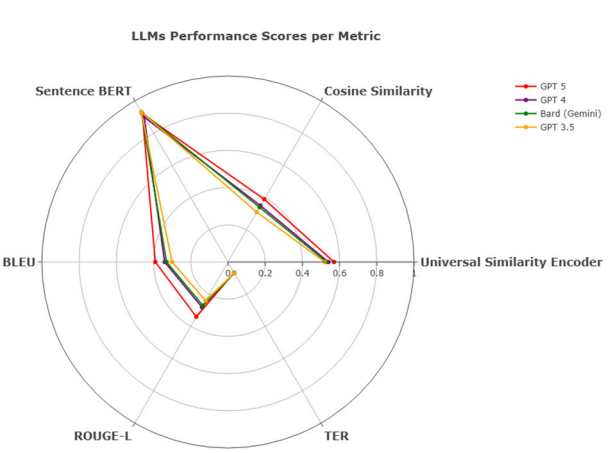

FIGURE 13
LLMs performance scores per metric - MADAR dataset.

Figure 14 visualizes the results where showing no strong correlation between the sentence length and the universal similarity encoder. Such low positive correlations indicate that there is no relation between the sentence length and the accuracy of the translation.

For MADAR, GPT 3.5/4 show a weak correlation, yet the highest compared to Bard with a value of 0.24 for some Maghreb Countries (i.e., DZ, MA, and TN) where Bard show no significant correlation. Figure 14 supports this finding as GPT 3.5/4 indicate a broader range of similarity scores as sentence length varies.

5 Conclusion

5.1 Concluding remarks

The study utilizes the QADI and MADAR datasets to evaluate the performance of GPT 3.5, GPT 4, and Bard (Gemini) in translating MSA to Arabic dialects, with GPT 5 evaluated exclusively on the MADAR dataset. Several performance metrics such as cosine similarity, universal similarity encoder, sentence BERT, BLEU, ROUGE, and TER were used to test the models' efficiency and accuracy. The analysis revealed close translations among LLMs in similarity and error rate. In QADI dataset, there was a significant difference between the models where GPT 4 was the best LLM in translating MSA to Arabic dialects showing a p-value of 0.000006 through ANOVA test on cosine similarity metric. It shows significant difference on all metrics except for BLEU and TER. For the MADAR dataset, there were no significant differences in translation performance measuring on sentence BERT, ROUGE-L, and TER. However, the results show significant differences through universal similarity encoder, cosine similarity, and BLEU, with GPT 5 being the top performer. GPT 4 demonstrates the best performance across both datasets (MADAR and QADI); it consistently showed high translation quality with low error rates. This proves the models sufficiency and the ability to be used in several dialect contexts and applications. GPT-4 showed consistent high translation scores for the majority of metrics, specifically on Levantine and Egyptian dialects; however, it shows low results on Maghrebi regions such as Tunisian dialect. Overall, GPT-4 provides the most reliable performance while GPT 5 outperforms all models

TABLE 16 ANOVA results for models per metric - QADI dataset.

Metric	*p*-value	*F*-statistic
Universal similarity encoder	0.009111	7.65
Cosine similarity	0.000006	28.85
Sentence BERT	0.000068	20.57
BLEU	0.058	3.85
ROUGE-L	0.00018	0.16
TER	0.56	0.59

TABLE 17 ANOVA results for models per metric - MADAR dataset.

Metric	*p*-value	*F*-statistic
Universal similarity encoder	0.005	4.64
Cosine similarity	0.00009	8.57
Sentence BERT	0.44	0.91
BLEU	0.000029	9.73
ROUGE-L	0.68	7.87
TER	0.31	1.2

TABLE 18 Countries with lowest values in MADAR dataset similarity metrics.

Model	Univ. Sim. Enc.	Cosine Sim.	Sent. BERT	BLEU	ROUGE
Bard	TN	TN	PL	TN	TN
GPT 3.5	TN	TN	LB	TN	TN
GPT 4	MA but TN similar score	TN–MA	LB	TN–MA	TN
GPT 5	DZ	DZ	Not applicable	DZ	DZ

TABLE 19 Countries with highest TER values in MADAR dataset.

Model	TER
Bard	JO but TN similar score
GPT 3.5	JO but TN similar score
GPT 4	JO but TN similar score
GPT 5	JO but DZ similar score

TABLE 20 Tunisia zero-shot metric performance.

Model	USE	Cosine Sim	S-BERT	BLEU	Rouge-L	TER
Bard	0.48	0.26	0.93	0.25	0.41	6.71%
GPT 3.5	0.48	0.24	0.93	0.23	0.49	6.52%
GPT 4	0.48	0.26	0.93	0.25	0.45	6.53%

TABLE 21 Tunisia few-shot metric performance.

Model	USE	Cosine Sim	S-BERT	BLEU	Rouge-L	TER
Bard	0.47	0.23	0.93	0.21	0.15	6.77%
GPT 3.5	0.48	0.24	0.92	0.24	0.16	6.53%
GPT 4	0.32	0.20	0.93	0.20	0.12	6.64%

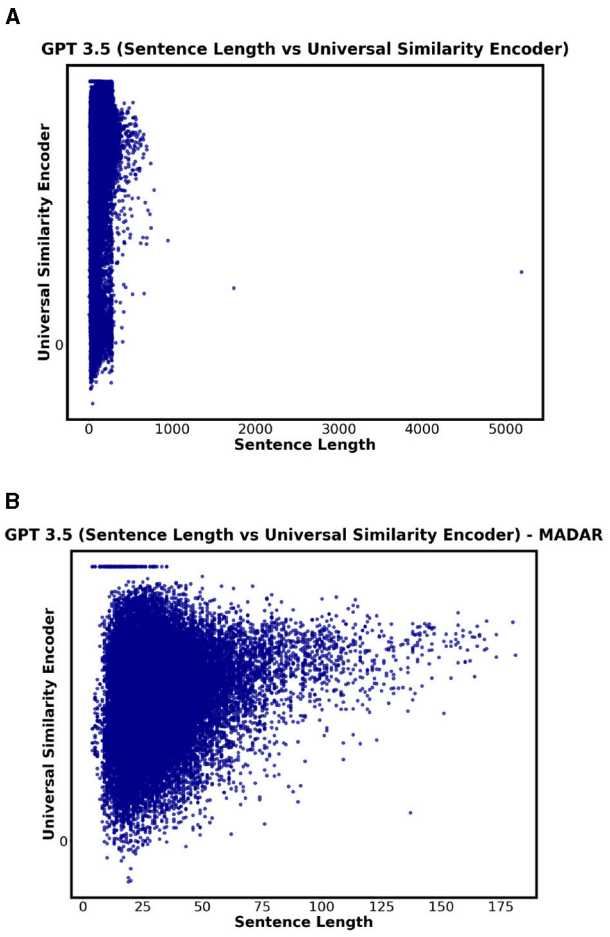

FIGURE 14
Correlation (sentence length vs universal similarity) for GPT 3.5. **(A)** GPT 3.5 - QADI. **(B)** GPT 3.5 - MADAR.

specifically on the MADAR dataset in finding sentences overlap measured by BLEU and ROUGE-L.

However, its performance is not uniform across all dialects' while it excels in dialects with larger training representation (e.g., Egyptian and Levantine), the accuracy slightly decreases in underrepresented dialects (e.g., Maghrebi). On the MADAR dataset, GPT-5 shows particularly strong performance on overlap-sensitive metrics such as BLEU and ROUGE-L, suggesting it captures sentence-level correspondences more effectively. Taken together, GPT-4 provides the most reliable overall performance across both datasets, while GPT-5 demonstrates an emerging advantage in fine-grained similarity for MADAR dialectal translations.

Furthermore, models have shown TER rates ranging from 6% up to 25%, indicating that despite slight errors, their translations are generally considered to be of good quality. However, GPT has shown better response to a given prompt in terms of output

results compared to Bard (Gemini). GPT in all versions specifically GPT 5 showed the best results for translation through the Levant countries. Zero-shot prompts were adapted for all countries, while few-shot for the country with the least translation performance, Tunisia. Unexpectedly, the few-shot technique did not enhance the performance of translation especially for Bard (Gemini) and GPT 4 as they performed worse while GPT 3.5 performed consistently in both prompting techniques. Overall, all LLMs proved capable and efficient in translating diverse Arabic dialects from over 15 countries to provide valuable insights for future applications in NLP.

This research establishes a benchmark for Arabic dialect translation and derives significant findings for advancing NLP capabilities in Arabic, paving the way for more inclusive and efficient models that address the linguistic diversity of the Arab world. Other researchers in the field may rely on GPT 4 and GPT 5 over GPT 3.5 and adapt Bard (Gemini), considering them feasible and effective LLMs for handling underrepresented languages, particularly Arabic and its linguistic complexities. The study also opens opportunities for future work, such as incorporating open source models, improving data sets, and optimizing prompting techniques. Moreover, we show the impact of few-shot prompting and how its impact was not significant, which could be replaced by other alternatives or prompt engineering techniques in future or relevant works.

5.2 Future works

We are aiming to extend this research by incorporating additional Arabic LLMs and other well-known applicable LLMs to generalize our findings. In addition, more data samples and datasets can be included to strengthen the analysis. Looking ahead, enhancing prompt and prompting techniques to optimize the translation process would add value to this research.

5.3 Limitations

This study faces several limitations that could influence the study results. Despite their remarkable success in various NLP tasks and the popularity of closed-source LLMs, models such as GPT 3.5, GPT 4, and GPT 5 have several limitations (Yu et al., 2023). These models are accessed through APIs which eliminates the need for computer infrastructure. Although cloud-based AI services are easy to use, they lack control over processing or training data. Furthermore, it is challenging to produce studies on closed-source models due to the high expense of conducting experiments through APIs. Another limitation is that the LLMs are closed models, as the name suggests, closed LLMs lack transparency in their internal architecture and training process, making it difficult for researchers to fully understand the output generation. The limitations also include cost constraints while running LLMs such as GPT 3.5/4 and Bard (Gemini) which results in running only 50K out of 500K samples in QADI dataset. Expanding the sample size in future studies could improve the robustness and reliability of the results. Moreover, both GPT and Bard (Gemini) had restrictions on the rate limit (i.e., the number of API requests). Thus, limiting the running process of the data to a specific rate daily, this consumed the time to complete the running. It is possible that recently published versions have increased the rate limit, which could be explored. In addition, there is lack in LLMs that can deal with Arabic dialects; having more LLMs would definitely strengthen the comparison. While this study adapted datasets encompassing 15 to 18 dialects, it does not cover all 22 Arabic-speaking countries, thus limiting the generalizability of the findings. Furthermore, QADI dataset, which is collected from X, may require cleaning to remove slang and informal expressions in social media, which can improve the quality of translation outputs. In addition, only one dataset (i.e., MADAR) had a MSA baseline, which was substituted by a back-translation algorithm for the QADI dataset. This approach may potentially limit the accuracy and effectiveness of the translations derived from QADI dataset. Moreover, the MADAR dataset exhibits a travel domain bias, which may affect the findings and limit the model's translation capability to other domains. In some cases, the models were not able to translate the dialect, resulting in an empty output, English translated sentence instead of Arabic or incomplete response. Finally, since most of the metrics are calculated as mean scores with only a single inferential statistical test (ANOVA) applied, generalizing the results might be tricky.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

AB: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. KM: Conceptualization, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. FG: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. IA: Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing. SA: Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Abdelali, A., Mubarak, H., Samih, Y., Hassan, S., and Darwish, K. (2020). Arabic dialect identification in the wild. *arXiv preprint arXiv:2005.06557*.

