frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 17 October 2025
DOI 10.3389/fcomp.2025.1683272

EDITED BY
Mohammed Saqr,
University of Eastern Finland, Finland

REVIEWED BY
Mira Hajj Hassan,
Université Claude Bernard Lyon 1, France
Soumia Ikiss,
National Institute of Posts and
Telecommunications (INPT), Morocco

*CORRESPONDENCE
Salma Abdulbaki Mahmood
✉ salma.mahmood@uobasrah.edu.iq

RECEIVED 10 August 2025
ACCEPTED 30 September 2025
PUBLISHED 17 October 2025
CORRECTED 12 November 2025

CITATION
Mahmood SA (2025) Optimizing architectural-feature tradeoffs in Arabic automatic short answer grading: comparative analysis of fine-tuned AraBERTv2 models. *Front. Comput. Sci.* 7:1683272. doi: 10.3389/fcomp.2025.1683272

Optimizing architectural-feature tradeoffs in Arabic automatic short answer grading: comparative analysis of fine-tuned AraBERTv2 models

Salma Abdulbaki Mahmood*

Department of Computer Information Systems, Computer Science and Information Technology College, University of Basrah, Basrah, Iraq

Automated essay evaluation systems represent a contemporary solution to the challenges presented by technological advancements in education, offering high accuracy in assessment while reducing reliance on human resources. This makes them essential in light of the growing demand for fast and reliable evaluation systems. However, a critical concern remains regarding the precision of these systems in their assessments and their ability to generalize in environments where large datasets are not readily available. This research aims to examine the generalizability of Automated Short Answer Grading (ASAG) systems under different training conditions, including unannotated data and annotated data. Through a comprehensive comparative methodology, the study evaluates the performance of precisely fine-tuned AraBERTv2 models integrated with three neural network architectures: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), while testing them with varying numbers of features (2, 3, 4) using the AS-ARSG dataset. The primary goal is to explore the models' generalizability when incomplete data is available (unannotated or partially annotated) and to develop a flexible framework that reduces dependence on human assessment while maintaining grading quality. The results confirm that the two-feature MLP model outperformed all others by achieving the best performance with less error and high correlation values (MAE = 1.31, Spearman's coefficient = 0.808). In contrast, performance degradation was noted with the increasing number of features, especially in LSTM models. Through this approach, the research contributes to developing Arabic ASAG systems capable of adapting to limited data scenarios, thereby enhancing their efficiency and practical applicability.

KEYWORDS

large language model (LLMs), AraBERT, neural network, Arabic natural language processing, educational assessment, Automated Short Answer Grading (ASAG)

1 Introduction

Automated Writing Evaluation (AWE) has emerged as a more effective alternative to traditional paper-and-pencil tests, and like AWE, these systems are fast, accurate, and capable of providing objective evaluations without human interference, thus enabling immediate feedback. The potential biases of human assessment are left behind, and AWE almost supports diverse contexts of learning, including language learning and other specialized subjects, enabling myriad assessment formats, including games, simulations, and other interactive tasks. AWE generates data-rich insights that guide educators in pinpointing strengths and weaknesses in curricula, instructional approaches, and learning outcomes, thereby applying this information to improve outcomes. Assessing, reporting, and recording in AWE is designed to foster effective learning beyond the scope of classical tests and multiple-choice examinations, thus expanding the boundaries of innovative assessment practices and improving efficiency. AWE is crafted to measure depth of understanding and broad knowledge acquired by students rather than test scores alone (Yan, 2020; Zawacki-Richter and Jung, 2023).

Educational Questions used in academic assessment fall into two general categories: closed questions (e.g., multiple choice, true/false) and open questions (e.g., short-answer, essay questions). While computer programs can quickly and accurately assess closed questions, they often fail to show how well students understand the material. This is due to issues like guessing and the ease of cheating (Wilianto and Suganda Girsang, 2023). On the other hand, open questions, particularly short-answer and essay questions, are better measures of students' understanding and test their ability to express themselves, as they enable examinees to express their knowledge in their own language using sentences, linguistic structures, or even entire paragraphs (Badry et al., 2023). Analyzing open questions is challenging. Manual scoring by human experts is time-consuming, especially with a large number of students, and is more susceptible to discrepancies and inconsistencies due to the lack of clear, pre-defined assessment criteria (Lagakis and Demetriadis, 2021).

Although automated scoring systems exist for closed-ended questions, accurate scoring systems for open-ended questions are severely limited due to the difficulty of analyzing the textual content of students' answers. This is due to several challenges, including, but not limited to, lexical coverage, correct spelling, precise grammatical structures, coherence, and logical coherence of the ideas presented, as well as freedom of narration, as textual content can be presented using different synonyms, grammatical constructions, and various sentence arrangements. The complexity further increases when determining whether to adopt a comprehensive or partial analysis of the textual content, particularly considering the different types of essays, such as argumentative, response, and narrative essays, each of which requires specialized treatment and processing (Yang et al., 2020). These challenges are further complicated in Arabic due to linguistic variations, including dialectal differences, morphological variants, synonymous vocabulary, and grammatical forms (Lotfy et al., 2023).

Automated Scoring Systems (ASS) systems have undergone a progressive methodological evolution, reflecting advancements in natural language processing (NLP) and artificial intelligence (AI). In their initial phase, these systems relied on traditional rule-based approaches, which depended on manual feature engineering—extracting predefined linguistic features such as statistical metrics (e.g., sentence count, lexical diversity, paragraph length), syntactic correctness (e.g., grammatical structures and cohesive devices), and semantic keyword lists. Although this method provided satisfactory outcomes in controlled environments, it encountered essential challenges, such as biased decision-making during feature selection and rigidity toward unconventional writing styles. It intensified personnel costs, especially in the context of expansive educational programs (Mahmood and Abdulsamad, 2024; Shermis and Burstein, 2013).

The development of machine learning methods, particularly artificial neural networks, has marked a significant change in ASAG systems, as these systems now possess the ability to extract features through more sophisticated processes automatically. During this period, machine comprehension was a distinctly strengthened capability of the systems, where the models were trained on annotated datasets and improved in scoring accuracy. These systems, however, faced significant deficits due to their overfitting to the training datasets, their limited availability, and their training configurations tailored to the educational context. Consequently, these systems were rendered useless in other assessment settings (Mahmood and Abdulsamad, 2024; Jong et al., 2023). The development of large language models (LLMs) and transformers, such as BERT and GPT, has progressed significantly in recent years, which have shown significant progress and revolutionized ASAG through two key innovations: (1) contextual embeddings that capture nuanced semantic relationships within texts, and (2) fine-tuning capabilities, allowing adaptation to scoring tasks with relatively limited labeled data (Zawacki-Richter and Jung, 2023; Paaß and Giesselbach, 2023).

The primary objective of our study is to enhance assessment generalization in Arabic ASAG. To this end, we integrate AraBERTv2 embeddings with complementary neural architectures—MLP, CNN, and LSTM—that have been widely validated in prior AWE and ASAG research as effective for capturing different representational levels. Specifically, MLP provides a strong starting point for assessing generalization and performing non-linear transformations, convolution neural networks are a strong local n-gram and phrase-level feature extractor for short-answer grading, and LSTMs' infinite memory aids in the evaluation process by modeling sequential dependencies and improving system coherence and context integration. By combining these architectures with AraBERTv2, our study systematically investigates how different neural mechanisms can jointly contribute to improved performance and robust generalization.

Despite their success, applying these models to low-resource languages like Arabic presents unique research challenges, including the scarcity of annotated datasets, morphological and syntactic complexities, and the absence of standardized evaluation benchmarks for Arabic AWE systems. Filling in these gaps, the current study combines the AraBERT language model with Neural Networks Methods to optimize Arabic Automated Essay Scoring. Unlike conventional methods that evaluate student answers in isolation, the proposed model is trained on enriched data tuples, comprising the question prompt, the model (reference) answer, the student response, and the human-assigned score. This methodology enables the system to capture the nuanced relationships between questions, model answers, and actual student responses. It uses evaluation metrics to assess the proposed methods that were applied with an 80:20 train-test split. Integrating these architectural features focuses on improving scoring precision while maintaining the efficiency gains noted in more basic models.

In this context, this study tries to focus on four core research questions to advance Arabic automated scoring systems:

- To what extent can AraBERT2′s semantic representations, combined with different neural architectures (MLP/CNN/LSTM), effectively automate Arabic answer assessment when using optimized feature selection?
- How does progressive feature expansion (2 → 4 features) impact scoring accuracy differently across MLP, CNN, and LSTM architectures?
- What performance advantages emerge when combining AraBERTv2 with MLP vs. CNN or LSTM regression heads under identical feature selection conditions?
- What optimal architecture-feature combinations emerge when balancing scoring accuracy (MAE/RMSE) against evaluation consistency (Pearson/Spearman) in Arabic assessment tasks?

These research questions guided our efforts toward developing and demonstrating practical contributions that directly address these inquiries through:

- Development of an innovative hybrid architecture integrating AraBERTv2 for contextual representation with a multilayer perceptron (MLP) regression head, significantly enhancing the accuracy of automated Arabic answer scoring compared to conventional methods.
- Comprehensive comparative analysis of three neural architectures (MLP, CNN, LSTM) with varying feature sets (2, 3, 4 features), systematically demonstrating the impact of feature quantity and model complexity on scoring accuracy and reliability.
- Establishment of a standardized evaluation framework utilizing the benchmark AS-ARSG dataset, providing researchers and developers with an objective metric for comparative assessment of Arabic automated scoring models.
- Empirical validation confirms the existence of a simplicity-performance tradeoff, where the model achieves better performance by being less complex and utilizing fewer features. This method decreases the need for manually annotated datasets and improves the time efficiency of both the model's training and inference processes.
- Implementation of a practical, deployable model exhibiting high computational efficiency, making it suitable for real-world integration in educational environments.

The remainder of this research is structured in the following way: Section "Related works" addresses the related works. Section "Research Methodology" outlines the suggested framework. Section "Results and discussion" displays the experimental findings and offers a discussion. Section "Discussion" provides the conclusion and explores future directions.

2 Related works

Recent developments in the past 10 years achieved within the domain of Automated Scoring Systems (ASS) have been significantly impacted by advancements in Natural Language Processing (NLP) and deep learning. With the implementation of LLMs such as BERT and GPT, the use of prominent hand-crafted features has been replaced. This has, in turn, led to progress in the accuracy of scoring, as well as the reduction of discrepancies between human and machine evaluation. This review focuses on three main themes: (1) traditional and hybrid AWE models, (2) challenges of scoring in different languages, especially with complex features like Arabic, and (3) the role of LLMs in enhancing automated assessment. We also examine research gaps in existing studies and how the current work addresses them.

The surveyed studies primarily focus on two key objectives: (1) developing efficient scoring systems and (2) enhancing model generalizability (Condor and Litster, 2021). Notably, research examining system reliability remains limited, except for their investigation of transformer-based language models (specifically GPT-3 text-davinci-003)—a non-programmed method for Automated Scoring Systems (ASS) using the TOEFL11 corpus (Mizumoto and Eguchi, 2023). Significant efforts have been directed toward developing domain-specific scoring systems. Representative examples include Mathematics-focused scoring systems (Mengxue et al., 2022), and Sociology-oriented evaluation frameworks (Lotfy et al., 2023; Shehab et al., 2018).

Methodologically, these studies can be categorized along two dimensions. The first dimension was the feature extraction approaches, which can be classified into manual feature engineering methods, hybrid, and Integrated end-to-end learning systems. Whereas, the second is represented by scoring computation methods, which take three main directions: mathematical similarity measures (e.g., cosine similarity), machine learning approaches, and state-of-the-art large language models techniques. The following works utilize machine learning approaches that require large annotated datasets. Cozma et al. (2018) proposes a hybrid approach for automated essay scoring (AES) that uses ν-SVR (Nu-Support Vector Regression) for fusion, integrating Histogram Intersection String Kernel (HISK) as surface-level features with semantic representations derived from the Bag-of-Super-Word-Embeddings (BOSWE) model. The study in Shehab et al. (2018) aimed to develop an automated Arabic essay grading system by implementing four linguistic processing methods: two string-based algorithms (Damerau-Levenshtein and N-Gram) and two corpus-based approaches (LSA and DISCO2). The researchers in Lotfy et al. (2023) evaluated six ML algorithms (Decision Tree, Random Forest, Adaboost, Lasso, Bagging, and K-Nearest Neighbor) on a dataset of 270 sociology essays (27 questions × 10 responses. Badry et al. (2023) developed an Automatic Arabic Short Answer Grading (AASAG) system leveraging Latent Semantic Analysis (LSA) with two distinct weighting schemas (local weight vs. hybrid local/global weight) on the AR-ASAG dataset containing 2,133 answer pairs.

Many works utilize state-of-the-art large language models with different feature extraction methods. The study in Yang et al. (2020) proposes R2BERT, an enhanced automated essay scoring (AES) model that innovatively combines regression and ranking losses during fine-tuning of pre-trained BERT. Using the ASAP dataset, the Automated Student Assessment Prize dataset. The model achieved state-of-the-art performance, outperforming existing neural models by nearly 3% in average Quadratic Weighted Kappa (QWK). Beseiso and Alzahrani (2020) suggest a comprehensive empirical analysis of Automated Essay Scoring (AES) models by framing the scoring process as both rescaled regression and quantized classification problems. Utilizing the ASAP benchmark dataset, the authors systematically compared combinations of 30 manually-engineered features, 300-dimensional Word2Vec representations, and 768-dimensional BERT embeddings. Their combination of the two approaches showed promising results (77.2 ± 1.7 Kappa in the regression task and 75.2 ± 1.0% accuracy for the classification. Condor and Litster (2021) examine the generalizability of Automatic Short Answer Grading (ASAG) models to out-of-sample questions, evaluating how different model components affect performance. The research utilized a dataset of 5,550 student responses from 558 students across 33 distinct questions, comparing three text representation methods (SBERT, Word2Vec, and Bag-of-Words) and two classification models (multinomial logistic regression and a three-layer feedforward neural network). The results showed that SBERT performed the best with an accuracy of 0.621, followed by Word2Vec and Bag-of-Words with an accuracy of 0.605 and 0.575, respectively. The study of Mengxue et al. (2022) proposes a novel in-context meta-learning framework for automatic short-answer grading in mathematics, utilizing MathBERT (a mathematical domain-adapted BERT variant) fine-tuned on a cleaned dataset (Dclean) of 131,046 responses to 1,333 questions from an online learning platform. The researchers utilized a novel in-context learning method that integrates scoring examples as input for the model to improve generalization to previously unencountered questions, resulting in impressive performance metrics (AUC: 0.736, RMSE: 0.610, Kappa: 0.758.

The study of Mizumoto and Eguchi (2023) investigates the reliability and accuracy of transformer-based language models (specifically GPT-3 text-davinci-003) for Automated Essay Scoring (AES) using the TOEFL11 corpus comprising 12,100 essays from learners of 11 native languages (including Arabic), evenly distributed across three proficiency levels (Low, Medium, High). The researchers employed GPT-3 to automatically score all essays while examining the complementary role of linguistic features in enhancing scoring accuracy. Results demonstrated statistically significant differentiation between proficiency levels (effect sizes: Low-Medium $d = 1.06$; Low-High $d = 1.74$; Medium-High $d = 0.68$), confirming GPT3′s potential as a reliable AES tool with particular strength in distinguishing extreme proficiency levels. The study in Li et al. (2023) proposes an innovative Automatic Essay Scoring (AES) method incorporating multi-scale feature analysis, combining document-scale global features, sentence-scale local features (using Sentence-BERT for vectorization), manually-crafted shallow linguistic features, and prompt-relevance features. Evaluated on the Kaggle ASAP dataset, the integrated approach achieved a 79.3% Quadratic Weighted Kappa score, demonstrating significant improvement over baseline methods. The study in Wilianto and Suganda Girsang (2023) evaluates the efficacy of semantic similarity methods for automatic short answer grading in high school e-learning environments, utilizing three pre-trained sentence transformer models (all-mpnet-base-v2, all-distilroberta-v1, all-MiniLM-L6-v2) to process 840 teacher-graded student answers. The implementation employed cosine similarity for automated scoring, with all-MiniLM-L6-v2 emerging as the optimal model, demonstrating both the highest alignment with teacher-assigned grades (lowest MAE values) and computational efficiency (processing all answers in 31 s). Meccawy et al. (2023) present a comprehensive analysis of automated Arabic short answer scoring, comparing three NLP approaches (BERT embeddings, Word2Vec, and Arabic WordNet-based similarity) across two datasets: the AR-ASAG corpus (2,133 cybercrime answers) and a Jordanian History Exam dataset (550 responses). Employing rigorous text preprocessing (normalization, stemming/lemmatization) and cosine similarity measurement, the results demonstrated BERT's superior performance with the lowest RMSE (1.00308) and highest Pearson correlation (0.841902). Chamidah et al. (2023) investigate the impact of sentence tokenization on Indonesian Automated Essay Scoring (AES) using pretrained SBERT embeddings and a Siamese Manhattan LSTM (MaLSTM) architecture, analyzing 2,157 student responses across 40 questions in four domains (politics, sports, lifestyles, technology). The hierarchical approach, employing distiluse-base-multilingual-cased-v2 embeddings without sentence splitting, achieved optimal performance (RMSE: 10.65, Pearson Correlation: 0.92), demonstrating that whole-text embeddings marginally outperformed tokenized approaches (+0.61% RMSE improvement).

Faseeh et al. (2024) propose a hybrid automated essay scoring (AES) approach that integrates RoBERTa contextual embeddings with handcrafted linguistic features (grammar, readability, sentence structure) using Lightweight XGBoost (LwXGBoost) on the ASAP dataset (12,976 essays across eight genres). The model achieved state-of-the-art performance (QWK: 0.941) by effectively combining deep semantic analysis with domain-specific feature engineering, demonstrating particular robustness against noisy and sparse data. The authors of the study (Ghazawi and Simpson, 2024) introduce AR-AES, a novel benchmark dataset for Arabic Automated Essay Scoring (AES) comprising 2,046 undergraduate essays from four disciplines, annotated with dual instructor ratings, gender information, and detailed rubrics (115,454 total tokens, 12,440 unique tokens). The research pioneers the application of AraBERT for Arabic AES, demonstrating exceptional performance on environmental chemistry essays (QWK: 0.971, F1: 0.95) while establishing methodological best practices through transparent annotation protocols and quality control measures. This research in Aggarwal et al. (2025) presents EngSAF, an engineering-domain ASAG dataset with 5.8K student responses to 119 questions, which employs the novel Label-Aware Synthetic Feedback Generation (LASFG) method to enrich traditional ASAG data with multifaceted feedback. The research benchmarks multiple LLM approaches (Llama-2/3, Mistral-7B, GPT-4o, DeepSeek) in both fine-tuned and zero-shot configurations, with Mistral-7B emerging as the optimal model (75.4% accuracy on unseen answers, 58.7% on unseen questions) while maintaining high feedback quality scores (4.23/5 for unseen answers via Gemini evaluation). The study presented in Mahmoud et al. (2024) introduces a parameter-efficient framework for Arabic Automated Essay Scoring (AES) using AraBART with innovative optimization techniques, including Parameter-Efficient Fine-Tuning (PEFT), Model Soup, and Multi-Round Inference, evaluated across multiple benchmark datasets (QALB-2014, QALB-2015, ZAEBUC). The approach targets explicitly grammatical assessment while maintaining extensibility for other scoring dimensions (content similarity, organization, prompt adherence). The authors (Sun and Wang, 2024) develop a novel multi-dimensional Automated Essay Scoring (AES) system by integrating fine-tuned BERT-based classifiers (RoBERTa, DistilBERT) with multiple regression techniques, evaluated across two L2 learner corpora: ELLIPSE (9,000 essays; 5 dimensions) and IELTS (16,500 essays; 6 dimensions). The hybrid architecture combines cross-entropy classification loss with MSE regression loss through dual output heads, enhanced by contrastive learning for prompt-aware scoring, achieving consistent performance (>0.8 QWK) across all assessment dimensions.