Abdelaziz, A. A. A., Elneima, A. H., and Darwish, K. (2024). "LLM-based MT data creation: Dialectal to MSA translation shared task," in *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation@ LREC-COLING 2024*, 112–116.

Abu-Haidar, F. (2011). "Shifting boundaries: the effect of msa on dialect convergence in Baghdad," in *Perspectives on Arabic Linguistics: Papers from the Annual Symposium on Arabic Linguistics. Volume IV: Detroit, Michigan 1990* (John Benjamins Publishing Company), 91–106. doi: 10.1075/cilt.85.07abu

Alabdullah, A., Han, L., and Lin, C. (2025). Advancing dialectal Arabic to modern standard Arabic machine translation. *arXiv preprint arXiv:2507.20301*.

Alahmari, S., Atwell, E., and Saadany, H. (2024). "Sirius_Translators at OSACT6 2024 shared task: fin-tuning ara-T5 models for translating arabic dialectal text to modern standard Arabic," in *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation@ LREC-COLING 2024* (ELRA and ICCL), 117–123.

Al-Gaphari, G., and Al-Yadoumi, M. (2010). A method to convert Sana'ani accent to modern standard Arabic. *Int. J. Inf. Sci. Manag.* 8, 39–49.

Alimi, T., Boujelbene, R., Derouich, W., and Belguith, L. (2024). Fine-tuned transformers for translating multi-dialect texts to modern standard Arabic. *Int. J. Cogn. Lang. Sci.* 18, 679–684.

Allingham, J. U., Ren, J., Dusenberry, M. W., Gu, X., Cui, Y., Tran, D., et al. (2023). "A simple zero-shot prompt weighting technique to improve prompt ensembling in text-image models," in *International Conference on Machine Learning* (PMLR), 547–568.

Alyafeai, Z., Alshaibani, M. S., AlKhamissi, B., Luqman, H., Alareqi, E., and Fadel, A. (2023). Taqyim: evaluating Arabic NLP tasks using ChatGPT models. *arXiv preprint arXiv:2306.16322*.

Atwany, H., Rabih, N., Mohammed, I., Waheed, A., and Raj, B. (2024). "OSACT 2024 task 2: Arabic dialect to MSA translation," in *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation@ LREC-COLING 2024*, 98–103.

Baert, G., Gahbiche, S., Gadek, G., and Pauchet, A. (2020). "Arabizi language models for sentiment analysis," in *Proceedings of the 28th International Conference on Computational Linguistics*, 592–603. doi: 10.18653/v1/2020.coling-main.51

Bakr, H. A., Shaalan, K., and Ziedan, I. (2008). "A hybrid approach for converting written egyptian colloquial dialect into diacritized Arabic," in *The 6th International Conference on Informatics and Systems, infos2008. Cairo University* (Citeseer).

Baniata, L. H., Kang, S., and Ampomah, I. K. E. (2022). A reverse positional encoding multi-head attention-based neural machine translation model for Arabic dialects. *Mathematics* 10:3666. doi: 10.3390/math10193666

Baniata, L. H., Park, S., and Park, S.-B. (2018). A neural machine translation model for Arabic dialects that utilizes multitask learning (MTL). *Comput. Intell. Neurosci.* 2018:7534712. doi: 10.1155/2018/7534712

Behr, D. (2017). Assessing the use of back translation: the shortcomings of back translation as a quality testing method. *Int. J. Soc. Res. Methodol.* 20, 573–584. doi: 10.1080/13645579.2016.1252188

Bhat, S., Varma, V., and Pedanekar, N. (2023). "Generative models for indic languages: evaluating content generation capabilities," in *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, 187–195. doi: 10.26615/978-954-452-092-2_021

Bouamor, H., Habash, N., and Oflazer, K. (2014). "A multidialectal parallel corpus of Arabic," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland* (European Language Resources Association (ELRA)), 1240–1245.

Bouamor, H., Hassan, S., and Habash, N. (2019). "The MADAR shared task on Arabic fine-grained dialect identification," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, 199–207. doi: 10.18653/v1/W19-4622

Chan, V., and Tang, W. K.-W. (2024). "GPT and translation: a systematic review," in *2024 International Symposium on Educational Technology (ISET)* (IEEE), 59–63. doi: 10.1109/ISET61814.2024.00021

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., et al. (2023). A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.* 15, 1–45. doi: 10.1145/3641289

De Varda, A., and Marelli, M. (2023). "Scaling in cognitive modelling: a multilingual approach to human reading times," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 139–149. doi: 10.18653/v1/2023.acl-short.14

Demidova, A., Atwany, H., Rabih, N., and Sha'ban, S. (2024). "Arabic train at NADI 2024 shared task: LLMs' ability to translate Arabic dialects into Modern Standard Arabic," in *Proceedings of the Second Arabic Natural Language Processing Conference* (Bangkok, Thailand: Association for Computational Linguistics). doi: 10.18653/v1/2024.arabicnlp-1.80

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Eberhard, D., Simons, G., and Fennig, C. (2019). *Ethnologue: Languages of Asia, Twenty-Second Edition*. Ethnologue Series. Sil International, Global Publishing.

Guellil, I., Azouaou, F., and Abbas, M. (2017). "Neural vs. statistical translation of Algerian Arabic dialect written with Arabizi and Arabic letter," in *The 31st Pacific Asia Conference on Language, Information and Computation Paclic*.

Guellil, I., Saâdane, H., Azouaou, F., Gueni, B., and Nouvel, D. (2021). Arabic natural language processing: an overview. *J. King Saud Univ. Comput. Inf. Sci.* 33, 497–507. doi: 10.1016/j.jksuci.2019.02.006

Hadi, M. U., al tashi, Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., et al. (2023). Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints* 1, 1–26. doi: 10.36227/techrxiv.23589741.v2

Hamada, S., and Marzouk, R. M. (2018). *Developing a Transfer-Based System for Arabic Dialects Translation*. Cham: Springer International Publishing, 121–138. doi: 10.1007/978-3-319-67056-0_7

Hamed, M. M., Hreden, M., Hennara, K., Aldallal, Z., Chrouf, S., and AlModhayan, S. (2025). "Lahjawi: Arabic cross-dialect translator," in *Proceedings of the 4th Workshop on Arabic Corpus Linguistics (WACL-4)*, 12–24.

Harrat, S., Meftouh, K., and Smaili, K. (2019). Machine translation for Arabic dialects (survey). *Inf. Proc. Manag.* 56, 262–273. doi: 10.1016/j.ipm.2017.08.003

Jiang, E., Olson, K., Toh, E., Molina, A., Donsbach, A., Terry, M., et al. (2022). "Promptmaker: prompt-based prototyping with large language models," in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 1–8. doi: 10.1145/3491101.3503564

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. (2020). How can we know what language models know? *Trans. Assoc. Comput. Linguist.* 8, 423–438. doi: 10.1162/tacl_a_00324

Jibrin, F., Mughaus, R., Abudalfa, S., Ahmed, M., and Abdelali, A. (2025). An empirical evaluation of Arabic text formality transfer: a comparative study. *Lang. Resour. Evaluat.* 2024, 1–16.

Joshi, A., Dabre, R., Kanojia, D., Li, Z., Zhan, H., Haffari, G., et al. (2024). Natural language processing for dialects of a language: a survey. *arXiv preprint arXiv:2401.05632*.

Kadaoui, K., Magdy, S. M., Waheed, A., Khondaker, M. T. I., El-Shangiti, A. O., Nagoudi, E. M. B., et al. (2023). TARJAMAT: evaluation of bard and chatGPT on machine translation of ten Arabic varieties. *arXiv preprint arXiv:2308.03051*.

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Indiv. Differ.* 103:102274. doi: 10.1016/j.lindif.2023.102274

Keselman, H. J., Huberty, C. J., Lix, L. M., Olejnik, S., Cribbie, R. A., Donahue, B., et al. (1998). Statistical practices of educational researchers: an analysis of their ANOVA, MANOVA, and ANCOVA analyses. *Rev. Educ. Res.* 68, 350–386. doi: 10.3102/00346543068003350

Kheiri, K., and Karimi, H. (2023). SentimentGPT: exploiting GPT for advanced sentiment analysis and its departure from current machine learning. *arXiv preprint arXiv:2307.10234*.

Khered, A. S., Benkhedda, Y., and Batista-Navarro, R. T. (2025). "Dial2MSA-verified: a multi-dialect arabic social media dataset for neural machine translation to modern standard Arabic," in *Proceedings of the 4th Workshop on Arabic Corpus Linguistics (WACL-4)*, 50–62.

Khondaker, M. T. I., Waheed, A., Nagoudi, E. M. B., and Abdul-Mageed, M. (2023). GPTAraEval: a comprehensive evaluation of ChatGPT on Arabic NLP. *arXiv preprint arXiv:2305.14976*.

Khoshafah, F. (2023). *ChatGPT for Arabic-English translation: Evaluating the accuracy*. Europe PubMed Central (PMC) Repository. doi: 10.21203/rs.3.rs-2814154/v1

Koubaa, A., Ammar, A., Ghouti, L., Najar, O., and Sibaee, S. (2024). ArabianGPT: Native Arabic GPT-based Large Language. *arXiv preprint arXiv:2402.15313*.

Lilli, S. (2023). ChatGPT-4 and Italian dialects: assessing linguistic competence. *Umanistica Digitale* 16, 235–263.

Lin, C.-Y., and Hovy, E. (2003). "Automatic evaluation of summaries using N-gram co-occurrence statistics," in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 150–157. doi: 10.3115/1073445.1073465

López Espejel, J., Ettifouri, E. H., Yahaya Alassan, M. S., Chouham, E. M., and Dahhane, W. (2023). GPT-3.5, GPT-4, or BARD? Evaluating LLMs reasoning ability in zero-shot setting and performance boosting through prompts. *Nat. Lang. Proc. J.* 5:100032. doi: 10.1016/j.nlp.2023.100032

Malaysha, S., El-Haj, M., Ezzini, S., Khalilia, M., Jarrar, M., Almujaiwel, S., et al. (2024). AraFinNlp 2024: The first Arabic financial NLP shared task. *arXiv preprint arXiv:2407.09818*.

Mashaabi, M., Al-Khalifa, S., and Al-Khalifa, H. (2024). A survey of large language models for Arabic language and its dialectss. *arXiv preprint arXiv:2410.20238*.

Mohamed, E., Mohit, B., and Oflazer, K. (2012). "'Transforming standard arabic to colloquial Arabic," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 176–180.

Mrinalini, K., Vijayalakshmi, P., and Nagarajan, T. (2022). SBSim: a sentence-BERT similarity-based evaluation metric for indian language neural machine translation systems. *IEEE/ACM Trans. Audio, Speech, Lang. Proc.* 30, 1396–1406. doi: 10.1109/TASLP.2022.3161160

Okpor, M. D. (2014). Machine translation approaches: issues and challenges. *Int. J. Comput. Sci. Issues* 11:159.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318. doi: 10.3115/1073083.1073135

Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., et al. (2023). The RefinedWeb dataset for falcon LLM: outperforming curated corpora with web data only. *arXiv preprint arXiv:2306.01116*.