The study in Doi et al. (2024) investigates the enhancement of Automated Essay Scoring (AES) through grammatical feature integration, employing multi-task learning (MTL) and Item Response Theory (IRT) on the ASAP and ASAP++ datasets (holistic and analytic scores across eight prompts). The methodology incorporates two grammatical feature types: (1) correctly used grammatical items (PFs) and (2) error counts (NFs), weighted via IRT parameters to reflect item difficulty and writer ability. As outlined in the key findings, MTL with IRT-weighted features achieves near-human performance accuracy in scoring accuracy, annotation independent, eliminating the need for labels. Su et al. (2025) introduce EssayJudge, the first multimodal benchmark for evaluating Automated Essay Scoring (AES) capabilities of Multimodal Large Language Models (MLLMs) across lexical-, sentence-, and discourse-level traits. Leveraging a dataset of 1,054 high-quality multimodal English essays (text + images) spanning 125 topics, the authors assess 18 MLLMs using Quadratic Weighted Kappa (QWK) against human evaluations. Highlights show that MLLMs perform well on lexical and sentence-level assessments, such as grammar and vocabulary, but lag behind human raters on coherence and argumentation, indicating a shortcoming in contextual reasoning. The benchmark seeks to resolve three primary shortcomings of automated essay scoring (AES): (1) reliance on handcrafted features, (2) inability to capture granular writing traits, and (3) lack of integration of multimodal context.

Reviewing the current literature has revealed multiple critical gaps related to Automated Essay Scoring (AES) and Automated Short Answer Grading (ASAG), as well as broader Automated Writing Evaluation (AWE) systems for the Arabic language. These gaps fall into three primary categories:

(1) Linguistic Resource Challenges: These systems suffer from a scarcity of standardized datasets and unified benchmarks for Arabic compared to other languages, attributable to the unique morphological and syntactic complexity of Arabic impedes contextual semantic processing, The multiplicity of local dialects and diverse linguistic structures and Conventional methods' inability to capture the semantic and contextual complexity of Arabic texts fully.

(2) Training Data Challenges: A severe shortage of annotated data is observed, particularly for extended essay responses, due to the high cost of human grading, and Human rater variability (Inter-rater Variability) negatively impacts model training stability.

(3) Generalization and Modeling Challenges: The existing systems are still unable to generalize across various questions and domains without human assistance, making fully automating electronic grading impossible. Three main categories address the issue of optimal feature selection, which are computer-aided and highly subjective or tedious manual processes. Computer-aided feature selection requires high-performance algorithms, whereas manual methods are inefficient, labor-intensive, and prone to bias. Several studies utilized hybrid approaches. In addition to lacking a universally agreed-upon standard in this field.

3 Research methodology

This investigation aims to develop a novel framework for Automated Short Answer Grading (ASAG) of Arabic texts, utilizing AraBERT's semantic and contextual embeddings in conjunction with advanced neural network architectures, including MLP-NN, CNN, and LSTM. The study undertakes a comparative evaluation of different input feature combinations, which are defined as: (1) reference answer-student answer pairs, (2) question-reference answer and student answer, and (3) question-reference answer-student answer-human expert score. With this methodology, we aim to create a comparative analytical method for determining the most effective framework that captures the subtle intricacies of the Arabic language while increasing robustness and generalization across various question types and domains. The proposed system achieves appropriate grading accuracy by proper feature representation and optimization of the neural network, thus significantly advancing the capabilities of AR-ASAG in Arabic. The following sections describe the proposed methodology.

AraBERT is an advanced Arabic pre-trained language model based on Google's BERT (Bidirectional Encoder Representations from Transformers) architecture, with two main versions: the base AraBERTv0.1 and AraBERTv1, which employs pre-segmented text using the Farasa Segmenter for morphological analysis. The model has demonstrated superior performance across multiple Arabic NLP tasks, including sentiment analysis on six benchmark datasets (HARD, ASTD-Balanced, ArsenTD-Lev, LABR), named entity recognition using ANERcorp, and Arabic question answering with Arabic-SQuAD and ARCD, outperforming comparable models like multilingual BERT. For this study, we utilize AraBERTv2-large (bert-large-arabertv2), "aubmindlab/bert-base-arabertv02", a state-of-the-art variant with 371 million parameters (1.38GB in size) that employs text pre-segmentation and was trained on 200 million sentences (77GB of textual data equivalent to 8.6 billion tokens), establishing it as one of the most sophisticated models for Arabic language understanding and processing (Antoun et al., 2020).

3.1 Dataset description

The proposed model is used on one of the Arabic scarce publicly available datasets, which is called (AR-ASAG). AR-ASAG, Arabic Dataset for Automatic Short Answer Grading, is the first openly and freely available Arabic dataset (Ouahrani and Bennouar, 2020). It contains questions taken from the cybercrimes teaching course and the responses of three classes of master's students. There are a total of 2,133 student responses in the dataset. There is a suggested model response for each question. Two human experts assessed the responses independently on a scale from 0 (totally inaccurate) to 5 (perfect answer). Both of the experts were instructors in computer science. AR-ASAG considered the gold standard to be the average grade of the two experts. There are several versions of the AR-ASAG Dataset, including TXT, XML, XML-MOODLE, and Database (.DB). Table 1 shows the distribution of Answers by Question Type.

3.2 Applied methodology

This section provides a comprehensive methodological description that consists of two main distinct steps: AraBERT training and ARaBERT Testing. The following Figure 1 illustrates, through a detailed flowchart, the logical sequence of implementation steps.

3.2.1 Preprocessing

Before training the model, the input texts in Arabic went through standard stages of preprocessing to improve the data and the model's performance. These steps involved the splitting of texts into individual components, the cessation of the participation of unimportant constituents, and the purging of irrelevant symbols, subsequently making sure the texts kept their essence without irrelevant frills. This preprocessing pipeline was carried out through NLTK using a typical list of Arabic stopwords, thereby yielding input text that was much clearer and more normalized, ready for embedding in AraBERTv2 and later neural network architectures.

3.2.2 Data splitting

This study employed an 80:20 data partitioning methodology utilizing a question-wise splitting strategy rather than conventional random splitting. Specifically, 20% of random responses for each question were systematically allocated *a priori* to constitute the test set, before any data processing or model implementation. This approach ensured complete segregation between training and test datasets while preventing potential data leakage between the two subsets. Table 2 presents the detailed distribution of randomly sampled responses across selected questions.

3.2.3 Feature selection

This investigation advances a comprehensive systematic feature study toward optimizing an ASAG system. Controlled experiments assessed three increasingly complex input configurations:

Dual-Feature Model/2-features: (Reference answer-student answer pair), these two features are essential for answer evaluation on a semantic, contextual, and similarity level.

Triple-Feature Model/3-features: (Question, reference answer, and student answer), including the question helps ensure the student's response is on topic since answers can be partially correct but completely off topic.

Quad-Feature Model/4-features: (Question, reference answer, student answer, and expert score). This combination enables the model to emulate human expert evaluation by learning non-linear feature relationships, thereby improving scoring accuracy.

The AraBERT model will be trained separately on each feature set, yielding three distinct models:

- A 2-feature model (reference answer + student answer).
- A 3-feature model (question + reference answer + student answer).
- A 4-feature model (question + reference answer + student answer + human score).

These models will generate comprehensive semantic and contextual embedding representations of the input features. Finally, the extracted representations will serve as inputs to three neural network architectures to capture nonlinear feature interactions and predict final scores with maximal accuracy.

3.2.4 AraBERTv2 training stage

This study develops a comparative framework for automatically scoring Arabic short answers assessed through three hybrid architectures of AraBERT:

(1) AraBERTv2-MLP applies a multilayer perceptron regression on contextual embeddings;
(2) AraBERTv2-CNN applies spatial feature extraction through convolutional layers;
(3) AraBERTv2-LSTM employs recurrent networks for capturing sequential dependencies.

Each configuration focuses on answering distinct desiderata: semantic understanding, local pattern recognition, and temporal

TABLE 1 Distribution of answers by question type.

Question type	Question type (In Arabic)	Total questions	Total answers
Define the scientific term	عرف المصطلح العلمي	6	291
Explain	إشرح	21	830
What are the consequences of	ما النتائج المترتبة على	6	282
Justify or give reasons for	علل	10	465
What is the difference between	ما الفرق بين	5	217
Total	5 types	48	2,085

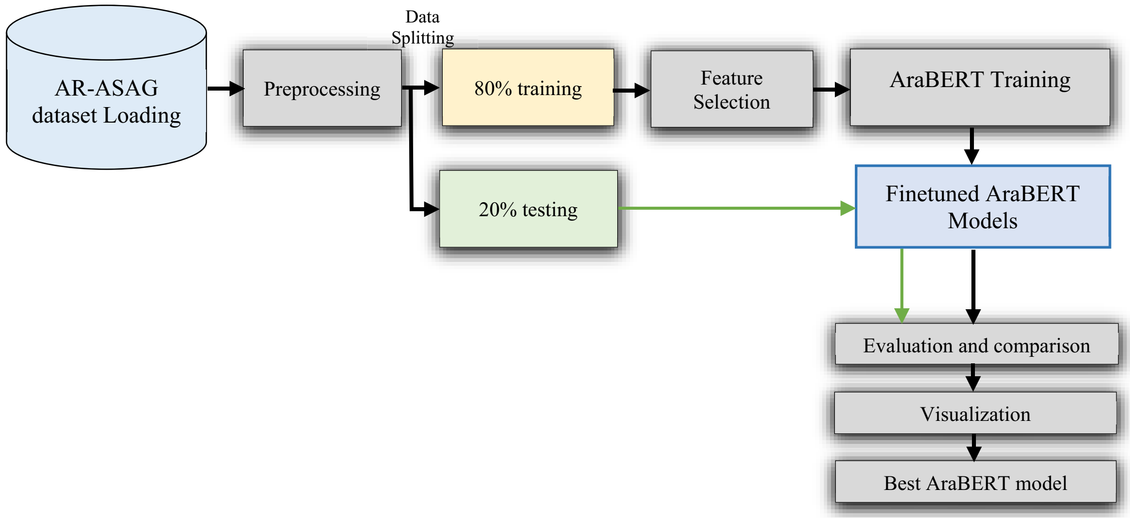

FIGURE 1
General workflow of the proposed automated Arabic short-answer grading model using AraBERTv2.

TABLE 2 Detailed distribution of randomly sampled responses across selected questions.

Q–No.	Question type	Total answers	Training answers	Test answers
1	Define the scientific term	46	36	10
26	Explain	47	37	10
28	What are the consequences of	48	38	10
35	Justify or give reasons for	51	40	11
45	What is the difference between	36	28	8

coherence. All while applying dropout regularization ($p = 0.2$) to limit overfitting. The comparative analysis of these architectures gives insights into feature extraction and scoring strategies for Arabic educational contextualized. This hybrid architecture achieves a favorable balance of performance, computation, and natural language efficiency, which is crucial for the automated evaluation of text responses in educational assessment frameworks. The models represent an advancement from conventional scoring paradigms through purpose-driven neural hybridization. It is noteworthy that the human expert scores were normalized from a 0–5 scale to a 0–10 scale to align with the output range of the fine-tuned language models. This normalization facilitated consistent performance evaluation during model training and testing. Below is a short overview of fine-tuning the AraBERTv2 architecture with these three different configurations.

3.2.4.1 AraBERTv2 with MLP training

In this section of work, we focus on fine-tuning the AraBERTv2 model by using the Hybrid AraBERTv2-MLP model for automatic classification/regression of Arabic responses, which stems from the advantages provided by the pre-trained AraBERTv2 model and the Multilayer Perceptron (MLP) model. The complete system contains two components:

AraBERv2T Base Layer: Textual feature extraction takes place through "bert-base-arabertv02" (aubmindlab) for Arabic texts. This layer processes input texts and outputs a 768-dimensional vector (embedding) spatially, which captures the words' contextual linguistic features as well as their relationships.

MLP Regressor Layer: This layer corresponds to a sequential Multilayer Perceptron (MLP) composed of:

- Input linear layer (768 → 256 neurons).
- ReLU-activated hidden layer.
- Output linear layer (256 → 1 neuron).
- Sigmoid function to bound outputs to (0–1)
- A final operation to scale (×10) projects the output into a grading scale of (0–10).

There is a Dropout layer (rate = 0.2) placed between the two main layers that functions to counter overfitting by randomly turning off a fraction of the units for the duration of training.

The model fine-tuning Process was trained using a vertically integrated approach, involving Mean Squared Error (MSE) as the loss function to quantify deviation between predictions and actual values. The AdamW optimizer algorithm was employed, with an initial learning rate of 2e−5. Execution Environment: As a matter of principle, both the model and the data will be moved to a GPU-sensitive unit if one exists to accelerate the execution performance. The cumulative loss for all batches is computed.

The data is separated into two main categories: Training set (trainloader), which is used in the model parameter fine-tuning optimization process. Validation set (valloader): Used in the assessment of the model's performance. The fine-tuning process utilized outputs from the preceding feature selection stage, comprising three distinct feature combinations: (1) a 2-features model (reference answer + student answer), (2) a 3-features model (question + reference answer + student answer), and (3) a 4-features model (question + reference answer + student answer + human score). The AraBERTv2 model underwent separate fine-tuning procedures for each feature combination, thereby generating three specialized fine-tuned AraBERTv2 variants. Figure 2 provides a schematic representation of the AraBERT-MLP training methodology

The AraBERT Setting used in this training stage:

```
def __init__(self, model_name="aubmindlab/bert-base-arabertv02", dropout_prob=0.2):
    super(AraBERTGrader, self).__init__()
    self.bert = AutoModel.from_pretrained(model_name)
    self.config = AutoConfig.from_pretrained(model_name)
    self.dropout = nn.Dropout(dropout_prob)
    self.regressor = nn.Sequential(
        nn.Linear(768, 256),
        nn.ReLU(),
        nn.Linear(256, 1),
        nn.Sigmoid() )
```

3.2.4.2 AraBERTv2 with CNN

The integrated architecture presented in this section combines the language model AraBERTv2 embedding and a Convolutional Neural Network (CNN) to grade Arabic answers automatically. Its architecture consists of three main components.

AraBERTv2 Base Model: generates 768-dimensional contextual embeddings from input text, leveraging a pretrained AraBERTv2 variant to capture rich linguistic features of Arabic through its transformer-based architecture.

CNN Feature Extractor: A one-dimensional convolutional neural network (CNN) consisting of the following:

- Two convolutional layers (768 → 256 → 128 channels) with kernel size 3.
- ReLU activation functions.
- Max pooling (standard and adaptive).
- This is designed to capture local n-gram and hierarchical textual features from the sequence embeddings.

MLP Regressor: The final neural component consisting of a Linear layer (128 → 1), Sigmoid activation, and Score scaling (×10). This multilayer perceptron serves as the scoring head, transforming extracted features into numerical grades.

The model undergoes supervised training with dropout regularization ($p = 0.2$) applied to embeddings, end-to-end optimization via backpropagation, and pretrained AraBERTv2 weights fine-tuned alongside CNN parameters. Finally, the model persists through state dictionary saving/loading. The final MLP layer plays three crucial roles: dimensionality reduction from feature space to scalar output, non-linear mapping of learned representations to scoring scale, output normalization via sigmoid activation, and scale adaptation to (0–10) grading range. This architecture demonstrates effective synergy between AraBERT2′s semantic comprehension, CNN's local pattern detection, and MLP's regression capabilities. Figure 3 provides a schematic representation of the AraBERT-CNN training methodology.

The AraBERTv2 Setting used in this training stage:

```
def __init__(self, model_name="aubmindlab/bert-base-arabertv02", dropout_prob=0.2):
    super(AraBERTGrader, self).__init__()
    self.bert = AutoModel.from_pretrained(model_dir)
    self.config = AutoConfig.from_pretrained(model_dir)
    self.dropout = nn.Dropout(0.2)
    self.cnn = nn.Sequential(
    nn.Conv1d(768, 256, kernel_size=3, padding=1),
    nn.ReLU(),
    nn.MaxPool1d(kernel_size=2),
    nn.Conv1d(256, 128, kernel_size=3, padding=1),
    nn.ReLU(),
    nn.AdaptiveMaxPool1d(1) )
```

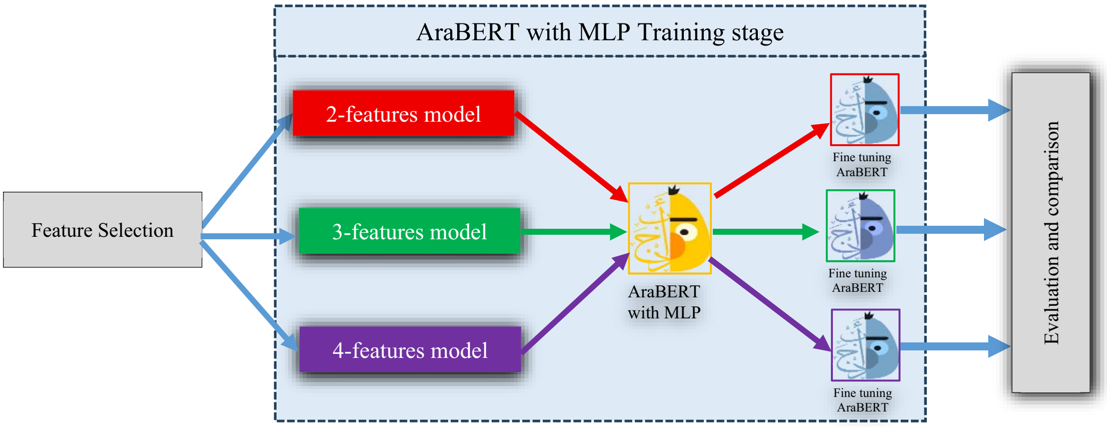

FIGURE 2
The AraBERT_MLP training methodology.

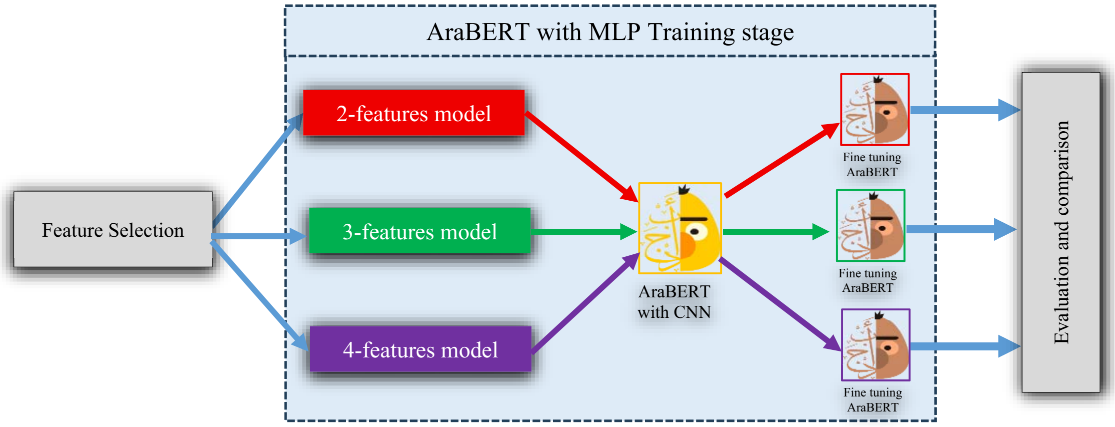

FIGURE 3
The AraBERT_CNN training methodology.

```
    self.regressor = nn.Sequential(
      nn.Linear(128, 1),
      nn.Sigmoid() )
```

3.2.4.3 AraBERTv2 with LSTM layer

This part represents a novel advanced hybrid architecture that uses the pre-trained AraBERTv2 combined with LSTM networks to capture temporal dependencies for inputs. It has three core components:

AraBERTv2 Base Layer: AraBERTv2 Base Layer using the bert-base-arabertv02 pre-trained model used to extract contextual embeddings (768-dimensional). Adds Dropout ($p = 0.2$) for better generalization and to prevent overfitting.