Peng, K., Ding, L., Zhong, Q., Shen, L., Liu, X., Zhang, M., et al. (2023). Towards making the most of ChatGPT for machine translation. *arXiv preprint arXiv:2303.13780*.

Ridouane, T., and Bouzoubaa, K. (2014). "A hybrid approach to translate Moroccan Arabic dialect," in *2014 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14)* (IEEE), 1–5.

Sallam, M., and Mousa, D. (2024). Evaluating ChatGPT performance in Arabic dialects: a comparative study showing defects in responding to Jordanian and Tunisian general health prompts. *Mesopot. J. Artif. Intell. Healthcare*. 2024, 1–7. doi: 10.58496/MJAIH/2024/001

Salloum, W., and Habash, N. (2012). "Elissa: a dialectal to standard Arabic machine translation system," in *Proceedings of COLING 2012: Demonstration Papers*, 385–392.

Sghaier, M. A., and Zrigui, M. (2020). Rule-based machine translation from Tunisian dialect to modern standard Arabic. *Procedia Comput. Sci.* 176, 310–319. doi: 10.1016/j.procs.2020.08.033

Shaikh, S., Daudpota, S. M., Yayilgan, S. Y., and Sindhu, S. (2023). "Exploring the potential of large-language models (LLMs) for student feedback sentiment analysis," in *2023 International Conference on Frontiers of Information Technology (FIT)* (IEEE), 214–219. doi: 10.1109/FIT60620.2023.00047

Sibaee, S., Nacar, O., Al-Habashi, Y., Ammar, A., and Boulila, W. (2025). SHAMI-MT: a Syrian Arabic dialect to modern standard Arabic bidirectional machine translation system. *arXiv preprint arXiv:2508.02268*.

Steele, J. L. (2023). To GPT or not GPT? Empowering our students to learn with AI. *Comput. Educ.* 5:100160. doi: 10.1016/j.caeai.2023.100160

Wright, W., and Caspari, C. P. (2011). *A Grammar of the Arabic Language*. Cosimo, Inc.

Yong, Z.-X., Schoelkopf, H., Muennighoff, N., Aji, A. F., Adelani, D. I., Almubarak, K., et al. (2022). BLOOM+1: adding language support to BLOOM for zero-shot prompting. *arXiv preprint arXiv:2212.09535*.

Yu, H., Yang, Z., Pelrine, K., Godbout, J. F., and Rabbany, R. (2023). Open, closed, or small language models for text classification? *arXiv preprint arXiv:2308.10092*.

Zhang, L., Fan, H., Peng, C., Rao, G., and Cong, Q. (2020). "Sentiment analysis methods for hpv vaccines related tweets based on transfer learning," in *Healthcare* (MDPI), 307. doi: 10.3390/healthcare8030307

Zhu, W., Liu, H., Dong, Q., Xu, J., Kong, L., Chen, J., et al. (2023). Multilingual machine translation with large language models: empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

frontiers | Frontiers in Artificial Intelligence

TYPE Systematic Review
PUBLISHED 16 October 2025
DOI 10.3389/frai.2025.1666349

OPEN ACCESS

EDITED BY
Shadi Abudalfa,
King Fahd University of Petroleum and Minerals, Saudi Arabia

REVIEWED BY
Baligh Babaali,
University of Medea, Algeria
Aadil Ganie,
Universitat Politècnica de València, Spain

*CORRESPONDENCE
Hooayda Allwaibed
✉ Hooayda@student.usm.my
Selvakumar Manickam
✉ selva@usm.my

RECEIVED 25 July 2025
ACCEPTED 29 September 2025
PUBLISHED 16 October 2025

CITATION
Allwaibed H, Anbar M, Manickam S and Bintang A (2025) Cyberbullying detection approaches for Arabic texts: a systematic literature review.
Front. Artif. Intell. 8:1666349.
doi: 10.3389/frai.2025.1666349

Cyberbullying detection approaches for Arabic texts: a systematic literature review

Hooayda Allwaibed[1,2]*, Mohammed Anbar[1], Selvakumar Manickam[1]* and Annisa Bintang[3]

[1]Cybersecurity Research Centre (CYRES), Universiti Sains Malaysia (USM), Penang, Malaysia, [2]Department of Computer Science, Applied College, Northern Border University, Arar, Saudi Arabia, [3]Universitas Indonesia Fakultas Ilmu Komputer, Depok, Indonesia

This study presents a comprehensive review of current methodologies, trends, and challenges in cyberbullying detection within Arabic-language contexts, with a focus on the unique linguistic and cultural factors associated with Arabic. This study reviews 35 peer-reviewed articles about the identification of cyberbullying in Arabic text. Reported accuracies across datasets and platforms range from approximately 73 to 96%, with precision frequently surpassing recall, suggesting that systems are more adept at identifying blatant bullying than at encompassing all pertinent instances. Methodologically, conventional machine learning utilizing Arabic-specific characteristics remains effective on smaller datasets, however deep neural architectures—especially CNN/BiLSTM—and transformer models like AraBERT yield superior outcomes when dialectal heterogeneity and orthographic noise are mitigated. Evaluation methodologies differ; research using a neutral class frequently indicates exaggerated accuracy, underscoring the necessity to emphasize macro-averaged F1 and per-class metrics. The evidence underscores deficiencies in dialectal representativeness, the uniformity of bullying notions compared to general abuse, and the transparency of annotation processes. Ethical and deployment considerations—privacy preservation, dialectal bias, and real-time robustness—are becoming increasingly significant. We integrate trends (models and features), standards (labeling and metrics), and future work directions, encompassing dialect-robust pretraining, cross-dataset evaluation, context-aware modeling, and human-in-the-loop frameworks. The review offers a comprehensive basis for researchers and practitioners pursuing culturally and linguistically tailored approaches to Arabic cyberbullying detection.

KEYWORDS

cyberbullying detection, Arabic language, systematic literature review, machine learning, deep learning, support vector machines, convolutional neural networks, natural language processing

1 Introduction

The extensive utilization of digital communication channels has resulted in a concerning rise in cyberbullying, a type of online harassment impacting persons of many age groups and demographics. This study evaluated the relevant research published from 2014 to 2024, to assess and contrast the efficacy of conventional machine learning methods, deep learning frameworks, and sentiment-oriented strategies in the classification of cyberbullying, highlighting the significance of linguistic and dialectal intricacies in detection precision.

IT communication platforms such as WhatsApp, Facebook Messenger, Viber, WeChat, Line, Telegram, Imo, and Kakao Talk have increased in use throughout the last years, with some having over 1.5 billion users (Urrutia Zubikarai, 2020). Several sources contended that offensive content in social media and communication platforms has become extremely

dangerous; for instance, issues relating to social media in public institutions, particularly during the election period, are related to offensive content and have become challenging for public institutions in light of how information should be controlled (Grégoire et al., 2015). Offensive content, generally in the form of foul language spouting racial hate, personal attacks, and sexual harassment, is prevalent. Hence, it is important to detect offensive use of language to maintain a healthy discussion and enhance the security of users through the suppression of such hateful acts and offences (Bertini et al., 2021; Niraula et al., 2021). Online content-generators have increased, allowing more users to experience the freedom to express themselves, covered with anonymity if they choose, which maximizes the chance for platform misuse and leads to an environment that promotes offensive language and even eventually violence (Sap et al., 2019). Also, social networking platforms display several types of offensive language like hate speech, aggressive content, cyberbullying, and toxic statements (Mirończuk and Protasiewicz, 2018). A possible way to curtail and control such a phenomenon is through the use of NLP techniques like text classification for the automatic detection of offensive language. More specifically, text classification is the process of labelling new text with pre-defined labels (Mirończuk and Protasiewicz, 2018).

2 Background of study

2.1 Cyberbullying

Cyberbullying has become a global concern with the rise of social media and online platforms, and research efforts are increasingly being devoted to detecting and mitigating it using Machine Learning (ML), Deep Learning (DL), and Natural Language Processing (NLP) approaches. While a significant amount of research has been conducted in languages like English, studies targeting cyberbullying in Arabic remain limited. This systematic literature review aims to explore existing research on cyberbullying detection in the Arabic language, with a focus on ML and DL techniques, and to identify future research directions based on the analysis of the reviewed studies.

2.2 Challenges in detecting in Arabic language

Identifying cyberbullying in the Arabic language poses difficulties, mostly due to the linguistic, cultural, and computational intricacies involved in processing Arabic content. A principal challenge is the significant range of Arabic dialects, which differ not only by area but also by socio-economic and cultural factors. Although Modern Standard Arabic (MSA) is extensively employed in formal discourse, social media exchanges primarily transpire in dialectal Arabic, which is characterized by the absence of standardized spelling, syntax, and vocabulary (Mubarak and Darwish, 2019; AbdelHamid et al., 2022). The lack of high-quality, labeled datasets that consider these changes intensifies the issue, resulting in diminished model performance in real-world Arabic cyberbullying detection tasks (Bashir and Bouguessa, 2021; Khairy et al., 2023). A fundamental problem is the morphological complexity and intricate syntax of Arabic, which markedly contrasts with Indo-European languages like English. Arabic lexicon demonstrates significant inflexion through affixation, root-based derivations, and contextual variants, complicating tokenization, stemming, and lemmatization (Alakrot et al., 2018; Haidar et al., 2019). The linguistic features create difficulty in text classification, as identical words may possess varying meanings based on diacritical marks, which are frequently absent in informal online communication. The scarcity of comprehensive pre-trained models tailored for Arabic dialects constrains the capacity of NLP algorithms to effectively identify harmful and abusive content (Alrashidi et al., 2023; Khezzar et al., 2023). Research indicates that sentiment analysis and lexicon-based methodologies can improve detection by identifying emotional indicators; however, their efficacy is limited by the necessity for manually curated lexicons specific to Arabic dialects (Farid and El-Tazi, 2020). An application of NLP that extracts structured information in the form of entities, entities' relationship and attributes describing them from unstructured documents in an automatic method is Information Extraction (IE) (Cowie and Lehnert, 1996). Besides, IE systems have been found effective in handling information overload issues, enabling the discernment of the most significant information portion from a huge portion of information in a timely and easy manner. On the whole, detection of offensive language online is possible through the development of a model using ML, AI, DL and NLP methods. This paper investigates the following research questions:

3 Research questions

> **Q1:** What are the current trends in cyberbullying detection for the Arabic language and which dialects do they cover?
>
> **Q2:** How cyberbullying been detected in previous studies based on standards that represent its definition and characteristics?
>
> **Q3:** What directions for future research in cyberbullying detection may be established based on the findings of this review?

4 Methodology

A systematic literature review was conducted to conduct a comprehensive analysis by focusing on existing studies from 2014 to 2024, evaluating trends and advancements in cyberbullying detection for Arabic texts. This methodology involves structured selection criteria to ensure that only relevant and high-quality sources are included. The Inclusion Criteria are as follows:

1. Studies published from 2014 to 2024
2. Articles in English
3. Research specific to Arabic text-based cyberbullying detection

The exclusion criteria were:

1. The research focused on social studies without technological elements
2. Studies in languages other than English and non-Arabic texts

3. Non-text-based detection methods (e.g., voice, image, video)
4. Conference papers and review articles

SLR protocol was applied to the study, the final selected studies were conducted, and theoretical and practical steps were taken while conducting the SLR.

5 Data sources and keywords

In the first step, four major research databases, ScienceDirect, Scopus, Web of Science, and Springer, were searched through queries, and as many papers as possible were collected. The search query is "detect" AND ("cyberbullying" OR "hate speech" OR "harassment" OR "offensive") AND ("machine learning" OR "natural language processing" OR "deep learning") AND "Arabic." Based on initial exclusion criteria, papers were selected after carefully reading the abstracts of the papers in the second step. A final list of papers is prepared after reading the full articles and applying further exclusion criteria (35 papers). Figure 1 depicts the literature review process.

6 Results

This review synthesizes findings from numerous studies on cyberbullying detection within Arabic-language content, identifying the main trends, challenges, and methodologies, including ML, DL, and sentiment analysis. The majority of the studies concentrated on cyberbullying detection, offensive language detection, and hate speech identification. A significant portion of the research applied to social media platforms like Twitter and YouTube. The focus was largely on identifying cyberbullying in dialects such as Saudi Arabian Arabic, Egyptian Arabic, and the Levantine dialects. The most frequently used machine learning models included Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF). For deep learning models, LSTM, CNN, and GRU were prominent. Ensemble techniques like stacking and boosting showed better performance compared to individual ML models. The datasets used in the reviewed studies varied widely in size, ranging from small manually annotated datasets to large datasets collected from social media. Many studies employed preprocessing techniques such as tokenization, stemming, lemmatization, and removal of hyperlinks or non-Arabic characters to clean the data before analysis. Preprocessing was critical in

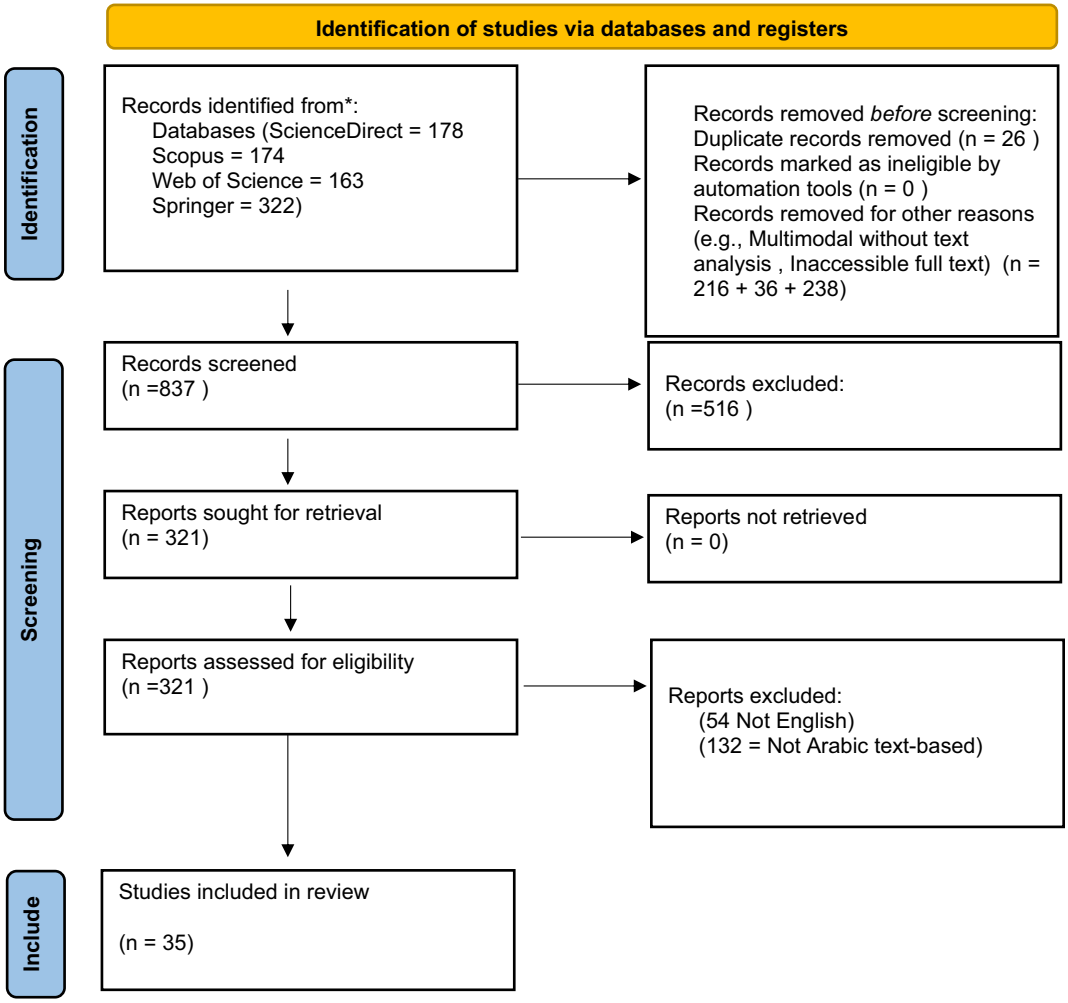

FIGURE 1
Systematic literature review process.

TABLE 1 Types of offensive language used in Arabic studies on cyberbullying and offensive content.

Type of Offensive Language	Description	Sources
Hate Speech	Language targeting specific groups based on religion, race, gender, or nationality. Includes:	Castaño-Pulgarín et al. (2021), Alsafari et al. (2020a, 2020b)
Insults and Personal Attacks	Abusive language aimed at degrading individuals, including name-calling, derogatory remarks, and personal insults about appearance, intelligence, or social status.	Alshalabi et al. (2024),
Profanity and Vulgar Language	Taboo words or phrases generally considered offensive, including swear words and obscenities that are often censored on public platforms.	Rosenbaum (2019)
Sexual Harassment	Inappropriate comments or sexually explicit content targeting individuals, often related to gender-based discrimination.	Abdelmonem (2015), Bouhlila (2019), Bertini et al. (2021), Niraula et al. (2021)
Bullying and Harassment	Repeated or persistent offensive behavior aimed at intimidating or humiliating someone, often through derogatory remarks about personal life or achievements.	Kanan et al. (2020)
Stereotyping and Discrimination	Generalizations that promote negative stereotypes about specific groups (e.g., based on age, nationality, profession). Includes implicit bias and discriminatory remarks.	Alsafari et al. (2020a, 2020b)
Mockery and Sarcasm	Humorous or sarcastic language used to belittle or degrade individuals or groups, often through irony or exaggeration, which can vary in offensiveness depending on context.	Abu Farha (2023).

ensuring the effectiveness of the ML and DL models. Across the reviewed studies, model performance is generally strong, with traditional machine learning and deep learning approaches demonstrating reliable detection capabilities in Arabic cyberbullying contexts. Reported results indicate that precision commonly exceeds recall, suggesting that systems are better at correctly identifying bullying instances than capturing all relevant cases. This pattern appears in works employing classical classifiers as well as ensemble strategies, with examples including Egyptian-dialect tweet classification (Farid and El-Tazi, 2020), Naïve Bayes–based detection pipelines (Mouheb et al., 2019), offensive language identification in user-generated video comments (Alakrot et al., 2018), and ensemble machine learning frameworks that optimize the balance of precision and recall (Haidar et al., 2019). In terms of offensive language and cyberbullying detection, researchers identify various types of offensive language, each reflecting specific social, cultural, and regional sensitivities. Table 1 illustrates the types of offensive language used in Arabic studies on cyberbullying and offensive content

6.1 Research question 1

The first research question was:

What are the current trends in cyberbullying detection for the Arabic language, and how do these trends account for various dialects?

The following themes were developed to answer the first research question 1:

6.1.1 Machine learning (ML) and deep learning (DL) approaches

Several studies have utilized ML and DL algorithms to detect cyberbullying, with Support Vector Machine (SVM) and Naïve Bayes (NB) being frequently applied (e.g., Haidar et al., 2017; Alakrot et al., 2018). More recently, DL methods such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated improved performance due to their ability to capture context and semantic meanings in text (e.g., Haidar et al., 2018; Mouheb et al., 2019; Mohaouchane et al., 2019). Ensemble learning, where multiple models are combined to improve prediction accuracy, has shown promise in boosting performance. For instance, stacking, boosting, and bagging techniques have demonstrated better performance in detecting Arabic cyberbullying content (e.g., Haidar et al., 2018; Khairy et al., 2023; Table 2).

6.1.2 Sentiment analysis and lexicon-based methods

Sentiment analysis, often coupled with lexicon-based approaches, is commonly used to detect harmful content. AlHarbi et al. (2019) and Farid and El-Tazi (2020) used sentiment-based lexicons for Arabic texts, finding that pointwise mutual information (PMI) and lexicon enhancement can improve detection accuracy. Sentiment-based approaches are also utilized alongside NLP tools, such as tokenization and stemming, for feature extraction, enhancing the ability to detect cyberbullying based on emotional cues.

6.1.3 Handling Arabic dialects and linguistic complexity

Dialectal Arabic presents a significant challenge, as standard ML models may not perform well on diverse dialects. Studies such as Alsubait and Alfageh (2021) and Al-Hassan and Al-Dossari (2022) indicate that datasets tailored to specific dialects (e.g., Egyptian, Levantine) enhance detection efficacy. Additionally, transformer-based models like AraBERT and multilingual BERT have emerged as effective tools for dealing with dialectal variations, as they can better capture semantic nuances across dialects (e.g., Khezzar et al., 2023; Alrashidi et al., 2023).

6.2 Research question 2

How has cyberbullying been detected in previous studies based on standards that represent its definition and characteristics?

The following themes were developed to answer the second research question.

TABLE 2 Summary of reviewed studies on Arabic hate/offensive/cyberbullying detection.