LSTM Temporal Processing Layer: This is a unidirectional LSTM with a hidden size of 256. It processes sequential features extracted from AraBERT and outputs Dropout for added robustness.

MLP Regressor: A different set is, the MLP scaler method becomes a regression head to order

- A linear layer (256 → 128) with ReLU activation.
- Additional Dropout layer ($p = 0.2$).
- Final linear layer (128 → 1) with Sigmoid activation for score normalization.
- Output scaled to a grading range (0–10).

The given model is trained under supervised training using Mean Squared Error (MSE) as a Loss Function, for comparing our prediction with our target and AdamW + LR (2e−5) optimizer. This architecture serves critical functions such as dimensionality reduction from temporal features to scalar predictions, score normalization via Sigmoid activation (0–1 range), overfitting prevention through Dropout regularization, and precision-efficiency balance in grade estimation. Figure 4 provides a schematic representation of the AraBERT-LSTM training methodology.

The architecture demonstrates effective synergy between deep linguistic understanding (AraBERTv2), temporal sequence processing (LSTM), and Numerical prediction accuracy (MLP). This integrated approach shows particular efficacy for handling Arabic morphological complexity, processing short-answer semantic relationships, and maintaining scoring consistency. The AraBERTv2 Setting used in this training stage:

```
    def    __init__(self,    model_name="aubmindlab/bert-base-
arabertv02", dropout_prob=0.2):
super(AraBERTGrader, self).__init__()
        self.bert = AutoModel.from_pretrained(model_name)
        self.config = AutoConfig.from_pretrained(model_name)
        self.dropout = nn.Dropout(dropout_prob)
        self.lstm = nn.LSTM(
          input_size=768,
          hidden_size=256,
          num_layers=1,
          batch_first=True,
          bidirectional=False )
```

3.2.5 AraBERTv2 testing stage

This study evaluates nine fine-tuned models derived from previous training phases, each developed using distinct feature combinations and neural architectures. For the testing phase, all models will be assessed against a rigorously isolated test set that was strategically partitioned from the original dataset, maintaining complete separation from both training and validation subsets to ensure unbiased evaluation, as specified in Section "Preprocessing". Performance will be measured using standard evaluation metrics, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Pearson Correlation Coefficient (r), and Spearman's Rank Correlation Coefficient (ρ). Detailed metric definitions are provided in the following Section 3.2.6. A comparative analysis of all evaluation results will identify the optimal model based on a comprehensive performance assessment.

3.2.6 Evaluation

These systems incorporate a self-evaluation mechanism that automatically compares automated scoring results with human

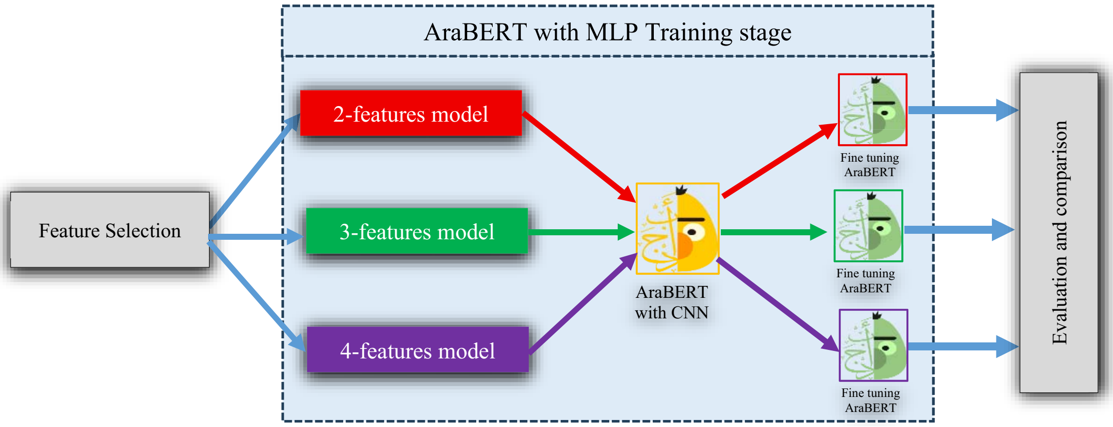

FIGURE 4
The AraBERT_LSTM training methodology.

grader scores through a set of precise statistical metrics (Géron, 2017), including:

- Mean Absolute Error (MAE): Measures the average absolute difference between predicted and human scores.

Quantifies scoring accuracy in the original unit (e.g., 0–5 scale). Lower values indicate better alignment with human graders.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{1}$$

- Root Mean Square Error (RMSE): The Square root of the average squared differences between predicted and actual scores. Penalizes larger errors more severely than MAE, making it sensitive to outlier scores.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

- Pearson Correlation Coefficient (r): Measures linear correlation between predicted and human scores (-1 to 1). Indicates whether the system maintains human ranking consistency (higher values indicate better performance).

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{3}$$

- Spearman's Rank Correlation Coefficient (ρ): Assesses monotonic (not necessarily linear) relationships using score ranks. Evaluates if the system preserves ordinal relationships, robust to non-linearities.

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)} \tag{4}$$

4 Results and discussion

The following sections present the study findings systematically and sequentially. It is important to note that all experiments were conducted on a Dell machine equipped with a 12th-generation Core i7 processor and running the Windows 11 operating system. The proposed methodology was implemented using a Python 3.12.3 environment with various TensorFlow libraries.

4.1 Results of fine-tuned AraBERTv2 with MLP

The evaluation shown in Figure 5 reveals distinct performance patterns across feature configurations during training. The 4-features model demonstrates exceptional performance with near-perfect correlation scores (Pearson = 0.999, Spearman = 0.998) and minimal error metrics (MAE = 0.18, RMSE = 0.20), accompanied by rapid loss reduction (713 → 7), suggesting potential overfitting. In contrast, the 2-features model shows more moderate but stable performance (Pearson = 0.847, MAE = 1.14) with gradual loss reduction (898 → 156), while the 3-features configuration presents intermediate results with comparable stability. Performance degradation occurs across all models during testing, with the 4-feature variant exhibiting the most severe drop (MAE increase from 0.18 to 1.77, correlation decrease by ∼30%), confirming overfitting concerns. The 2-features model maintains superior generalization (MAE = 1.31, Pearson = 0.803) with minimal performance gap between phases, while the 3-features model shows moderate degradation (MAE = 1.48, Pearson = 0.744). The configuration with 2-features shows the best compromise between the efficiency of the training phase's performance and the reliability of the test phase. The 4-features

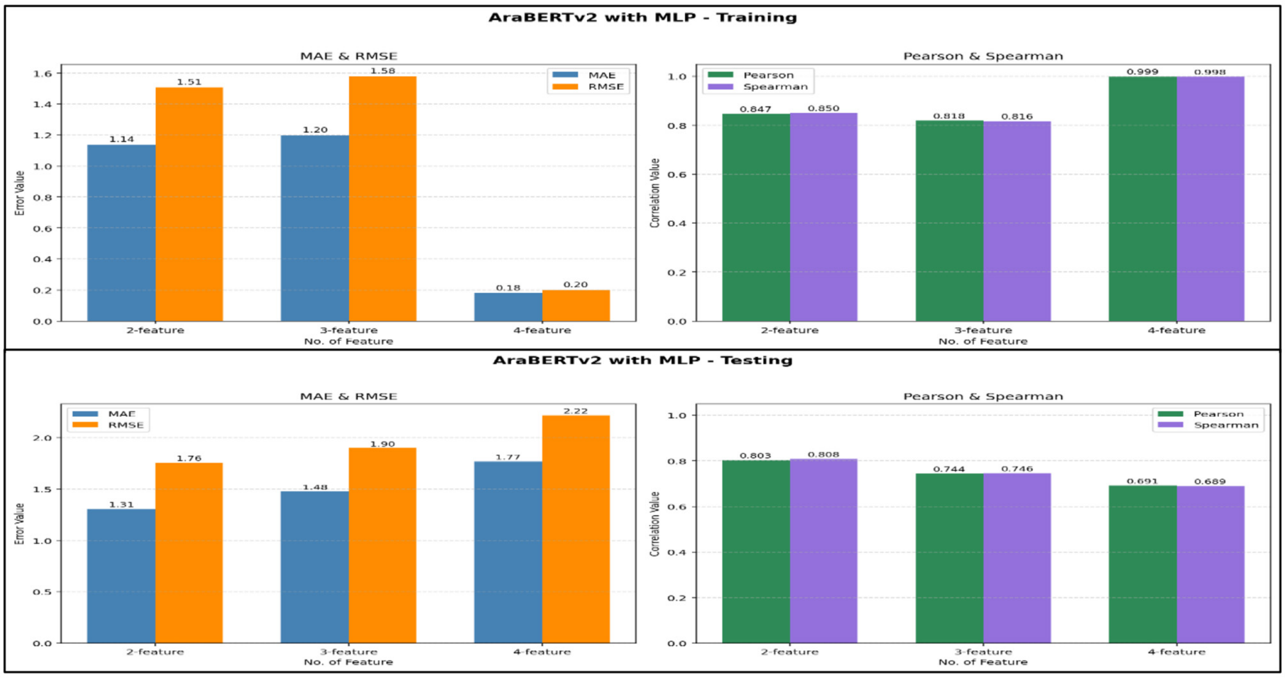

FIGURE 5
Performance evaluation of AraBERTv2 with MLP model using different feature sets: training vs. testing results.

TABLE 3 Performance evaluation of AraBERTv2 with MLP model using different feature sets: training vs. testing results.