No	Study	Model(s)	Dataset and Platform	Dialect/ Domain	Performance Metrics	Limitations
1	Haidar et al. (2017)	Naïve Bayes, SVM	Posts (Twitter, Facebook, Formspring)	Saudi Arabic	NB: Precision 90.85%; SVM: Precision 0.815 (yes class)	Imbalanced dataset; few bullying instances; precision misleading
2	Haidar et al. (2018)	Feed-forward Neural Network (DL)	Twitter dataset (binary labels)	General Arabic	Validation accuracy 91.17% (7 hidden layers)	Limited to binary labels; dataset size not large
3	Alakrot et al. (2018)	SVM	YouTube comments	General Arabic	Precision 90.05%	Small dataset; not specific to cyberbullying
4	AlHarbi et al. (2019)	Lexicon + Sentiment Analysis (PMI, Chi-square, Entropy)	Tweets	Twitter (Saudi Arabic)	PMI accuracy 81% vs. Chi-square 62.11%	Lexicon-based; potential bias; dataset context-limited
5	Mubarak and Darwish (2019)	ML classifiers	Arabic tweets	General Arabic	High classification accuracy	Focused only on offensive, not cyberbullying
6	Farid and El-Tazi (2020)	Lexicon-based Sentiment Analysis + Emojis	Tweets in Modern Standard + Egyptian Dialect	Egyptian Arabic	Accuracy >73% for bullying hashtags	Lexicon limited; reliance on emojis and history
7	Alsafari et al. (2020b)	LR, LSTM, Sluice, BERT, ELMo, SVM	Labeled tweets (Twitter)	Mixed Arabic dialects	SVM + ngrams: Acc. 85.16%; CNN + mBERT F1-macro 66.86%	Limited samples per class; subjectivity in annotation
8	Bashir and Bouguessa (2021)	LSTM, SVM, Naïve Bayes	Twitter dataset (cyberbullying keywords)	General Arabic	LSTM accuracy 72%	Keyword-based data collection; lower accuracy
9	Fati (2022)	Sentiment Analysis Framework	Twitter	General Arabic	Accuracy 81% (10-fold CV)	Limited validation; binary annotation
10	Al-Hassan and Al-Dossari (2022)	LSTM, CNN + LSTM, GRU, CNN + GRU	Labeled tweets	General Arabic	CNN + LSTM: Precision 72%, Recall 75%, F1 73%	Moderate dataset size; limited categories
11	Alsubait and Alfageh (2021)	Multinomial NB, Complement NB, Logistic Regression	YouTube comments	General Arabic	Avg. F1: TF-IDF 77.9% vs. CountVec 77.5%	Dataset modest; no deep learning comparison
12	Alhashmi and Darem (2022)	RF, NB, SVM, XGB, ANN, Stacked DL; Consensus-Based Ensemble	(Twitter, WhatsApp, Vine, Instagram, Packet; incl. Translated data)	Mixed Arabic + translated	Consensus ensemble improved accuracy by 1.3% over best classifier; RF strongest	Dataset partly translated; mixed domains; modest gain over baselines
13	Bouliche and Rezoug (2022)	Dynamic Graph Neural Network (DGNN)	Arabic comments (tweets)	General Arabic	Accuracy 74%	Model performance modest; needs refinement; small dataset
14	El-Alami et al. (2022)	BERT (multilingual, transfer learning)	Bilingual dataset (English + Arabic tweets)	General Arabic + English	High accuracy and F1; BERT outperformed other models	Ambiguous language still difficult; early-stage
15	AbdelHamid et al. (2022)	AraBERT, ArabicBERT, GigaBERT vs. RF, SVM	Syrian/Levantine tweets	Levantine dialect	GigaBERT: AUC 94.6%, Macro F1 0.81	Focused on Levantine; dataset scope limited
16	AlFarah et al. (2022)	SVM, RF, NB, LR, KNN	Twitter + YouTube, oversampled	General Arabic	NB highest AUC 89%; SVM and LR also strong	Class imbalance; dataset moderate in size
17	Anezi (2022)	Deep Recurrent Neural Network (DRNN)	Custom Arabic comments dataset	General Arabic	Binary Acc 99.73%; 3-class Acc 95.38%; 7-class Acc 84.14%	Dataset unique but limited disclosure; overfitting risk
18	Althobaiti (2022)	BERT + Sentiment + Emoji features vs. SVM, LR	Arabic tweets	General Arabic	BERT model highest F1 across all tasks	Single dataset; limited external validation
19	Ali and Kurdy (2022)	SVM, SGD, KNN, LR, AdaBoost, Bagging	Syrian Facebook comments + questionnaire	Syrian slang	SVM and SGD accuracy 77%; AdaBoost precision 94%	Imbalanced recall (47%); small dataset

(Continued)

TABLE 2 (Continued)

No	Study	Model(s)	Dataset and Platform	Dialect/ Domain	Performance Metrics	Limitations
20	Alduailaj and Belghith (2023)	SVM + FarasaNLTK vs. NB	Twitter + YouTube comments	General Arabic	SVM best accuracy 95.74% (TF-IDF n-gram)	Keyword-based collection; possible bias
21	Khairy et al. (2023)	Ensemble (Voting) vs. LR, SVC, KNN	New balanced dataset	General Arabic	Voting model highest Acc, F1, Recall, Precision; LR best single Acc 65.1%	Small dataset; limited to ML
22	Rachidi et al. (2023)	ML (SVM, NB, RF, LR) and DL (LSTM)	Instagram Moroccan dialect	Moroccan Arabic	LSTM Acc 83.64%; SVM Acc 75.04%	Scarcity of tools/datasets for dialect; modest results
23	Alrashidi et al. (2023)	Fine-tuned Arabic BERT, Multi-task Learning	Multi-aspect abusive tweets dataset	General Arabic	MTL + BERT > DL baselines; GitHub data shared	Imbalanced datasets; Arabic only
24	Elzayady et al. (2023)	CNN-LSTM, CNN-BiLSTM, CNN-GRU, AraBERT +Personality Features	Twitter hate speech dataset	General Arabic	AraBERT + personality features Acc 82.3%; CNN-LSTM 77%	Personality inference adds complexity; dataset size moderate
25	Khezzar et al. (2023)	LR, SVC, DT, CNN, AraBERT; web app (arHateDetector)	arHateDataset (merged public sets), Twitter	Standard + dialectal Arabic	AraBERT accuracy 93%; precision/recall/F1 reported	Aggregated datasets may introduce label/definition drift; external validation not detailed
26	Alsafari et al. (2020a)	Single and ensemble CNN/BiLSTM; AraBERT vs. non-contextual embeddings	Twitter; fine-grained two/three/six-class corpora	Mixed Arabic dialects	Ensemble F1: 91% (2-class), 84% (3-class), 80% (6-class); AraBERT > non-contextual; CNN > BiLSTM	Class granularity increases difficulty; error analysis shows issues with implicit/ defensive language
27	Aljuhani et al. (2022)	BiLSTM with domain-specific embeddings; LR, SVM baselines	Tweets (seeded crawl, cleaned, labeled)	General Arabic (Twitter)	LR on char n-grams P/R/F1 = 92%; SVM ≈ 90%; BiLSTM competitive with domain embeddings	Seed-term collection bias; translation/generalization across topics not assessed
28	Amer Hamzah and Dhannoon (2023)	BiLSTM + Temporal Convolutional Network (TCN)	CASH: tweets on sexual harassment	Sexual-harassment domain	Accuracy 96.65%; F0.5 = 0.969; > XGBoost baseline	Task/domain specific; dialectal robustness not analyzed
29	Boulouard et al. (2022)	BERT EN, AraBERT, mBERT (AR/EN), LinearSVC, LSTM	YouTube comments (Gulf, Egyptian, Iraqi); Tweets	Mixed Arabic dialects; EN translations	BERT EN Acc 98%; AraBERT Acc 96%; mBERT-AR Acc 83%; LSTM Acc 82%	Translation pipeline may inflate EN results; sarcasm remains challenging
30	Aljarah et al. (2021)	SVM, NB, DT, RF; feature sets (TF-IDF, profile, emotion)	Twitter	General Arabic (varied topics)	RF best: Acc/G-mean 0.910; Recall 0.923; Precision 0.902 with all features	Small corpus after filtering; two-annotator protocol; neutrals excluded from training
31	Mouheb et al. (2019)	Naïve Bayes	Twitter + YouTube	General Arabic	Accuracy 0.959	Small dataset; limited feature diversity
32	Alakrot et al. (2021)	LR, SVM/LinearSVC, NB, DT, RF; POS + n-grams; feature selection	YouTube comments	Mixed dialects (YouTube)	LinearSVC highest accuracy (reasonable overall); gains from feature selection	Focus on offensive, not CB; dependence on preprocessing choices
33	Omar et al. (2021)	LinearSVC, NB variants, SVM, LR, DT, SGD, RF; multilabel pipeline	OSN posts across 11 classes; vulgar-speech set	General Arabic (Facebook/Twitter)	With Chi-square FS: Acc 97.92%; F1 97.92%; Precision 97.92%; Recall 97.93%; multilabel LinearSVC + TF-IDF Acc 82.29%, F1 92.48%	High feature counts; results sensitive to FS; generalizability outside OSN mix not shown

(Continued)

TABLE 2 (Continued)

No	Study	Model(s)	Dataset and Platform	Dialect/ Domain	Performance Metrics	Limitations
34	Shannaq et al. (2022)	Word-embedding fine-tuning + GA-optimized SVM/XGBoost	ArCybC (CB/Non-CB/Off/Non-Off)	Twitter; cyberbullying + offensive	SVM Acc 86.5% → 87.5%; XGB Acc 84.9% → 85.2% after optimization	Incremental gains; relies on a single public corpus
35	Kanan et al. (2021)	Unsupervised K-Means vs. EM (clustering)	(Facebook/Twitter)	General Arabic	Evaluated via training time, SSE (e.g., 7,796.363), and log-likelihood (e.g., 3,606.4669)	No precision/recall/F1; clustering quality hard to align with downstream moderation needs

TABLE 3 Examples of the datasets addressing cyberbullying in Arabic.

Dataset (year)	Platform	Labels	Study
Instagram-Based Benchmark Dataset for Cyberbullying in Arabic (2022)	Instagram	Comments collected; multi-class sub-categories for bullying with sentiment variants used in evaluation (incl. Positive/negative/neutral)	Albayari and Abdallah (2022)
ArCybC / ArCyC—Arabic Cyberbullying Corpus (2022 article; 2023 data release)	Twitter (X)	Tweets; dual annotation tasks: CB vs. non-CB and Offensive vs. non-Offensive; 5 annotators	Shannag et al. (2022)
ArbCyD—Arabic Post Dataset for Cyberbullying Detection (2024)	Twitter (X)	Posts: bullying vs. non-bullying binary labels	Aljalaoud et al. (2025)

6.2.1 Development and use of cyberbullying datasets

Arabic cyberbullying detection relies heavily on curated datasets. Studies often use platform-specific datasets from Twitter, YouTube, and Facebook, with datasets labeled for harmful or offensive language (e.g., Bashir and Bouguessa, 2021; Khairy et al., 2023). These datasets include common cyberbullying characteristics like threats, insults, and hate speech. However, the issue of dataset imbalance (more non-cyberbullying content than cyberbullying) persists, affecting model performance. Techniques like oversampling and downsampling have been employed to address this imbalance, as seen in AlFarah et al. (2022). Table 3. Shows some examples of the existing datasets addressing cyberbullying in Arabic.

The ArCybC/ArCyC corpus represents one of the few openly accessible multi-dialect Twitter datasets that makes a clear distinction between cyberbullying and general offensive content. Its development is supported by detailed documentation of the annotation pipeline and guidelines, ensuring methodological transparency (Shannag et al., 2022). The ArbCyD dataset significantly expands the available volume by including annotated Twitter posts (Aljalaoud et al., 2025).

6.2.2 Standards and evaluation metrics

Standards such as precision, recall, F1-score, and accuracy are commonly used to evaluate detection methods (e.g., Haidar et al., 2017; Alakrot et al., 2018). Although precision and recall are essential for accurate detection, the unique characteristics of the Arabic language and cyberbullying-specific terms often require additional metrics and customized standards. Studies such as El-Alami et al. (2022) and Amer Hamzah and Dhannoon (2023) advocate for using contextual features like sentiment polarity, emojis, and user history in cyberbullying detection. These standards help capture the nuanced characteristics of online abuse, especially within specific platforms or dialects.

Some evaluations adopt three-way labeling schemes that distinguish bullying/abusive content, non-bullying content, and neutral content. When overall accuracy is computed across all classes, the typically high prevalence of neutral instances can inflate the metric and obscure a system's effectiveness on the bullying class, which is the primary target in safety-critical applications. For example, the Instagram-based Arabic cyberbullying benchmark provides a multi-class design with positive (bullying), negative (non-bullying), and neutral categories, together with inter-annotator agreement reporting and baseline models (Albayari and Abdallah, 2022). In such settings, macro-F1 and per-class F1 are preferable for comparing systems intended to detect bullying, whereas accuracy across all three classes can be misleading when neutral content dominates the distribution.

6.2.3 Application of linguistic and psychological standards

Recent research has incorporated psychological theories to enhance cyberbullying detection by analyzing underlying personality traits in text (e.g., Elzayady et al., 2023). Such frameworks align detection methods with broader behavioral standards, moving toward a more human-centered approach in identifying abusive content. Other studies, such as Boulouard et al. (2022), address multilingual standards by analyzing Arabic text in translation and leveraging cross-linguistic BERT models, thus ensuring consistency in detecting cyberbullying characteristics across languages.

6.3 Research question 3

The third RQ was:

What future research directions in cyberbullying detection may be established based on the findings of the provided systematic review?

The following themes were developed to answer the third research question.

6.3.1 Expansion of dialect-specific datasets and multilingual analysis

Future research could focus on developing larger, dialect-specific datasets to address the significant linguistic diversity in Arabic. Datasets for Moroccan, Syrian, and Gulf dialects remain limited and would improve detection accuracy for specific regions (e.g., Rachidi et al., 2023; Ali and Kurdy, 2022). Studies also suggest expanding multilingual capabilities to improve cross-linguistic performance, with transformer models like BERT and mBERT showing potential for multilingual hate speech analysis (e.g., Alrashidi et al., 2023; Shannaq et al., 2022).

For limited-resource settings, few strategies with large language models can be grounded in complementary lines of evidence. First, in-context learning has been shown to deliver strong few-shot performance without gradient updates; GPT-3's original study established that scaling enables task-agnostic adaptation via a handful of exemplars embedded in the prompt, a result that has shaped subsequent methodology for low-data regimes (Brown et al., 2020). Second, prompt-based and prompt-free fine-tuning methods consistently improve over naïve fine-tuning when labeled data are scarce. Pattern-Exploiting Training and its generative extension reframe supervision as cloze-style patterns to amplify supervision from very small datasets, while LM-BFF automates prompt construction and demonstration selection to yield large gains across classification and regression tasks (Schick and Schütze, 2020). Complementing these, SetFit avoids handcrafted prompts altogether by contrastively fine-tuning sentence-transformer encoders on a handful of pairs and then training a lightweight classifier on the induced embeddings, matching or surpassing larger fully fine-tuned models under strict few-shot budgets (Tunstall et al., 2022). Moreover, parameter-efficient adaptation techniques such as LoRA reduce trainable parameters by orders of magnitude while preserving or improving downstream quality, which is particularly attractive when domain transfer must be achieved under tight compute and annotation constraints (Hu et al., 2022). To mitigate the scarcity of human-written instructions, Self-Instruct bootstraps synthetic instruction–input–output triplets from the model itself and shows substantial gains over the base model, offering a practical path when labeled data are limited (Wang et al., 2022). Evidence from multilingual and domain-specific studies indicates that these approaches translate beyond English benchmarks. Cross-lingual in-context learning studies report consistent benefits for genuinely low-resource languages and highlight alignment techniques that stabilize label semantics across languages, while evaluations in biomedical and clinical NLP show that instruction-tuned LLMs can perform competitively on few-shot entity recognition, QA, and relation extraction when carefully prompted (Cahyawijaya et al., 2024).

6.3.2 Enhanced deep learning models and feature engineering

Future research could involve advancing feature engineering, particularly through contextual embeddings, attention mechanisms, and personality inference models. These methods could enhance the interpretability of cyberbullying detection systems and better capture contextual aspects of offensive language (e.g., Mohaouchane et al., 2019; Elzayady et al., 2023). Additionally, hybrid models combining CNN, RNN, and BERT-based architectures have shown promise for handling complex language features, and future studies could explore further model fusion or ensemble methods for improved accuracy (e.g., Mohaouchane et al., 2019; Althobaiti, 2022).

6.3.3 Ethical considerations and real-time detection systems

Ethical standards and privacy concerns will play a growing role in future cyberbullying detection research. Privacy-preserving algorithms, especially those that anonymize or filter sensitive information, can support ethical AI use on social media platforms (e.g., Omar et al., 2021). Another area for future exploration is real-time cyberbullying detection systems that respond dynamically to harmful content as it is posted. While challenging, real-time models could be feasible with lightweight DL architectures tailored for social media monitoring (e.g., Amer Hamzah and Dhannoon, 2023; Kanan et al., 2021).

Ethical risks arise at each stage of dataset development and deployment for Arabic cyberbullying detection, beginning with data collection. The Instagram-based benchmark demonstrates the value of reporting annotation protocols and inter-annotator agreement alongside careful corpus descriptions; however, as with Twitter- and YouTube-based datasets, the presence of user mentions and cross-post threads can inadvertently expose targets and perpetrators if not aggressively sanitized (Albayari and Abdallah, 2022; Haidar et al., 2019; Alakrot et al., 2018; Alduailaj et al., 2023; Al-Ajlan and Ykhlef, 2018; Alrougi et al., 2024). Representativeness is a second, persistent ethical and scientific concern. Arabic social media is heterogeneous across dialects, platforms, and communities; yet several widely used datasets skew toward particular dialect clusters or platform norms, such as Egyptian or Gulf Twitter, pan-Arab YouTube comments, or Instagram captions from specific demographic groups (Haidar et al., 2019). Studies that publish clear guidelines, show label distributions, and report inter-annotator agreement support more accountable modeling than those that provide only aggregate scores (Albayari and Abdallah, 2022). Curators should also protect annotator wellbeing through workload limits, content warnings, and access to support, and they should state these safeguards in their documentation. The evaluation protocol has ethical implications because metric choice shapes decision thresholds used in practice. Practical architectures therefore favor lightweight normalizers and dialect-aware tokenization before model inference, with privacy-preserving logging that stores only hashed text fingerprints or short-lived embeddings for auditing (Alakrot et al., 2018). The more explicit dataset papers are about these elements, the less likely it is that downstream models will inadvertently encode representational harms or privacy leakage.

6.3.4 Integration of psychological and social dimensions

Integrating psychological and social analysis within detection algorithms is emerging as an essential direction. Personality-based approaches could be particularly useful, helping identify users more likely to engage in or be affected by cyberbullying (e.g., Elzayady et al., 2023).

Additionally, cross-disciplinary research involving psychology, sociology, and computational linguistics could establish standards for understanding the social dynamics underlying cyberbullying, offering

insights beyond linguistic patterns (e.g., Omar et al., 2021). Table 4 shows the summary of the themes related to each research question.

The results of the research emphasize the necessity of culturally sensitive detection models, sophisticated methodologies, and tailored approaches to effectively capture the distinctive characteristics of the Arabic offensive language. Arabic is an extremely diverse language, with significant variations in dialects across regions (e.g., Egyptian, Gulf, Levantine), each with its own vocabulary, syntax, and expressions. The detection of objectionable language is further complicated by this diversity, as models that have been trained in Modern Standard Arabic frequently encounter difficulties with dialectal content. These results suggest that the model's ability to identify nuanced or implicit forms of offensive language, such as sarcasm or mockery, is improved by the inclusion of sentiment and lexicon-based features that are specifically designed for Arabic dialects and slang. Many categories of offensive language, including religious hate speech, ethnic hate, and political offence, have been classified by researchers. These types of language are particularly sensitive in Arabic-speaking societies. These categories are indicative of regional and cultural priorities, emphasizing the social and religious values that influence online discourse in Arabic contexts. The importance of accounting for these categories is underscored by research, as they pertain to highly sensitive subjects that may vary in severity and context in comparison to other languages. The results indicate that culturally aware models that identify these particular forms of objectionable language can improve the accuracy and relevance of the models.

Although numerous studies have examined cyberbullying detection methods broadly or across various languages, there is a paucity of focused analyses on Arabic-language detection, given the unique challenges presented by Arabic's morphological intricacies and dialectal diversity (Mubarak and Darwish, 2019; AbdelHamid et al., 2022). The majority of the earlier studies predominantly analyze general patterns in cyberbullying detection, concentrating on English-language research (Alakrot et al., 2018; Bashir and Bouguessa, 2021). Although current studies recognize dataset imbalances and biases in social media-derived training data, they frequently neglect to consider privacy concerns and the ethical ramifications of automated cyberbullying detection among Arabic-speaking groups (Omar et al., 2021; Amer Hamzah and Dhannoon, 2023). This study addresses real-time detection concerns, the balance between moderation and free speech, and the necessity for privacy-preserving machine learning algorithms in social media monitoring (Kanan et al., 2021). This paper distinctly focuses on the thorough assessment of ML and DL models in detecting cyberbullying in Arabic. The prior systematic literature review by Castaño-Pulgarín et al. (2021), addressed cyberbullying detection on studies that provided exploratory data about the Internet and social media as a space for online hate speech, types of cyberhate, terrorism as an online hate trigger, online hate expressions and the most common methods to assess online hate speech. Balakrisnan and Kaity (2023) also did an SLR focusing on three main areas regarding cyberbullying detection through machine learning: the algorithms employed, the features used for detection, and the performance measures of these methods. The prior studies and reviews neglect Arabic-specific issues such as root-based word creation, tokenization complexities, and script intricacies.

The results of this study underscore the necessity of creating extensive, dialect-specific datasets and enhancing NLP models to address syntactic and lexical discrepancies among Arabic dialects. Deep learning architectures such as CNNs and BiLSTMs generally surpass classical baselines once training sets exceed the low-thousands and when preprocessed to handle orthographic variation, elongation, and code-mixing. Transformer models fine-tuned on Arabic corpora—especially variants trained with substantial dialectal coverage—consistently lead when the label definitions align with the pretraining distribution and when macro-averaged F1 rather than accuracy guides optimization. A recurring empirical pattern is precision outpacing recall, reflecting systems that confidently flag explicit bullying but struggle with implicit attacks, sarcasm, and context-dependent harassment. Performance differences are driven first by data composition. Dialectal diversity, platform genre, and class design are the most decisive factors. Models trained on tweets in Egyptian or Gulf dialects tend to degrade on Levantine, Maghrebi, or code-mixed content because lexical cues and morphological patterns shift, and subword tokenizers learned on Modern Standard Arabic under-segment dialectal forms. Domain shift between platforms—short, slang-heavy tweets versus longer Instagram captions or YouTube comments—likewise reduces transfer, as does the prevalence of emojis, creative spellings, and Arabizi. Class definitions also vary: some corpora equate cyberbullying with general abuse or toxicity, whereas others require intent, repetition, or power imbalance. The broader the "bullying" label, the higher the apparent scores, but the weaker the comparability across studies. Evaluation choices amplify these effects. Where annotation guidelines were explicit and inter-annotator agreement documented, models learned more stable decision boundaries; where guidelines were minimal or borrowed from sentiment analysis, models overfit to superficial polarity and miss community-specific bullying norms. Pretraining and representation learning explain the remaining variance. Yet, when fine-tuning data are severely imbalanced, even strong encoders prioritize surface toxicity over nuanced bullying constructs. In contrast, classical models augmented with curated lexicons and character-level features sometimes outperform deep baselines on noisy, low-resource dialects because they are less sensitive to tokenization errors and require fewer examples to generalize.