Model	Stage	No. of feature	MAE	RMSE	Pearson correlation	Spearman's correlation	Epoch 1–5
AraBERTv2 with MLP	Training	2-feature	1.14	1.51	0.847	0.85	898 → 533 → 347 → 250 → 156
		3-feature	1.2	1.58	0.818	0.816	1,026 → 614 → 263 → 185
		4-feature	0.18	0.2	0.999	0.998	713 → 34 → 13 → 9 → 7
	Testing	2-feature	1.31	1.76	0.803	0.808	
		3-feature	1.48	1.9	0.744	0.746	
		4-feature	1.77	2.22	0.691	0.689	

model's sharp decline illustrates the performance catastrophe associated with unnecessary intricacy. The results reinforce the idea that generalization should be the primary concern in model selection, rather than achieving a flawless fit in training. Prioritize generalization capability over perfect training fit.

The Table 3 presents the performance Evaluation results of AraBERTv2 with MLP Model Using Different Feature Sets during the training and testing stages.

4.2 Results of fine-tuned AraBERTv2 with CNN

The training results in Figure 6 reveal distinct performance patterns across different feature configurations. The 4-features model achieved near-perfect training performance (MAE = 0.24, RMSE = 0.27) with near-unity Pearson (0.999) and Spearman (0.998) correlations, accompanied by a sharp decline in loss (773 → 6), suggesting potential overfitting despite strong initial convergence. Conversely, the 2-features model showed somewhat more moderate but still robust performance (MAE = 1.22, Pearson = 0.849), with consistent loss reduction (1,092 → 227), indicating stable learning. The 3-features model produced intermediate performance (MAE = 1.17, Pearson = 0.833), accompanied by a parallel gradual loss decline (1,057 → 205), demonstrating a balance between the complexity of the model and generalization.

During testing, all models showed some degree of performance degradation, with the 4-features model suffering the most severe drop (MAE = 2.63, Pearson = 0.607). The 2-features model maintained the strongest performance with the best generalization (MAE = 1.45, Pearson = 0.784), while the 3-features model experienced a moderate drop (MAE = 1.6, Pearson = 0.746). These results indicate an optimal 2-features model configuration, where fewer features provided greater robustness while more features provided diminishing returns to generalization despite favorable training performance. The Table 4 presents the performance Evaluation results of AraBERTv2 with CNN Model Using Different Feature Sets during the training and testing stages.

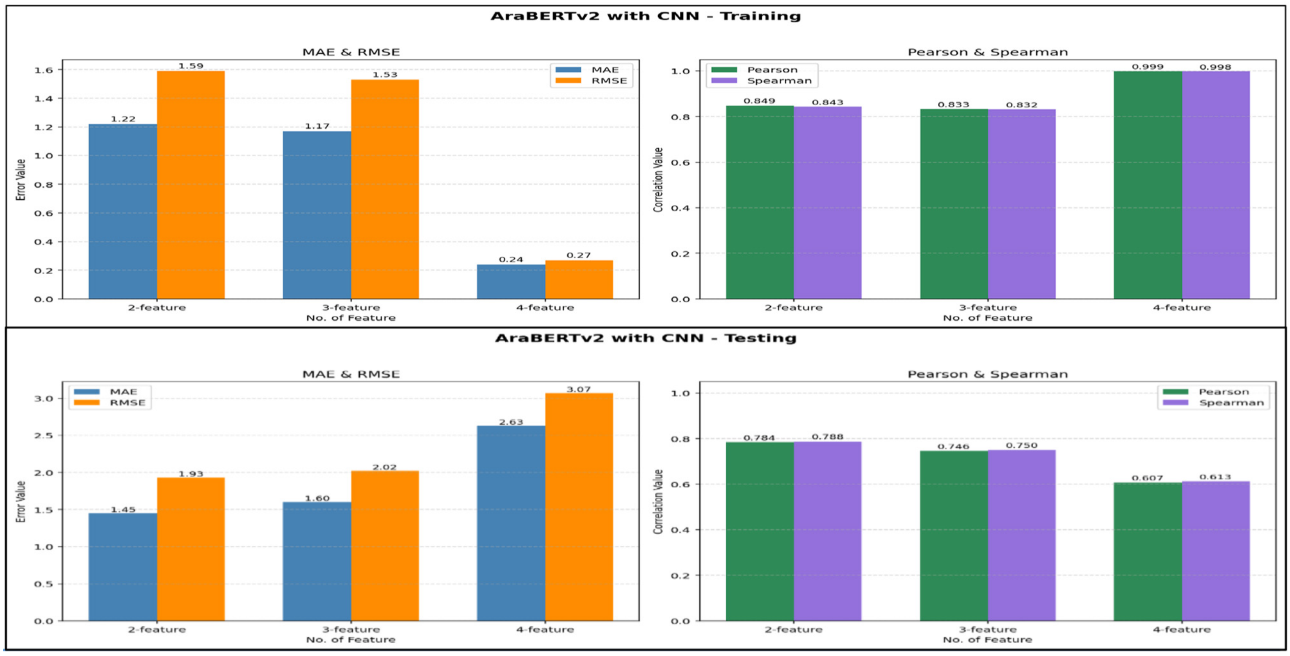

FIGURE 6
Performance evaluation of AraBERTv2 with CNN model using different feature sets: training vs. testing results.

TABLE 4 Performance evaluation of AraBERTv2 with CNN model using different feature sets: training vs. testing results.

Model	Stage	No. of features	MAE	RMSE	Pearson correlation	Spearman's correlation	Epoch 1–5
AraBERTv2 with CNN	Training	2-feature	1.22	1.59	0.849	0.843	1,092 → 610 → 427 → 306 → 227
		3-feature	1.17	1.53	0.833	0.832	1,057 → 567 → 379 → 280 → 205
		4-feature	0.24	0.27	0.999	0.998	773 → 28 → 12 → 8 → 6
	Testing	2-feature	1.45	1.93	0.784	0.788	
		3-feature	1.6	2.02	0.746	0.75	
		4-feature	2.63	3.07	0.607	0.613	

4.3 Results of fine-tuned AraBERTv2 with LSTM

The training results in Figure 7 disclose three specific patterns. The 4-features model not only performed excellently but also attained an MAE of 0.14 and a Pearson score of 0.998, while converging from 728 to 19 loss. However, this might indicate some overfitting despite using a dropout value of 0.2. Both 2-features and 3-features models showed comparable, more moderate performance (MAE = 1.26–1.27, Pearson = 0.81–0.82) with steady loss reduction patterns (1,147 → 262 and 1,141 → 267, respectively), indicating stable learning dynamics. Notably, the 3-features model showed slightly worse metrics than the 2-features version despite higher complexity.

Testing has revealed extreme differences in performance as follows: The 4-features model experienced catastrophic failure (MAE = 3.62, Pearson = 0.388), confirming extreme overfitting. In contrast to the 2-features model which demonstrated more robust performance (MAE = 1.48, Pearson = 0.757) and outperformed the 3-features version, which returned a MAE of 1.60, although both yielded identical Pearson scores. It is also noteworthy that the three-feature model was less accurate than the two-feature model but had no correlation advantage.

In summary, the two-feature LSTM configuration stands out as the most dependable architecture, exhibiting the best balance between overfitting and generalization. These findings suggest that for LSTM networks, the degree of feature intricacy does more than fail to enhance performance: It actively undermines a model's robustness when exposed to unseen data. The Table 5 presents the performance Evaluation results of AraBERTv2 with LSTM Model Using Different Feature Sets during the training and testing stages.

4.4 Comparison with previous works

The Tables 6, 7 shows a comparative performance evaluation of our proposal, Arabic Automated Short Answer Grading (ASAG) Systems, with previous works that utilize the same dataset.

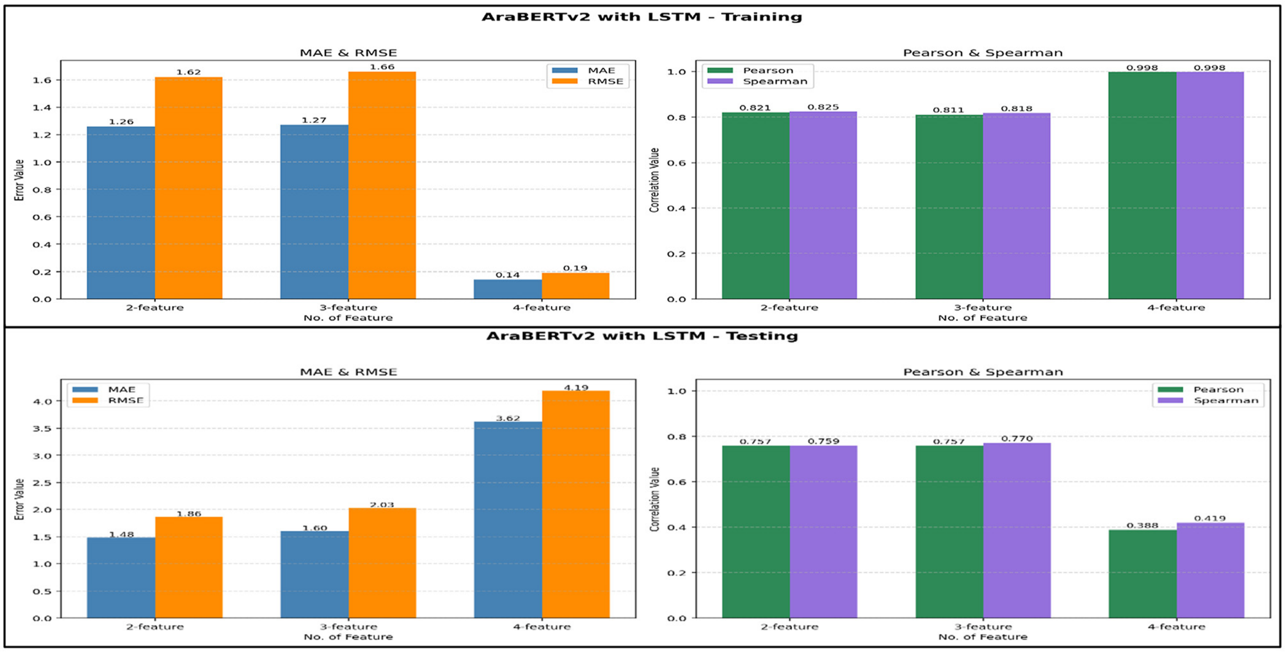

FIGURE 7
Performance evaluation of AraBERTv2 with LSTM model using different feature sets: training vs. testing results.