The most promising methodological direction is dialect- and domain-robust modeling anchored in standardized evaluation. Progress depends on a benchmark suite that harmonizes label schemas for cyberbullying versus general abuse, publishes class priors, and mandates macro-F1 and per-class F1 with clear treatment of the neutral class. Cross-dataset testing should be routine, with models trained on one corpus evaluated zero-shot on another to measure real-world robustness. Data and supervision strategies also offer leverage. Active learning and disagreement-focused annotation can densify minority bullying phenomena such as threats, doxxing, or body-shaming. Weak supervision that combines lexicon rules, community guidelines, and pattern matchers can cheaply label large pools for pretraining, followed by human verification on hard examples. Span-level rationales and multi-label tags for bullying types improve transparency and enable error analysis beyond single-label outcomes, while adversarial training with paraphrases and sarcasm

TABLE 4 Summary of the themes related to each research question.

Research Question	Theme	Description	Sources
RQ1: Current trends in cyberbullying detection for Arabic language and dialects	ML and DL Approaches	ML models (e.g., SVM, Naïve Bayes) and DL models (e.g., CNN, BERT) are common for cyberbullying detection, with ensemble methods improving accuracy.	Haidar et al. (2017); Alakrot et al. (2018); Alrashidi et al. (2023)
	Sentiment Analysis and Lexicon-Based Methods	Sentiment analysis and lexicon-based approaches capture emotional tones and harmful language, essential for handling Arabic's diverse dialects.	AlHarbi et al. (2019); Farid and El-Tazi (2020)
	Handling Arabic Dialects and Complexity	Specialized datasets and models (e.g., AraBERT, multilingual BERT) address dialectal variability, enhancing model accuracy for Arabic.	Mubarak and Darwish (2019); AbdelHamid et al. (2022); Khezzar et al. (2023)
RQ2: Standards used for detecting cyberbullying based on its characteristics	Development of Cyberbullying Datasets	Creation of Arabic-specific datasets that include dialectical variations and cyberbullying characteristics, though issues like imbalanced datasets (few cyberbullying instances) impact model performance.	Bashir and Bouguessa (2021); Khairy et al. (2023); AbdelHamid et al. (2022)
	Evaluation Standards and Metrics	Precision, recall, F1-score, and accuracy are commonly used metrics, supplemented by specialized metrics tailored to Arabic-language characteristics to ensure reliable detection performance.	Haidar et al. (2017); Alakrot et al. (2021); Boulouard et al. (2022)
	Linguistic and Psychological Standards	Integration of linguistic and psychological insights, such as personality inference, allows a deeper understanding of user behavior, helping to identify cyberbullying based on more human-centered behavioral traits.	Elzayady et al. (2023); Omar et al. (2021); Shannaq et al. (2022)
	Contextual and Cultural Considerations	Incorporation of cultural sensitivity, including the use of dialect-specific language features, emojis, and contextual sentiment, provides a more nuanced and culturally accurate detection of offensive language.	AlHarbi et al. (2019); Farid and El-Tazi (2020); Khezzar et al. (2023)
RQ3: Future research directions for Arabic cyberbullying detection	Dialect-Specific Datasets and Multilingual Models	Expansion of dialect-specific datasets and multilingual models to enhance detection across Arabic dialects and improve cross-linguistic applicability.	Ali and Kurdy (2022); Rachidi et al. (2023); Shannaq et al. (2022)
	Advanced Feature Engineering and Hybrid Models	Development of hybrid models (e.g., CNN-LSTM-BERT) and advanced feature engineering, such as attention mechanisms and personality-based features, for richer context and improved detection accuracy.	Mouheb et al. (2019); Elzayady et al. (2023); Boulouard et al. (2022)
	Real-Time Detection and Privacy Considerations	Focus on real-time cyberbullying detection models for immediate response, with privacy-preserving techniques to ensure user data protection and ethical AI application.	Amer Hamzah and Dhannoon (2023); Omar et al. (2021); Kanan et al. (2021)
	Cross-Disciplinary Research	Integration of psychological, sociological, and linguistic insights for a more comprehensive understanding of the social and behavioral dynamics underlying Arabic cyberbullying.	Farid and El-Tazi (2020); Omar et al. (2021); Elzayady et al. (2023)

transformations hardens models against implicit aggression. Context modeling is a further frontier. Many failures stem from sentence-level isolation. Incorporating conversation threads, author–target history, and lightweight social signals can disambiguate teasing from harassment and detect repetition, a hallmark of bullying. Graph-based representations of interactions, when coupled with privacy-preserving design and strict ethical safeguards, can capture power asymmetries and coordinated attacks without storing sensitive personal attributes.

Finally, instruction-tuned large language models adapted to Arabic show potential as few-shot labelers, error analyzers, and data generators, but their deployment must be paired with rigorous calibration, bias auditing across dialects and demographics, and conservative thresholding in safety-critical pipelines. Taken together, the evidence suggests that the field is moving from accuracy on single, homogeneous datasets toward robust, dialect-inclusive systems evaluated under standardized, recall-sensitive protocols, with the integration of context and improved supervision likely to yield the next substantive gains.

7 Limitations and suggestions for future studies

A key limitation of this review is the absence of a formal quality appraisal or risk-of-bias assessment of the included studies. Established tools such as AMSTAR, AMSTAR-2, or ROBIS are often used in systematic reviews to evaluate the methodological rigor of primary studies and to distinguish between stronger and weaker evidence. The present review treats all included studies as methodologically equivalent, regardless of variations in their design, sampling strategies, or analytical robustness.

The majority of the studies reviewed are based on restricted or specific datasets, which may not adequately represent the complete range of Arabic dialectal diversity or the diverse forms of cyberbullying that are present on different platforms. However, the absence of standardized datasets for the detection of Arabic cyberbullying also presents obstacles to the attainment of generalizable results. Despite the potential of dialect-specific models, the complexity and extensive variations among Arabic dialects pose a significant obstacle. The results may not be broadly applicable because current models may not perform consistently across all dialects. The detection of real-time cyberbullying is still in its infancy, particularly in the context of Arabic texts. Although some studies incorporate psychological insights, there is a void in the comprehensive integration of insights from sociology, linguistics, and psychology to develop a holistic understanding of cyberbullying behaviors specific to Arabic-speaking regions. Another limitation of this review is the exclusion of conference proceedings, despite their prominence as venues for innovation in natural language processing. Nonetheless, this exclusion may have led to the omission of some cutting-edge contributions. Future reviews should consider incorporating both journal articles and high-quality conference proceedings to present a more comprehensive view of the research landscape.

Future research may investigate sophisticated deep learning architectures and hybrid models that amalgamate various methodologies to enhance detection, to improve contextual comprehension and classification precision. Another vital avenue for future study is the enhancement of sentiment-based and context-aware models for detecting cyberbullying. The problem of dataset imbalance persists, as cases of cyberbullying are markedly underrepresented relative to non-offensive content.

8 Conclusion

This study offers a thorough examination of the most recent academic research, methodologies, and challenges in the detection of cyberbullying in Arabic texts. This review emphasizes the substantial advancements that have been achieved in this field by evaluating the efficacy of ML and DL models, sentiment analysis, lexicon-based methods, and dialectal considerations. The significance of specialized datasets for Arabic dialects, the efficacy of composite models and ensemble learning, and the value of sentiment-based and contextual analysis are underscored by the key findings.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

Author contributions

HA: Conceptualization, Data curation, Investigation, Methodology, Project administration, Software, Visualization, Writing – original draft, Writing – review & editing. MA: Investigation, Methodology, Project administration, Supervision, Validation, Writing – review & editing. SM: Methodology, Project administration, Supervision, Validation, Writing – review & editing. AB: Project administration, Software, Supervision, Validation, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-SAFIR-2025".

Acknowledgments

The authors would like to thank their academic peers and institutional colleagues for their feedback during the early stages of this research.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

AbdelHamid, M., Jafar, A., and Rahal, Y. (2022). Levantine hate speech detection in twitter. *Soc. Netw. Anal. Min.* 12:121. doi: 10.1007/s13278-022-00950-4

Abdelmonem, A. (2015). Reconceptualizing sexual harassment in Egypt: a longitudinal assessment of el-Taharrush el-Ginsy in Arabic online forums and anti-sexual harassment activism. *Kohl: J. Body Gender Res.* 1, 23–41. doi: 10.36583/kohl/1-1/

Abu Farha, I. (2023). Arabic sarcasm detection.

Al, Z. N. (2019). Divine impoliteness: how Arabs negotiate Islamic moral order on twitter. *Russ. J. Linguist.* 23, 1039–1064.

Alakrot, A., Fraifer, M., and Nikolov, N. S. (2021). "Machine learning approach to detection of offensive language in online communication in Arabic." in *2021 IEEE 1st international Maghreb meeting of the conference on sciences and techniques of automatic control and computer engineering MI-STA*, pp. 244–249.

Alakrot, A., Murray, L., and Nikolov, N. S. (2018). Towards accurate detection of offensive language in online communication in Arabic. *Proc. Comput. Sci.* 142, 315–320. doi: 10.1016/j.procs.2018.10.491

Albayari, R., and Abdallah, S. (2022). Instagram-based benchmark dataset for cyberbullying detection in Arabic text. *Data* 7:83. doi: 10.3390/data7070083

AlFarah, M. E., Kamel, I., Al Aghbari, Z., and Mouheb, D. (2022). "Arabic cyberbullying detection from imbalanced dataset using machine learning" in Soft computing and its engineering applications. eds. K. K. Patel, G. Doctor, A. Patel and P. Lingras, vol. *1572* (Changa, Anand, India: Springer International Publishing), 397–409. doi: 10.1007/978-3-031-05767-0_31

AlHarbi, B. Y., AlHarbi, M. S., AlZahrani, N. J., Alsheail, M. M., Alshobaili, J. F., and Ibrahim, D. M. (2019). *Automatic cyber bullying detection in Arabic social media*. 12(12).

Al-Hassan, A., and Al-Dossari, H. (2022). Detection of hate speech in Arabic tweets using deep learning. *Multimedia Systems* 28, 1963–1974. doi: 10.1007/s00530-020-00742-w

Al-Ajlan, M. A., and Ykhlef, M. (2018). Optimized Twitter cyberbullying detection based on deep learning. In *Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC)*. 1–5. IEEE. doi: 10.1109/NCG.2018.8593146

Alduailaj, A. M., and Belghith, A. (2023). Detecting Arabic cyberbullying tweets using machine learning. *Mach. Learn. Knowl. Extr.* 5, 29–42. doi: 10.3390/make5010003

Alhashmi, A. A., and Darem, A. A. (2022). Consensus-based ensemble model for Arabic cyberbullying detection. *Computer Systems Science and Engineering*, 41, 241–254. doi: 10.32604/csse.2022.020023

Ali, R., and Kurdy, D. M.-B. (2022). Cyberbullying detection in Syrian slang on social media by using data mining. *Int. J. Eng. Res.* 11.