TABLE 5 Performance evaluation of AraBERTv2 with LSTM model using different feature sets: training vs. testing results.

Model	Stage	No. of features	MAE	RMSE	Pearson correlation	Spearman's correlation	Epoch 1–5
AraBERTv2 with LSTM	Training	2-feature	1.26	1.62	0.821	0.825	1,147 → 718 → 524 → 356 → 262
		3-feature	1.27	1.66	0.811	0.818	1,141 → 675 → 456 → 349 → 267
		4-feature	0.14	0.19	0.998	0.998	728 → 62 → 31 → 22 → 19
	Testing	2-feature	1.48	1.86	0.757	0.759	
		3-feature	1.6	2.03	0.757	0.77	
		4-feature	3.62	4.19	0.388	0.419	

TABLE 6 Performance comparison of AraBERTv2 fine-tuned models with MLP, CNN, and LSTM architectures using different feature sets.

Fine-tuned models	MAE	RMSE	Pearson correlation	Spearman's correlation
2-features-AraBERTv2 with MLP	**1.31**	**1.76**	**0.803**	**0.808**
2-features-AraBERTv2 with CNN	1.45	1.93	0.784	0.788
2-features-AraBERTv2 with LSTM	1.48	1.86	0.757	0.759
3-features-AraBERTv2 with MLP	1.48	1.9	0.744	0.746
3-features-AraBERTv2 with CNN	1.6	2.02	0.746	0.75
3-features-AraBERTv2 with LSTM	**1.6**	**2.03**	**0.757**	**0.77**
4-features-AraBERTv2 with MLP	**1.77**	**2.22**	**0.691**	**0.689**
4-features-AraBERTv2 with CNN	2.63	3.07	0.607	0.613
4-features-AraBERTv2 with LSTM	3.62	4.19	0.388	0.419

The bold values represent the optimal results obtained from our experimental analysis.

Our study demonstrates an optimal equilibrium between predictive accuracy (RMSE 1.31) and model generalizability under data constraints, while (Meccawy et al., 2023) Achieves marginally superior precision (RMSE 1.003) through computationally intensive text preprocessing that may limit operational flexibility in novel contexts. The choice between approaches depends on whether absolute accuracy (BERT) or generalizability with limited resources (AraBERTv2+MLP) represents the priority.

TABLE 7 Comparative performance evaluation of Arabic Automated Short Answer Grading (ASAG) systems.

Criterion/study	Methodology	Dataset	Best RMSE	Best Pearson/Spearman	Key strength	Primary limitation
Our study (AraBERTv2)	- Fine-tuned AraBERTv2 with MLP/CNN/LSTM - Tested 2/3/4 feature configurations	AS-ARSG (2,133 answers)	1.31	- Pearson: 0.803 - Spearman: 0.808	Optimal balance between generalizability and accuracy with limited data	Performance degradation in LSTM with added features
(4)	Latent Semantic Analysis (LSA) with local/hybrid weighting	AR-ASAG (2,133 answers)	N/A	N/A	Effective semantic weighting	Limited capacity for capturing complex contextual relationships
(19)	- BERT vs. Word2Vec/AWN comparison - Intensive text preprocessing	- AR-ASAG (2,133) - Jordanian History (550)	1.00308	Pearson: 0.841902	Demonstrated BERT's superiority over traditional approaches	Heavy dependency on text normalization and stemming

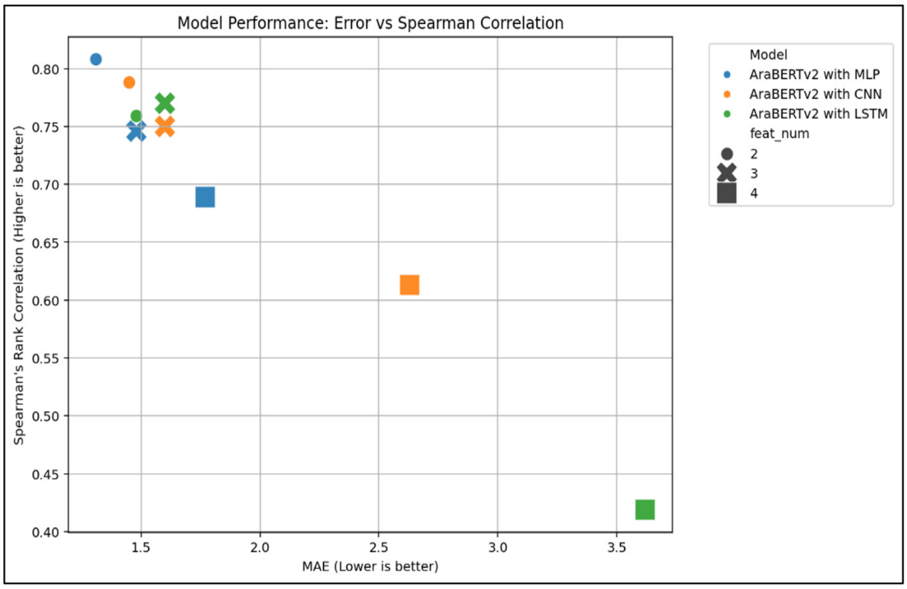

FIGURE 8
Fine-tuned models performance: MAE vs. spearman correlation.

5 Discussion

The discussion in this section aims to analyze and evaluate the performance of nine fine-tuned AraBERTv2 language models employing three distinct architectures: MLP, CNN, and LSTM. The Table 6 illustrates the evaluation results obtained from fine-tuning AraBERTv2 models using different neural architectures (MLP, CNN, LSTM) and varying the number of features used (2, 3, and 4 features). The 2-feature models consistently outperform the other models with a greater number of features for all the metrics used (MAE, RMSE, Pearson, and Spearman correlations). Notably, 2-features AraBERTv2 with MLP performed best (MAE = 1.31, RMSE = 1.67, Pearson = 0.803, and Spearman = 0.808), suggesting that for this task, simpler models with fewer components extract features and reduce overfitting. On the other hand, 3-features and 4-features seem to contribute to a downward performance spiral that culminates in the 4-features LSTM yielding the worst performance (MAE = 3.62, Pearson = 0.388). Interestingly, while LSTM is competitive for 3-features, its performance relative to other architectures and models declines sharply with 4-features, while MLP is less sensitive to feature increase. The results suggest that a tradeoff between the number of features and the complexity of the model is needed in designing Arabic NLP systems.

The graph in Figure 8 illustrates the relationship between two key performance metrics: Mean Absolute Error (MAE) on the horizontal axis (where lower values display better performance) and Spearman's Correlation Coefficient on the vertical axis (where higher values denote superior performance).

The best performing models have been distinctly grouped in the upper-left quadrant of the graph with optimal values. Leading the group is the AraBERTv2 with MLP model (in blue circle), which achieved the highest Spearman's correlation of approximately 0.81 and one of the lowest MAE values of around 1.35. Following closely is AraBERTv2 with CNN model, noted as an orange '*x*', demonstrating very competitive results with a strong Spearman's correlation of approximately 0.78 and a low MAE of 1.45. AraBERTv2 with LSTM model, shown as a green square, also displays strong results with a Spearman's correlation of 0.77 and a low MAE of 1.55. These results indicate that all three architectures can deliver strong results when using an appropriate number of features.

Conversely, some models exhibit notably poor performance. The AraBERTv2 with LSTM using 4-features (larger green square) ranks as the worst performer, displaying the highest MAE (exceeding 3.5) and lowest Spearman's correlation (below 0.45). This sharp performance degradation suggests that increasing feature count in this architecture may be counterproductive. Additionally, the AraBERTv2 with MLP using 4-features (large gray square) demonstrates mediocre-to-poor performance, with MAE above 1.8 and correlation below 0.7, confirming that performance depends not only on architecture but significantly on feature quantity.

The investigation demonstrates that achieving optimal performance within a model necessitates a balance between its architecture and the selected features. AraBERTv2 with MLP and minimal features (features_num = 2) performed the best, thus marking it as the best choice from the experiment. This is related to minimizing noise that arises from incorporating a greater number of less informative features. This observation corresponds with established concepts in NLP and ASAG research, where simpler models tend to generalize more effectively in situations with limited data, while also preserving interpretability.

These results stress the need for thorough evaluation of a model under different scenarios, while also noting that augmenting model complexity by adding features does not improve performance.

6 Conclusion

This section provides a dedicated conclusion that highlights the significance of the study, summarizes the key findings, acknowledges its limitations, and outlines potential directions for future research.

6.1 Study importance

This research study was designed to systematically evaluate and compare the performance of nine distinct AraBERTv2 model configurations incorporating three neural network architectures (MLP CNN LSTM) with varying feature set sizes (2 3 4 features) The primary objective centered on developing an automated Arabic question scoring system capable of generalization while minimizing dependence on human-annotated training data with particular focus on optimizing prediction accuracy through MAE and RMSE reduction while maximizing correlation metrics including Pearson and Spearman coefficients. The experimental results revealed several significant findings that advance our understanding of Arabic language model optimization The 2-features AraBERTv2 implementation with MLP architecture demonstrated superior performance across all evaluation metrics achieving an MAE of 1.31 and Spearman correlation of 0.808 establishing it as the most effective configuration for this specific task Performance exhibited consistent degradation as feature complexity increased with this effect being particularly pronounced in LSTM architectures where the 4-feature model showed substantially degraded results (MAE 3.62 and Spearman 0.419). Comparative analysis indicated MLP architectures maintained greater robustness during feature expansion relative to both CNN and LSTM variants, with results clearly illustrating an inverse relationship between model complexity as measured by feature count and overall predictive performance. These findings make substantive contributions to Arabic NLP research by establishing empirical guidelines for architecture-feature optimization demonstrating the viability of reduced-feature models for automated scoring applications and highlighting the potential pitfalls of unnecessary model complexity The work provides a concrete framework for developing expert-independent scoring systems while emphasizing the critical importance of balanced architecture-feature selection over indiscriminate model complexity enhancement offering practical implementation insights for educational technology applications in Arabic language assessment contexts.

The primary advancements of this research encompass the implementation of the first hybrid architecture that combines AraBERTv2′s linguistic capabilities with optimized regression capabilities of advanced neural networks for Arabic assessment; rigorous benchmarking of alternative architectures using controlled feature sets; development of reproducible evaluation protocols for Arabic NLP tasks; demonstration of resource-efficient model optimization principles; and deployment of an operational system that meets both accuracy and latency requirements for educational applications.

6.2 Limitation

Several significant limitations must be acknowledged when interpreting these findings. The study's exclusive reliance on the AS-ARSG dataset may affect generalizability to other Arabic language corpora. At the same time, the constrained feature range investigation (2–4 features) potentially overlooks performance characteristics in more complex feature environments. The research focus remained limited to AraBERTv2 without comparative analysis against alternative transformer architectures, and computational resource constraints prevented exploration of potentially valuable hybrid architectures or deeper network configurations. Further reflection is also needed on dataset size, potential bias from domain-specific content, and the absence of student demographic variation, which may have influenced the observed results and limited broader applicability.

6.3 Future works

Future research directions emerging from this work should prioritize several key areas including multi-corpus validation to strengthen result reliability and generalizability alongside development of advanced feature selection methodologies that transcend basic count-based approaches, investigation of ensemble architectures combining the respective strengths of MLP CNN and LSTM approaches warrants attention as does expansion to diverse Arabic NLP tasks and datasets, development of sophisticated attention mechanisms capable of handling complex feature spaces along with comprehensive computational efficiency analyses would provide valuable supplementary insights. Furthermore, future work should explore the potential of cross-lingual transfer learning and the creation of larger benchmark datasets, which would enhance the applicability and robustness of Arabic ASAG systems across different contexts.

Data availability statement

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

Author contributions

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

Funding

The author declares that no financial support was received for the research and/or publication of this article.

Conflict of interest

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

Correction note

A correction has been made to this article. Details can be found at: 10.3389/fcomp.2025.1734114.

Generative AI statement

The author declares 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

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

Aggarwal, D., Sil, P., Raman, B., and Bhattacharyya, P. (2025). "I understand why I got this grade": automatic short answer grading with feedback. arXiv [Preprint]. *arXiv:2407.12818v2*. Available online at: http://arxiv.org/abs/2407.12818 (Accessed August 4, 2025).

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* (Stroudsburg, PA: Association for Computational Linguistics), 9–15.

Badry, R. M., Ali, M., Rslan, E., and Kaseb, M. R. (2023). Automatic Arabic grading system for short answer questions. *IEEE Access* 11, 39457–39465. doi: 10.1109/ACCESS.2023.3267407

Beseiso, M., and Alzahrani, S. (2020). An empirical analysis of BERT embedding for automated essay scoring. *Int. J. Adv. Comput. Sci. Appl.* 11, 27–34. doi: 10.14569/IJACSA.2020.0111027

Chamidah, N., Yulianti, E., and Budi, I. (2023). Evaluating the impact of sentence tokenization on Indonesian automated essay scoring using pretrained sentence embeddings. *RIA* 37, 1101–1108. doi: 10.18280/ria.370502

Condor, A., and Litster, M. (2021). "Automatic short answer grading with SBERT on out-of-sample questions," in *Proceedings of The 14th International Conference on Educational Data Mining (EDM 2021)* (Providence, RI: International Educational Data Mining Society), 345–352.

Cozma, M., Butnaru, A. M., and Ionescu, R. T. (2018). Automated essay scoring with string kernels and word embeddings. arXiv [Preprint]. *arXiv:1804.07954*. Available online at: http://arxiv.org/abs/1804.07954 (Accessed August 4, 2025).

Doi, K., Sudoh, K., and Nakamura, S. (2024). Automated essay scoring using grammatical variety and errors with multi-task learning and item response theory. arXiv [Preprint]. *arXiv:2406.08817*. Available online at: http://arxiv.org/abs/2406.08817. doi: 10.5715/jnlp.32.438 (Accessed August 4, 2025).

Faseeh, M., Jaleel, A., Iqbal, N., Ghani, A., Abdusalomov, A., Mehmood, A., et al. (2024). Hybrid approach to automated essay scoring: integrating deep learning embeddings with handcrafted linguistic features for improved accuracy. *Mathematics.* 12:3416. doi: 10.3390/math12213416

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow* (Sebastopol, CA: O'Reilly Media, LLC), 541.

Ghazawi, R., and Simpson, E. (2024). Automated Essay Scoring in Arabic: A Dataset and Analysis of a BERT-based System. arXiv [Preprint]. arXiv:2407.11212. https://arxiv.org/abs/2407.11212 (Accessed July 15, 2024).

Jong, Y. J., Kim, Y. J., and, Ri, O. C. (2023). *Review of Feedback in Automated Essay Scoring. SSRN.* Available online at: https://www.ssrn.com/abstract=4462105 doi: 10.2139/ssrn.4462105(Accessed August 9,2025).

Lagakis, P., and Demetriadis, S. (2021). "Automated essay scoring: a review of the field," in *2021 International Conference on Computer, Information and Telecommunication Systems (CITS)* (Istanbul, Turkey: IEEE), 1–6. Available online at: https://ieeexplore.ieee.org/document/9618476/ doi: 10.1109/CITS52676.2021.9618476 (Accessed August 9,2025).

Li, F., Xi X, Cui, Z., Li, D., and Zeng, W. (2023). Automatic essay scoring method based on multi-scale features. *Appl. Sci.* 13:6775. doi: 10.3390/app13116775

Lotfy, N., Shehab, A., Elhoseny, M., and Abu-Elfetouh, A. (2023). An enhanced automatic arabic essay scoring system based on machine learning algorithms. *CMC.* 77, 1227–1249. doi: 10.32604/cmc.2023.039185

Mahmood, S. A., and Abdulsamad, M. A. (2024). Automatic assessment of short answer questions: Review. *J. Applied Sci. Technol. Trends* 8, 9158–9176. doi: 10.55214/25768484.v8i6.3956

Mahmoud, S., Nabil, E., and Torki, M. (2024). Automatic scoring of arabic essays: a parameter-efficient approach for grammatical assessment. *IEEE Access* 12, 142555–145568. doi: 10.1109/ACCESS.2024.3470728

Meccawy, M., Bayazed, A. A., Al-Abdullah, B., and Algamdi, H. (2023). Automatic essay scoring for arabic short answer questions using text mining techniques. *Int. J. Adv. Comput. Sci. Appl.* 14, 682–690. doi: 10.14569/IJACSA.2023.0140682

Mengxue, Z., Baral, S., Heffernan, N., Lan, A. (2022). *Automatic Short Math Answer Grading via In-context Meta-learning*, eds. A. Mitrovic and N. Bosch. Available online at: https://zenodo.org/record/6853032 (Accessed August 4, 2025).

Mizumoto, A., and Eguchi, M. (2023). Exploring the potential of using an AI language model for automated essay scoring. *Res. Methods Applied Linguist.* 2:100050. doi: 10.1016/j.rmal.2023.100050

Ouahrani, L., and Bennouar, D. (2020). "AR-ASAG An ARabic dataset for automatic short answer grading evaluation," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)* (Paris: European Language Resources Association), 2634–2643.

Paaß, G., and Giesselbach, S. (2023). *Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media*. Cham: Springer International Publishing. Available online at: https://link.springer.com/10.1007/978-3-031-23190-2 doi: 10.1007/978-3-031-23190-2(Accessed August 9,2025).

Shehab, A., Faroun, M., and Rashad, M. (2018). An automatic arabic essay grading system based on text similarity algorithms. *Int. J. Adv. Comput. Sci. Appl.* 9, 37–44. doi: 10.14569/IJACSA.2018.090337

Shermis, M. D., and Burstein, J. (2013). *Handbook of Automated Essay Evaluation.* Abingdon, Oxfordshire: Routledge. Available online at: https://www.taylorfrancis.com/books/9781136334801 doi: 10.4324/9780203122761(Accessed August 9,2025).

Su, J., Yan, Y., Fu, F., Zhang, H., Ye, J., Liu, X., et al. (2025). *EssayJudge: a multi-granular benchmark for assessing automated essay scoring capabilities of multimodal large language models*. arXiv [Preprint]. *arXiv:2502.11916*. Available online at: http://arxiv.org/abs/2502.11916 (Accessed August 4, 2025).

Sun, K., and Wang, R. (2024). Automatic essay multi-dimensional scoring with fine-tuning and multiple regression. arXiv [Preprint]. *arXiv:2406.01198*. Available online at: http://arxiv.org/abs/2406.01198 (Accessed August 4, 2025).

Wilianto, D., and Suganda Girsang, A. (2023). Automatic short answer grading on high school's e-learning using semantic similarity methods. *TEM J.* 12, 297–302. doi: 10.18421/TEM121-37

Yan, D (2020). *Handbook of Automated Scoring; Theory into Practice* (New York, NY: Taylor and Francis Group, LLC), 580.

Yang, R., Cao, J., Wen, Z., Wu, Y., and He, X. (2020). "Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking," in *Findings of the Association for Computational Linguistics: EMNLP 2020.* Online: Association for Computational Linguistics, 1560–1569. Available online at: https://www.aclweb.org/anthology/2020.findings-emnlp.141 doi: 10.18653/v1/2020.findings-emnlp.141 (Accessed August 4, 2025).

Zawacki-Richter, O., and Jung, I. (2023). *Handbook of Open, Distance and Digital Education*. Singapore: Springer Nature. Available online at: https://link.springer.com/10.1007/978-981-19-2080-6 doi: 10.1007/978-981-19-2080-6 (Accessed August 9, 2025).