Aljalaoud, H., Dashtipour, K., and AI Dubai, A. (2025). Arabic cyberbullying detection: a comprehensive review of datasets and methodologies. *IEEE Access*. doi: 10.1109/ACCESS.2025.3561132

Aljarah, I., Habib, M., Hijazi, N., Faris, H., Qaddoura, R., Hammo, B., et al (2021). Intelligent detection of hate speech in Arabic social network: A machine learning approach. *J. Inf. Sci.* 47, 483–501. doi: 10.1177/0165551520917651

Aljuhani, O., Alyoubi, K., and Alotaibi, F. (2022). Detecting Arabic offensive language in microblogs using domain-specific word Embeddings and deep learning. *Tehnički Glasnik* 16, 394–400. doi: 10.31803/tg-20220305120018

Alrashidi, B., Jamal, A., and Alkhathlan, A. (2023). Abusive content detection in Arabic tweets using multi-task learning and transformer-based models. *Appl. Sci.* 13:5825. doi: 10.3390/app13105825

Alrougi, M., Alamoudi, G., and Algamdi, H. (2024). ArbCyD: An Arabic post dataset for cyberbullying detection. *J. Electr. Syst.* 20, 1583–1589.

Alsafari, S., Sadaoui, S., and Mouhoub, M. (2020a). "Deep Learning Ensembles for Hate Speech Detection." in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 526–531.

Alsafari, S., Sadaoui, S., and Mouhoub, M. (2020b). Hate and offensive speech detection on Arabic social media. *Online Soc. Networks Media* 19:100096. doi: 10.1016/j.osnem.2020.10009

Alshalabi, N., Lahiani, H., and Yasin, A. (2024). The role of culture in abusive language on social media: examining the use of English and Arabic derogatory terms. *Theory Pract. Lang. Stud.* 14, 3057–3066. doi: 10.17507/tpls.1410.06

Alsubait, T., and Alfageh, D. (2021). Comparison of machine learning techniques for cyberbullying detection on YouTube Arabic comments. *Int. J. Comput. Sci. Netw. Secur.* 21, 1–5. doi: 10.22937/IJCSNS.2021.21.1.1

Althobaiti, M. J. (2022). BERT-based approach to Arabic hate speech and offensive language detection in twitter: exploiting emojis and sentiment analysis. *Int. J. Adv. Comput. Sci. Appl.* 13:5109. doi: 10.14569/IJACSA.2022.01305109

Amer Hamzah, N., and Dhannoon, B. N. (2023). Detecting Arabic sexual harassment using bidirectional long-short-term memory and a temporal convolutional network. *Egypt. Inform. J.* 24, 365–373. doi: 10.1016/j.eij.2023.05.007

Anezi, F. Y. A. (2022). Arabic hate speech detection using deep recurrent neural networks. *Appl. Sci.* 12:6010. doi: 10.3390/app12126010

Balakrisnan, V., and Kaity, M. (2023). Cyberbullying detection and machine learning: a systematic literature review. *Artif. Intell. Rev.* 56, 1375–1416. doi: 10.1007/s10462-023-10553-w

Bashir, E., and Bouguessa, M. (2021). Data mining for cyberbullying and harassment detection in Arabic texts. *Int. J. Inform. Technol. Comp. Sci.* 13, 41–50. doi: 10.5815/ijitcs.2021.05.04

Bertini, F., Allevi, D., Lutero, G., Montesi, D., and Calzà, L. (2021). Automatic speech classifier for mild cognitive impairment and early dementia. *ACM Trans. Comp. Healthcare* 3, 1–11. doi: 10.1145/3469089

Bouhlila, D. S. (2019). Sexual harassment and domestic violence in the Middle East and North Africa. Arab Barometer, 2.

Bouliche, A., and Rezoug, A. (2022). Detection of cyberbullying in Arabic social media using dynamic graph neural network. In *Proceedings of the 1st Tunisian-Algerian Joint Conference on Applied Computing (TACC 2022)*. 1–11.

Boulouard, Z., Ouaissa, M., Ouaissa, M., Krichen, M., Almutiq, M., and Gasmi, K. (2022). Detecting hateful and offensive speech in Arabic social media using transfer learning. *Appl. Sci.* 12:12823. doi: 10.3390/app122412823

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Adv. Neural Inf. Proces. Syst.* 33, 1877–1901.

Cahyawijaya, S., Lovenia, H., and Fung, P. (2024). Llms are few-shot in-context low-resource language learners. *arXiv preprint arXiv*:2403.16512.

Castaño-Pulgarín, S. A., Suárez-Betancur, N., Vega, L. M. T., and López, H. M. H. (2021). Internet, social media and online hate speech. Systematic review. *Aggress. Violent Behav.* 58:101608. doi: 10.1016/j.avb.2021.101608

Cowie, J., and Lehnert, W. (1996). Information extraction. *Commun. ACM* 39, 80–91. doi: 10.1145/234173.234209

El-Alami, F., Ouatik El Alaoui, S., and En Nahnahi, N. (2022). A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model. *J. King Saud Univ. - Comput. Inf. Sci.* 34, 6048–6056. doi: 10.1016/j.jksuci.2021.07.013

Elzayady, H., Mohamed, M. S., Badran, K. M., and Salama, G. I. (2023). A hybrid approach based on personality traits for hate speech detection in Arabic social media. *Int. J. Elect. Comp. Eng.* 13:1979–88. doi: 10.11591/ijece.v13i2.pp1979-1988

Farid, D., and El-Tazi, D. N. (2020). Detection of cyberbullying in tweets in Egyptian dialects. 18(7).

Fati, S. M. (2022). Detecting cyberbullying across social media platforms in Saudi Arabia using sentiment analysis: A case study. *Comput. J.* 65, 1787–1794. doi: 10.1093/comjnl/bxab019

Grégoire, Y., Salle, A., and Tripp, T. M. (2015). Managing social media crises with your customers: the good, the bad, and the ugly. *Bus. Horiz.* 58, 173–182. doi: 10.1016/j.bushor.2014.11.001

Haidar, B., Chamoun, M., and Serhrouchni, A. (2017). A multilingual system for cyberbullying detection: Arabic content detection using machine learning. *Adv. Sci. Technol. Eng. Syst. J.* 2, 275–284. doi: 10.25046/aj020634

Haidar, B., Chamoun, M., and Serhrouchni, A. (2019). Arabic cyberbullying detection: Enhancing performance by using ensemble machine learning. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 323–327. IEEE. doi: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00074

Haidar, B., Chamoun, M., and Serhrouchni, A (2018). "Arabic cyberbullying detection: Using deep learning." in 7th International Conference on Computer and Communication Engineering (ICCCE), IEEE. pp. 284–289.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., et al. (2022). Lora: low-rank adaptation of large language models. *ICLR* 1:3.

Kanan, T., Aldaaja, A., and Hawashin, B. (2020). Cyber-bullying and cyber-harassment detection using supervised machine learning techniques in Arabic social media contents. *J. Internet Technol.* 21, 1409–1421.

Kanan, T., Kanaan, G. G., Al-Shalabi, R., and Aldaaja, A. (2021). Offensive language detection in social networks for Arabic language using clustering techniques.

Khairy, M., Mahmoud, T. M., Omar, A., and Abd El-Hafeez, T. (2023). Comparative performance of ensemble machine learning for Arabic cyberbullying and offensive language detection. *Lang. Resour. Eval.* 58, 695–712. doi: 10.1007/s10579-023-09683-y

Khezzar, R., Moursi, A., and Al Aghbari, Z. (2023). ArHatedetector: detection of hate speech from standard and dialectal Arabic tweets. *Discov. Internet Things* 3:1. doi: 10.1007/s43926-023-00030-9

Mirończuk, M. M., and Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Syst. Appl.* 106, 36–54. doi: 10.1016/j.eswa.2018.03.058

Mohaouchane, H., Mourhir, A., and Nikolov, N. S. (2019). "Detecting Offensive Language on Arabic Social Media Using Deep Learning." in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pp. 466–471.

Mouheb, D., Albarghash, R., Mowakeh, M. F., Aghbari, Z. A., and Kamel, I. (2019). Detection of Arabic cyberbullying on social networks using machine learning.

Mubarak, H., and Darwish, K. (2019). "Arabic offensive language classification on twitter" in Social informatics. eds. I. Weber, K. M. Darwish, C. Wagner, E. Zagheni, L. Nelson and S. Arefet al., vol. *11864* (Doha, Qatar: Springer International Publishing), 269–276.

Niraula, N. B., Dulal, S., and Koirala, D. (2021). "Offensive Language Detection in Nepali Social Media." in *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*. pp. 67–75.

Omar, A., Mahmoud, T. M., Abd-El-Hafeez, T., and Mahfouz, A. (2021). Multi-label Arabic text classification in online social networks. *Inf. Syst.* 100:101785. doi: 10.1016/j.is.2021.101785

Rachidi, R., Ouassil, M. A., Errami, M., Cherradi, B., Hamida, S., and Silkan, H. (2023). Classifying toxicity in the Arabic Moroccan dialect on Instagram: a machine and deep learning approach. *Indones. J. Electr. Eng. Comput. Sci.* 31:588. doi: 10.11591/ijeecs.v31.i1.pp588-598

Rosenbaum, G. M. (2019). Curses, insults and taboo words in Egyptian Arabic: in daily speech and in written literature. *Romano-Arabica* 19, 153–188.

Sap, M., Card, D., Gabriel, S., Choi, Y., and Smith, A. N. (2019). The risk of racial bias in hate speech detection. ACL.

Schick, T., and Schütze, H. (2020). Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv*:2001.07676.

Shannag, F., Hammo, B. H., and Faris, H. (2022). The design, construction and evaluation of annotated Arabic cyberbullying corpus. *Educ. Inf. Technol.* 27, 10977–11023. doi: 10.1007/s10639-022-11056-x

Shannaq, F., Hammo, B., Faris, H., and Castillo-Valdivieso, P. A. (2022). Offensive language detection in Arabic social networks using evolutionary-based classifiers learned from fine-tuned embeddings. *IEEE Access* 10, 75018–75039. doi: 10.1109/ACCESS.2022.3190960

Tunstall, L., Reimers, N., Jo, U. E. S., Bates, L., Korat, D., Wasserblat, M., et al. (2022). Efficient few-shot learning without prompts. *arXiv preprint arXiv*:2209.11055.

Urrutia Zubikarai, A. (2020). Appled NLP and ML for the detection of inappropiarte text in a communications platform, Universitat Politècnica de Catalunya.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., et al. (2022). Self-instruct: aligning language models with self-generated instructions. arXiv preprint arXiv:2212.10560.

Frontiers in
Artificial Intelligence

Explores the disruptive technological revolution of AI

A nexus for research in core and applied AI areas, this journal focuses on the enormous expansion of AI into aspects of modern life such as finance, law, medicine, agriculture, and human learning.

Discover the latest Research Topics

See more →

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact