Harnessing Transformers for Enhancing Arabic Educational Assessment

Emad Nabil[1,*], Mostafa Mohamed Saeed[2], Rana Reda[3], Safiullah Faizullah[4], and Wael Hassan Gomaa[5]

[1]Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia
[2]Computational Approaches to Modeling Language (CAMeL) Lab, New York University, Abu Dhabi, United Arab Emirates
[3]Digital Egypt for Investment Co., Ministry of Communications and Information Technology (MCIT), Cairo, Egypt
[4]Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia
[5] Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef, Egypt
Email: e.nabil@fci-cu.edu.eg (E.N.); mms10094@nyu.edu (M.M.S.); rana.reda@defi.com.eg (R.R.); safi@iu.edu.sa (S.F.); wael.goma@gmail.com (W.H.G.)
[*]Corresponding author
Manuscript received May 6, 2025; revised July 4, 2025; accepted July 28, 2025; published December 19, 2025

***Abstract*—This study introduces an intelligent scoring approach that leverages natural language processing and transformer-based models to evaluate student responses across various academic subjects. Given the linguistic complexity and variability of Arabic short-answer questions, the research proposes a novel grading method that moves beyond traditional techniques. Using the Cairo University Dataset a widely recognized benchmark focused on environmental science the study explores different preprocessing strategies and applies multiple transformer models. These models are integrated into a custom regression-based neural network designed specifically for Arabic short-answer grading. The proposed system achieves a Pearson correlation of 92.34%, surpassing the current state-of-the-art on the Cairo University Dataset. To evaluate generalizability, the model was also tested on the Arabic Short Answer Grading dataset, achieving an 80% Pearson correlation and outperforming existing benchmarks. These results demonstrate the approach's strong potential for educational applications, offering a scalable and fair grading solution that reduces teacher workload while maintaining assessment accuracy.**

***Keywords*—automatic scoring, Arabic short answer scoring, Natural Language Processing (NLP), transformers, Artificial Intelligence (AI) in education**

I. Introduction

In the dynamic landscape of educational assessment, the Arabic Short Answer Grading (ASAG) task stands as a pivotal and progressive method. At its core, ASAG involves evaluating student responses against a model answer. However, it transcends traditional grading approaches, embodying a sophisticated process that necessitates a comprehensive understanding of the depth, relevance, and accuracy of student responses, especially within the intricate framework of the Arabic language [1].

The significance of ASAG in modern education extends beyond conventional grading methodologies, heralding a new era of effectiveness and precision in educational contexts. ASAG holds significant value in contemporary educational environments. In an era characterized by educators' substantial teaching commitments, ASAG emerges as a notable facilitator. It notably diminishes the time and effort educators must allocate to grading, thereby liberating resources for teaching and curriculum enrichment endeavors. Additionally, this system accelerates the grading process, providing timely feedback crucial for students' academic progress and comprehension [2, 3].

The swift and accurate evaluation of short answers, particularly within the nuanced domain of the Arabic language, is essential for maintaining educational excellence and assessment standards. In the era of digitization, the frontiers of Machine Learning (ML) and Deep Learning (DL) stand poised to revolutionize diverse domains, including education. The integration of ML and DL methodologies into the Automated Short Answer Grading (ASAG) task heralds a new era characterized by heightened efficiency and precision.

These technologies leverage sophisticated algorithms and models endowed with the ability to comprehend, interpret, and assess student submissions in a manner reminiscent of human cognition, albeit with superior velocity and consistency. Within the realm of Deep Learning for Natural Language Processing (NLP), notable progress has been achieved, propelled by the emergence of advanced architectures such as Recurrent Neural Networks (RNN) [4], Long Short-Term Memory (LSTM) networks [5], Bidirectional LSTM (BiLSTM) configurations [6], and Gated Recurrent Units (GRU) [7]. These architectures have played a pivotal role in augmenting the efficacy of language processing endeavors [8, 9], offering distinctive capabilities in capturing sequential patterns and dependencies within textual data. Their deployment in ASAG applications for the Arabic language holds immense potential, facilitating a more nuanced and contextually aware evaluation of student responses.

At the core of this technological revolution are the Transformers [10], a class of models in Deep Learning distinguished for their remarkable aptitude in grasping the semantic and syntactic intricacies embedded within textual data. Transformers are particularly suited for this task as they capture contextual and semantic meaning, allowing for robust evaluation of answers even when students use diverse or unconventional phrasing. Their integration into the ASAG domain marks a pivotal juncture, particularly significant for the Arabic language, celebrated for its intricate and multifaceted linguistic structure. Harnessing sophisticated architectures, Transformers demonstrate unparalleled proficiency in unraveling the subtle connotations interwoven within student responses, thereby establishing a grading framework distinguished by its swiftness, effectiveness, and profound comprehension of the language's intricacies.

This paper aims to scrutinize the transformative influence exerted by advanced Deep Learning (DL) technologies,

encompassing RNN, LSTM, BiLSTM, and Transformers, alongside select Natural Language Processing (NLP) techniques, within the Automated Short Answer Grading (ASAG) task. It delves into the way these models redefine the fundamental tenets of language comprehension and assessment within the domain of Arabic education, furnishing an unprecedented level of accuracy and insight in the grading process.

This study addresses the challenge of automatically grading Arabic short answers by leveraging recent advancements in deep learning. The key contributions of our proposed approach are summarized as follows:

- We present a robust and efficient method for the automatic grading of Arabic short answers, utilizing the advanced capabilities of pre-trained deep learning transformer models.
- The study investigates the impact of various text preprocessing techniques on grading performance and evaluation metrics.
- An ensemble approach is employed, combining multiple pre-trained embedding models to capture diverse semantic representations.
- The method is based on the hypothesis that integrating features from multiple pre-trained models enhances both grading accuracy and computational efficiency.
- The integration strategy results in improved performance while reducing computational overhead.
- This work represents a significant advancement in the field of Arabic automatic short answer scoring.

The rest of this paper is organized as follows: Section II presents an in-depth review of current automatic short-answer grading systems. Section III introduces the Arabic datasets used in this study, along with representative examples and a detailed explanation of the proposed methodology. Section IV describes the experiments conducted and reports the results. Section V provides a discussion, offering insights and interpretations of the findings. Finally, Section VI concludes the paper by summarizing the key outcomes and suggesting potential directions for future research.

II. Literature Review

The subsequent literature review delves into the body of literature concerning the task of short answer grading, with a specific focus on contributions and advancements pertinent to the Arabic language. By synthesizing previous research efforts, this literature review aims to provide a comprehensive overview of the progress made in the field of Arabic short answer grading. It not only highlights the challenges encountered but also elucidates the innovative solutions and methodologies proposed by researchers to overcome these obstacles.

Magooda *et al.* [11] tackled a short answer grading challenge by framing it as a similarity problem rather than relying on templates or linguistic analyses. They employed a diverse array of similarity measures, including string-based [12], knowledge-based, corpus-based, and word vector representations. The study delineates preprocessing, similarity measurement, and scaling modules, utilizing Support Vector Regression (SVR) [13] to map similarity scores to grades spanning from 0 to 5. Evaluation across various datasets, such as the Texas computer science dataset Extended Texas computer science dataset, Cairo University Dataset (CUD), and SemEval 2013 dataset, showcases the system's effectiveness. Notably, the results encompass a Pearson's Correlation of 0.84 and an RMSE of 0.89 for the CUD, a Pearson's Correlation of 0.59 for the Texas dataset, and a Pearson's Correlation of 0.550 accompanied by an RMSE of 0.91 for the Extended Texas dataset.

Ouahrani and Bennouar [14] present AR-ASAG, an innovative Arabic dataset tailored for automated short answer grading, comprising 2133 pairs of model answers and student responses. This resource accommodates various formats to cater to diverse applications. The study employs an unsupervised corpus-based [15] grading technique for Arabic, leveraging the COALS algorithm to establish semantic associations between words. The method amalgamates a summation vector model, term weighting, and common words to gauge alignment between the teacher's model answers and student responses. The paper delineates three primary approaches: employing the COALS semantic space approach with pre-processing, introducing term weighting to identify discriminative words, and exploring the impact of stemming. The findings reveal a Pearson's Correlation of 0.7037 and an RMSE of 1.0454 for AR-ASAG, alongside a Pearson's Correlation of 0.7220 and an RMSE of 0.76 for this dataset.

Gomaa and Fahmy [16] concentrate on scoring Arabic short answers by juxtaposing various string-based and corpus-based similarity measures, evaluating their amalgamation, and providing immediate constructive feedback to students. The methodology encompasses the assessment of fourteen string-based similarity algorithms employing both holistic and partitioning models and implementing strategies like Raw, Stop, Stem, and StopStem to handle student answers comprehensively and partially. The approaches leverage techniques such as stop word elimination and stemming, utilizing the Arabic Stemmer from the Information Science Research Institute (ISRI) [17] for stemming non-stop words. The study utilizes the Philosophy Dataset, achieving a Pearson's Correlation of 0.862 and an RMSE of 0.76 for this dataset.

Nael *et al.* [18] introduce a deep learning-driven methodology for assessing Arabic short answers. The approach involves preprocessing steps, encompassing cleaning, removal of numbers, symbols, and artifacts from translation, along with lemmatization to reduce vector size in non-transformer fine-tuning experiments. The research explores diverse models, with a specific focus on Arabic, to ensure equitable comparison. Two baseline experiments are conducted: a reference-based method utilizing cosine similarity with normalized scores, and a TF-IDF vectorization coupled with SVM classification [19], optimizing hyperparameters to enhance Quadratic Weighted Kappa (QWK) scores [20]. Initially, default models such as RNN, LSTM, and BI-LSTM are employed, followed by fine-tuning of BERT and ELECTRA [21]. The evaluation utilizes the AR-ASAG dataset, revealing varying effectiveness across different models, with scores ranging from 0.46 for the reference-based model to 0.78 for the ELECTRA model.

Badry *et al.* [22] present an automated grading methodology tailored for Arabic short-answer questions. The methodology encompasses preprocessing steps such as

tokenization, stop word removal and lemmatization. The core implementation revolves around the application of Latent Semantic Analysis (LSA) [23] between student and reference answers. The LSA process entails term weight representation (with the option of local or local and global weights) and Singular Value Decomposition (SVD) calculations [24]. Subsequently, semantic scoring is conducted using cosine similarity to assess the semantic alignment between student and reference responses. The research leverages the AR-ASAG dataset for evaluation, identifying the optimal model as the one utilizing both local and global weights, achieving a 0.745 Pearson correlation.

While previous research has made notable contributions to Arabic short answer grading, several limitations persist. First, many studies primarily rely on classical similarity-based methods or individual deep learning models, without fully exploring the capabilities of modern transformer-based architectures tailored for Arabic. Second, few works have investigated the combined impact of multiple preprocessing techniques on grading accuracy, often relying on fixed or minimal text cleaning pipelines. Third, the potential of ensemble modeling (particularly through the integration of multiple sentence embedding models) remains underexplored in Arabic contexts. Moreover, some studies use datasets with limited diversity or scope, affecting the generalizability of the results.

In contrast, our work addresses these limitations by:

- Utilizing an ensemble of seven Arabic-specific pre-trained transformer models for richer semantic representation.
- Conducting extensive experiments across six preprocessing pipelines, including combinations, to optimize data preparation.
- Evaluating the proposed method on two benchmark Arabic datasets (CUD and AR-ASAG), demonstrating both high accuracy and strong generalization capabilities.

Salam *et al.* [25] introduce a pioneering method for automating the grading of Arabic short answer questions. The research leverages a proprietary dataset collected from various schools in Qalyubia, encompassing responses from 300 students across 50 questions, totaling 15,000 submissions. Evaluation was conducted using a manual grading system with scores ranging from 0 to 5. The proposed model entails three main steps. Firstly, preprocessing is performed, involving the removal of punctuation and special characters, word tokenization, stop word elimination and word stemming to their root forms. Secondly, feature extraction is pursued utilizing the Bag of Words (BOW) technique, which transforms pre-processed words into numerical vectors. Finally, the model incorporates a Long Short-Term Memory (LSTM) layer combined with the Grey Wolf Optimization (GWO) technique [26]. This optimization fine-tunes the values of dropout layers and the dropout rate within the LSTM layer itself, ultimately assigning a grade to the input sentence. Notably, the approach yields promising performance metrics, including a Root Mean Squared Error (RMSE) of 0.807, R-Square of 0.826, and a Pearson's correlation coefficient of 0.909.

Magooda *et al.* [27] introduce a novel scoring system named Ans2vec, which operates on the CUD. This model comprises three distinct processes. Firstly, both student and model answers are transformed into vectors using the pre-trained Skip-Thought vectors model. Subsequently, component-wise product operations and absolute differences are applied between the two vectors, resulting in concatenated vectors. These concatenated vectors serve as input for the final phase of the model, wherein a Logistic Linear regressor is employed to generate the ultimate grade output. The model achieves commendable performance, evidenced by a Pearson's correlation coefficient of 0.79 on CUD Dataset and a Pearson's correlation coefficient of 0.63 on Texas Dataset.

Abdeljaber [28] presents an innovative scoring methodology devised for automated Arabic short answer grading, leveraging the concept of Longest Common Contiguous Subsequence (LCCS) [29] alongside Arabic Word-Net [30]. The study utilizes a proprietary dataset meticulously curated from an AI course at the computer science department of Prince Sattam Bin Abdulaziz University in Saudi Arabia. This dataset comprises 330 responses from 33 student volunteers across 10 questions, each associated with model answers. The responses underwent annotation by two human experts, and the average of their scores, ranging from 0 to 10, was considered. The model's methodology entails two primary steps. Firstly, preprocessing tasks are conducted, including word tokenization, handling punctuation and special characters, stop word removal, lemmatization, and synonym retrieval. The latter is facilitated using Arabic WordNet, wherein synonyms from the model answer are replaced with synonyms from the student answer. Subsequently, the Longest Common Contiguous Subsequence (LCCS) is employed to derive the final grade. This approach yields promising results, with a Root Mean Squared Error (RMSE) of 0.81 and a Pearson's correlation coefficient of 0.94. As shown in Table 1, it presents a comprehensive summary of the results from the entire literature review, showcasing the datasets examined in previous studies along with their respective correlation scores.

Several systems have approached Short Answer Grading (SAG) [31–35] as a text similarity task, leveraging the strong conceptual overlap between the two. In these systems, the student answer is evaluated by computing its similarity to a reference (model) answer, using a range of linguistic features. The final grade is derived from the degree of similarity between the two texts.

A. Short Answer Grading Taxonomy

Fig. 1 presents a detailed diagram elucidating the components and methodologies utilized for grading short answers, potentially within the realm of educational technology. The diagram is segmented into five primary sections: Pre-processing Techniques, Text Vectorization Techniques, Text Similarity Measures, ML and DL Models, and Evaluation Metrics.

1) Preprocessing techniques

The preprocessing task focuses on preparing text data before analysis [36], a crucial step in cleansing and standardizing the text to enhance its readability and make it more suitable for machine learning models. Key techniques like tokenization, stemming, lemmatization, and normalization are applied to refine the text by breaking it down into manageable units, reducing words to their base or root forms, removing irrelevant characters and words, and

ensuring consistency throughout the dataset. The pre-processing techniques are essential for preparing the text for effective analysis. These methods include:

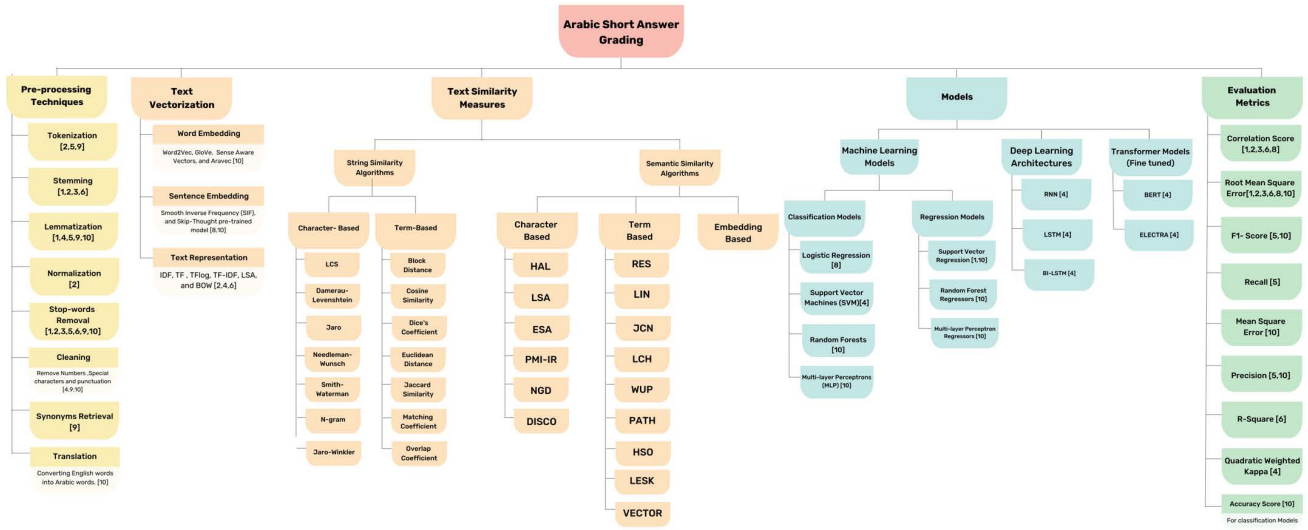

Fig. 1. Taxonomy of the short answer grading techniques.

Table 1. Comparison of literature review

Ref.	Dataset	Method/Approach	Correlation Score
[31]	Texas Dataset	Vectorized-SVR	0.59
[32]		IAA	0.644
		Text-to-Text	0.509
[27]		Ans2vec	0.63
[31]	Extended Texas Dataset	Vectorized-SVR	0.550
[33]		IAA	0.586
		SVR	0.518
		SVM	0.464
[15]		Text-to-Text Similarity	0.504
[34]	Cairo University Dataset	IAA	0.86
		Arabic system	0.73
[35]		Arabic vectors	0.82
[31]		Basis System	0.78
		Vectorized-SVR	0.84
[27]		Ans2vec	0.79
[14]	AR-ASAG	-	0.70
[18]		ELECTRA	0.78
[22]		-	0.745
[16]	Philosophy Dataset	-	0.86
[25]	dataset of 300 responses from students in Qalyubia to 50 questions	(private dataset)	0.90
[28]	dataset of 330 responses from 33 student across 10 questions	(private dataset)	0.94

- Tokenization: Dividing text into individual words or phrases.
- Stemming: Reducing words to their root form.
- Lemmatization: Similar to stemming but ensures the root word belongs to the language.
- Normalization: Standardizing text.
- Stop-word Removal: Eliminating common words that may not contribute to meaning.
- Cleaning: Removing numbers, special characters, and punctuation.
- Synonyms Retrieval: Finding synonyms to expand understanding.
- Translation: like converting English words to Arabic words for Arabic language processing.

2) Text vectorization

Once the text has undergone pre-processing, the next step is to convert it into a numerical format that machine learning algorithms can process. This is where Text Vectorization becomes essential. This section delineates various methods for transforming text into numerical data, including Word Embedding, Sentence Embedding, and different Text Representation techniques such as TF-IDF and Bag of Words (BOW). These methods facilitate the capture of the semantic and syntactic essence of the text in a format that algorithms can effectively analyze. Text vectorization techniques include the following:

- Word Embedding: Utilizing methods like Word2Vec, Aravec [37], GloVe [38], and Sense Aware Vectors.
- Sentence Embedding: Incorporating techniques such as Smooth Inverse Frequency and Skip-Thought pre-trained models [39].
- Text Representation: Encompassing methods like TF, TF-IDF, LSA, and Bag of Words (BOW) [39].

3) Text similarity measures

Fig. 1 illustrates various text similarity techniques used to compare a student's answer with a set of reference answers or key concepts. These techniques are categorized into two main approaches: exact string matching (String Similarity Algorithms) and meaning-based comparison (Semantic Similarity Algorithms) [40, 41]. They range from straightforward character- or term-based comparisons to more advanced algorithms that analyze semantic relationships between words or phrases. This enables the system to assess the depth of the student's understanding and the accuracy of their responses more effectively.

a) String Similarity encompasses

- Character-based methods such as LCS, Damerau-Levenshtein, Jaro, Needle-man-Wunsch, Smith-Waterman, N-gram, and Jaro-Winkler.
- Term-based approaches including Block Distance, Cosine Similarity, Dice's Co-efficient, Euclidean Distance, Jaccard Similarity, Matching Coefficient, Overlap

Coefficient, as well as additional algorithms like HAL, LSA, and ESA for char-acter-based, and PMI-IR, NGD, and DISCO for term-based comparisons.

b) Semantic Similarity

Semantic Similarity Algorithms encompass various types ranging from simpler ones like RES, LIN, and JCN to more complex ones like WUP, PATH, HSO, LESK, and VECTOR [42], which may entail a deeper understanding of the text.

4) Models

This section enumerates the various machine learning models suitable for grading short answers, encompassing both traditional approaches like Classification and Regression Models, as well as more advanced Deep Learning Architectures and Transformer Models. These models are trained to comprehend the language context, evaluate answer quality, and determine grades based on the degree of alignment between student responses and expected answers or the coverage of required concepts. Fig. 1 contains the following models' categories.

- Classification Models: Including Logistic Regression, Support Vector Machines, Random Forests, Multi-layer perceptron [43].
- Regression Models: Encompassing Support Vector Regression, Random Forest Regressors, and Multi-Layer Perceptron Regressors.
- Deep Learning Models: Such as RNN, LSTM, and BI-LSTM which offer enhanced capabilities in understanding language context.
- Transformer Models: Represented by fine-tuned versions of BERT [44], ELECTRA, which demonstrate advanced proficiency in contextual comprehension within a language.

5) Evaluation metrics

Finally, the effectiveness of the grading models is evaluated using a range of performance metrics. As shown in Fig. 1, this part highlights techniques for assessing model performance to ensure grading accuracy and fairness. Metrics such as Correlation Score [45], Root Mean Square Error (RMSE), F1-score, and others are employed to analyze various aspects of the models, including their accuracy, reliability, and ability to replicate human grading decisions. The Evaluation Metrics part provides a detailed list of the metrics utilized to measure model performance, which includes the following: Correlation Score, Root Mean Square Error, F1-Score, Recall, Mean Square Error, Precision, R-Square, Quadratic Weighted Kappa and Accuracy Score for classification models.

The five parts of Fig. 1 outline a sophisticated, multi-layered approach to automating the grading of short answers written, ensuring thorough processing and evaluation of student responses from initial text preparation to final grading.

III. Datasets and Methodology

A. Datasets

The Cairo University Short Answer Grading dataset emerges as an Arabic dataset specifically tailored for benchmarking the automatic grading of Arabic short answers [16]. Importantly, the responses in this dataset were written by actual students, not synthetically generated, ensuring that the model is trained and evaluated on authentic, real-world data. This corpus encompasses a comprehensive range of questions extracted from an Egyptian official curriculum for Environmental Science (ES). The data set features 61 distinct questions, each accompanied by 10 student responses, culminating in a total of 610 answers. A key feature of this corpus is its inclusion of a diverse collection of student answers along with their respective grades, which were meticulously assessed by two annotators.

These annotators assigned scores ranging from 0 to 5, with the data set providing an average score for each response. The corpus is designed to support four distinct types of short-answer questions: definitions of scientific terms, explanations, explorations of consequences, and reasons behind certain phenomena.

Fig. 2 illustrates a sample of the dataset utilized in this study, showcasing two examples. Each example comprises a question, responses from two students, the model or reference answer, and the grade assigned to the student's responses relative to the model answer. Fig. 3 shows the distribution of the grades across the CUD.

	Question	Student Answer	Model Answer	Grade
Original Arabic Text	بم تفسر للبيئة المصنوعة أثرها السيئ على البيئة الطبيعية.	لأن الإنسان يضر بالبيئة الطبيعية بسبب مخلفات المصانع واستخدام المبيدات وقطع الاشجار و البناء على الأراضي الزراعية	لأن البيئة المصنوعة قد تسئ للبيئة الطبيعية عندما يسمح الإنسان لمخلفات الصناعة بتلويث البيئة ، أو عندما يتخذ قرارا باستخدام مبيد ما ،دون دراسة كافية لأثاره السلبية	4.5
Translated English Text	How does the manufactured environment explain its negative impact on the natural environment?	Because humans harm the natural environment due to factory waste, the use of pesticides, cutting trees, and building on agricultural lands.	Because the manufactured environment may harm the natural environment when a person allows industrial waste to pollute the environment, or when he makes a decision to use a pesticide without adequate study of its negative effects.	
Original Arabic Text	عرف الإيكولوجيا	تعنى علم دراسة مكان المعيشة أو يقصد بها العلاقات المتبادلة بين الأحياء والبيئة	الدراسة التي تتناول جوانب الطبيعة بما يحدده حياة الكائن وكيفية استخدامه لمكونات البيئة	1.5
Translated English Text	Define ecology	It means the science of studying the living space or the mutual relationships between living things and the environment.	The study deals with aspects of nature, the life of an including what determines organism and how it uses the environment. components of the	

Fig. 2. Samples of the CUD.

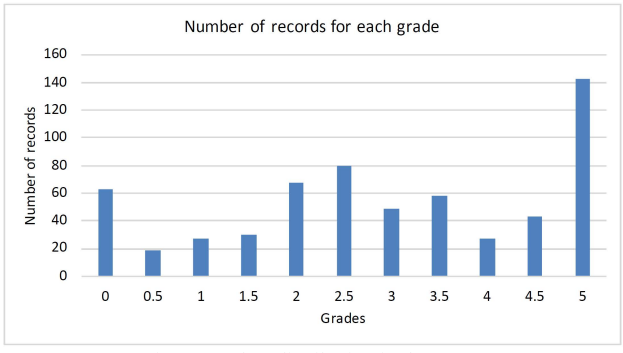

Fig. 3. Grades' distribution in the CUD.

Fig. 4 illustrates the average word count of student answers for each model answer in the CUD. On the X-axis, unique IDs represent different model answers, while the Y-axis shows the corresponding average word count in student responses. Each bar signifies the average length of each specific model answer, providing a clear visual comparison. The graph is designed for easy interpretation, with rotated X-axis labels for better readability. It is a useful tool for analyzing the variation in response lengths across various students' answers.

To ensure the generality of the proposed model, in addition to evaluating it on the CUD, we also validated it on the AR-ASAG dataset [14]. The AR-ASAG dataset, designed for automatic short answer grading, comprises 48 questions focused on cyber crimes and is divided into five categories:

defining, explaining, identifying consequences, justifying, and distinguishing differences. It contains 2,133 student model-answer pairs, each assigned two marks, along with an average gold score ranging from 0 to 5, which is used for prediction. The dataset includes sentence lengths ranging from 1 to 76 words, with an average length of 20 words, offering a diverse set of short answers for grading and assessment.

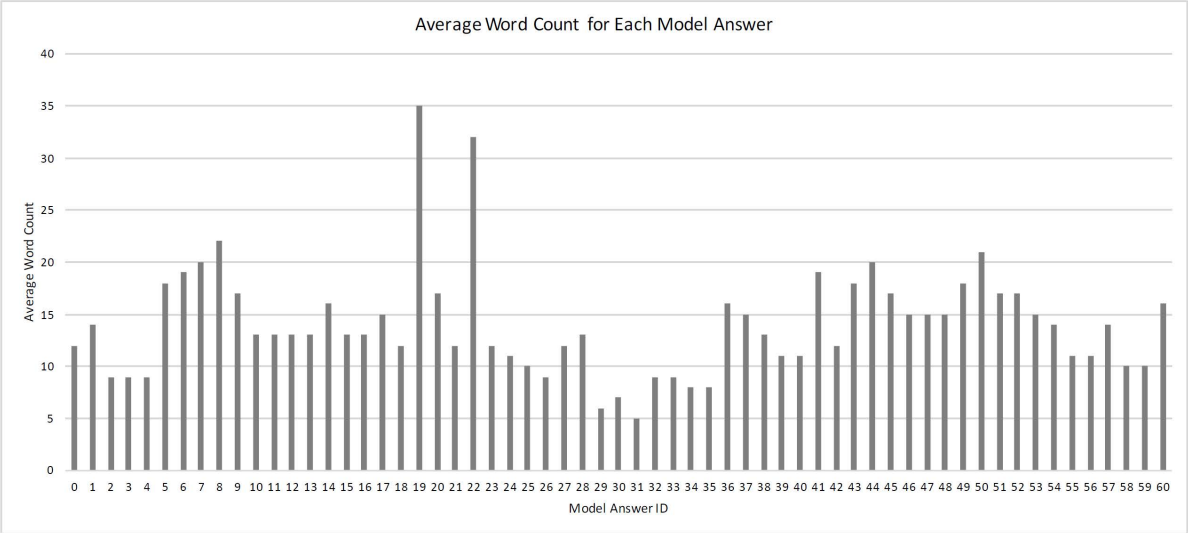

Fig. 4. Average word count per model answer in the CUD.

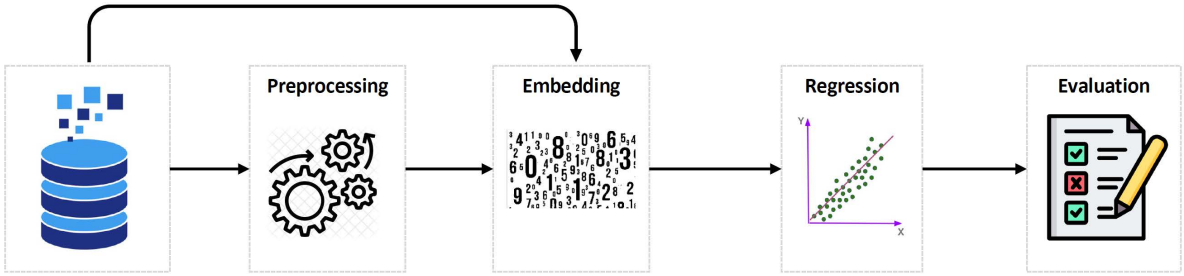

Fig. 5. High-level conceptual overview of the proposed methodology.

B. Methodology

The approach is organized into five sequential phases, as illustrated in Fig. 5, each carefully designed to contribute to the development of the Arabic Short Answer Grading (ASAG) system. The phases are preprocessing, embedding, regression, and evaluation.

1) Pre-processing phase

The preprocessing phase began with a thorough inspection of the raw dataset to identify and correct inconsistencies, missing values, and anomalies. The data was then converted into a standardized format to ensure quality and readiness for analysis. We applied various preprocessing techniques, including normalization to standardize text input, and both stemming and lemmatization to reduce words to their base or root forms while preserving meaning. After evaluating each technique individually, we further explored the synergistic effects of combining multiple preprocessing methods. This step aimed to identify the most effective preprocessing pipeline to enhance model performance and consistency.

2) Embedding Phase

This phase is pivotal in capturing the linguistic nuances of the Arabic language:

- Selection of Arabic Embedding Models: We explored various state-of-the-art Arabic embedding models, each known for its unique approach to capturing semantic features.
- Embedding Experiments: Each selected model underwent rigorous testing with our dataset. This includes evaluating their effectiveness in capturing contextual meanings and subtleties of the Arabic language.
- Models Comparison: We compared the effectiveness of each model on our dataset using the adopted evaluation metric.

3) Regression Grading Model Phase

This phase focused on the development and optimization of a Regression Neural Network model:

Design and Implementation: Utilizing PyTorch, we designed a neural network tailored to our specific requirements. The initial version is designed based on existing literature and best practices in the field.

- Hyperparameter Tuning: To optimize the regression-based neural network, we conducted a series of controlled experiments aimed at fine-tuning critical hyperparameters. The model was trained using the AdamW optimizer and the Mean Squared Error (MSELoss) function, which is well-suited for continuous regression tasks. A learning rate of 1e-5 was chosen to ensure stable convergence, and the training process was carried out over 100 epochs. Both the training and validation phases employed a batch size of 16, balancing computational efficiency with model performance. This configuration was selected based on empirical evaluation and its ability to yield consistently high Pearson correlation scores on the validation set.

- Iterative Refinement: The network is iteratively refined based on performance metrics, with a focus on improving accuracy and reducing overfitting.

In addition to our primary efforts, we also explored the possibility of framing the task as a classification problem. However, this approach yielded unsatisfactory results, rendering it a less promising avenue for the research objectives.

4) Evaluation phase

The final stage involved a comprehensive evaluation of the proposed model:

- Implementation of Pearson Correlation: The Pearson correlation coefficient is the primary metric for evaluating the proposed model's performance. This choice is aligned with standard practices in Automated Short Answer Grading.
- Comparative Analysis: The performance of our model is compared against existing benchmarks and models in the field to assess its efficacy and potential for real-world application.

5) Detailed methodology

According to Fig. 1, which delineates a comprehensive list of techniques implemented across various phases of the task at hand, spanning pre-processing, text vectorization, text similarity measurement, utilized models, and evaluation metrics, our research is positioned to not only re-implement some of the techniques previously explored but also introduce unique contributions. Specifically, our aim is to experiment with novel combinations within the pre-processing phase and explore the effectiveness of employing diverse models to address this task. This strategic approach is expected to enhance the existing methodology by potentially revealing more efficient and insightful ways of tackling the inherent challenges of the task.

Fig. 6 delineates a detailed methodology for the proposal. Initially, the data is divided into a training set comprising 80% of the data and a testing set comprising 20%. Subsequently, in the pre-processing phase, an exhaustive exploration of data handling techniques is undertaken to optimize the quality and relevance of the raw data for subsequent analysis. The process begins with an examination of the efficacy of utilizing the raw data in its unaltered state, serving as a baseline for subsequent comparisons. This baseline, depicted in passing through two of the diagrams, transitions directly from the Splitting phase to the Embedding phase without applying any pre-processing techniques. This step is crucial for recognizing the potential of data transformation techniques to enhance the interpretability and consistency of the CUD. Following the establishment of the baseline, various standard pre-processing methods are individually applied, as depicted in pass one of Fig. 6. This iterative approach allows for a comprehensive evaluation of the impact of each pre-processing technique on the dataset.

The first pre-processing technique was normalization, a technique aimed at scaling individual data elements to ensure uniformity in the range and distribution of data values.

This approach was particularly beneficial in mitigating the effects of outliers and varying scales among different features, thereby facilitating a more coherent analysis framework. Subsequently, we shifted our focus to lemmatization, a linguistic pre-processing method designed to reduce words to their base or dictionary form.

By transforming various inflected forms of a word into a common base form, lemmatization helped streamline the textual data, thus enhancing the consistency and comparability of linguistic elements within our dataset.

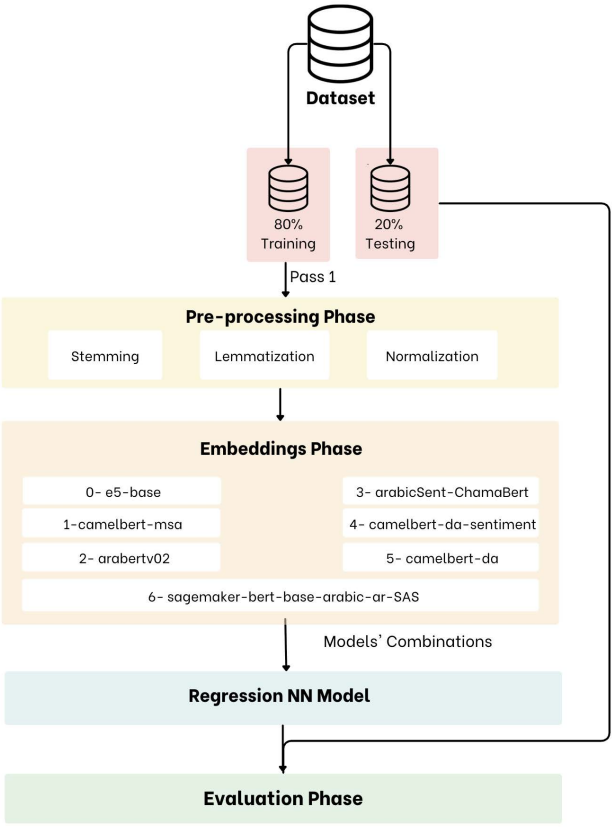

Fig. 6. Detailed illustration of the adopted methodology.

Additionally, we explored the application of stemming as part of our linguistic pre-processing pipeline. Following individual applications, we also experimented with sequential combinations of these techniques to assess their cumulative effects. Specifically, we applied normalization followed by lemmatization, and separately, stemming followed by normalization, to examine how the order of operations influences model performance.

In the embedding phase, our research embarked on an ambitious exploration by delving into seven distinct embedding models, as outlined in Table 2. Each model underwent meticulous analysis to evaluate its individual performance and characteristics. The uniqueness of each model lies in its ability to represent textual data in a multi-dimensional space, capturing various semantic and syntactic nuances. Following the individual assessments, we embarked on a comprehensive investigation of potential combinations of these embeddings. This involved embedding a single sentence using all of the seven models, thereby yielding seven distinct embedding representations. Our goal was to experiment with various combinations of these embeddings to uncover the most effective amalgamations. This exhaustive combinatorial approach was based on the hypothesis that each embedding has its own power and that combining two or more embeddings could unveil nuances in textual data, thereby enhancing the overall analytical robustness.

After the embedding phase, we proceeded to develop a new regression Neural Network (NN). The chosen detailed architecture, as depicted in Fig. 6, was selected for its potential to effectively utilize the pre-processed and embedded data, thereby serving as a robust foundation for our neural network model.

This statistical measure was selected for its efficacy in quantifying the degree of linear correlation between the outputs of our neural network model and the expected results. By employing Pearson correlation as our primary evaluative metric, we aimed to rigorously assess the predictive accuracy and reliability of the proposed approach, thereby ensuring a comprehensive and quantitatively robust evaluation of its performance.

Table 2. Pre-trained embedding models

Model #	Model Name
0	intfloat/multilinguale-e5-base [46]
1	CAMeL-Lab/bert-base-arabic-camelbert-msa [47]
2	audmindlab/bert-base-arabertv02-twitter [48]
3	ChaimaaBouafoud/arabicSent-ChamaBert [48]
4	CAMeL-Lab/bert-base-arabic-camelbert-da-sentiment [47]
5	CAMeL-Lab/bert-base-arabic-camelbert-da [47]
6	Osaleh/sagemaker-bert-base-arabic-ar-SAS [49]

IV. Experiments and Results

This section outlines the experimental framework used in this research, providing a comprehensive overview of the various pre-processing methods applied, along with an exploration of the combinations of embedding models and a discussion of the results.

Experiments were conducted on a high-performance computing workstation equipped with an Intel Xeon W-3235 CPU (12 cores, 3.3 GHz), 128 GB RAM, and an NVIDIA RTX 6000 GPU (24 GB). The implementation environment included Python, PyTorch, and the Hugging Face Transformers library for model development and training. Data preprocessing and handling were performed using pandas and NumPy, while performance evaluation metrics (e.g., Pearson correlation) were computed using scikit-learn. This environment ensured efficient experimentation, scalability, and reproducibility.

Six distinct pre-processing techniques were applied to the CUD, inclusive of experimental runs on raw data. For each pre-processing technique, a systematic investigation was conducted using seven pre-trained embedding models.

This involved an exhaustive examination starting from the individual application of each model, progressively extending to the integration of all seven models. The experimental structure was methodically designed to evaluate the performance of single model configurations, followed by incremental combinations of two, three, four, five, six, and ultimately, seven models.

Consequently, the resultant data from these experiments are systematically presented in tables, each consisting of seven rows. These rows correspond to the peak correlation scores achieved in each combination associated with a specific pre-processing technique. This structured presentation facilitates a clear and comprehensive understanding of the performance dynamics across different pre-processing and model combination scenarios. Therefore, there are 127 trials have been implemented in each experiment from the upcoming experiments.

A. Lemmatization as a Preprocessing Technique

As shown in Table 3, This experiment meticulously investigates the impact of lemmatization as a pre-processing technique on the performance of combined pre-trained models. The objective was to ascertain how different combinations of embedding models could influence correlation scores, which are pivotal in determining the effectiveness of Natural Language Processing (NLP) tasks, thereby improving the model's ability to understand the context and semantics of text data.

The analysis was structured around comparing the top seven correlation scores achieved in each combination phase of the pre-trained models. This rigorous evaluation led to the identification of a combination of four embedding models that achieved an unprecedented high correlation score of 90.85%. This result underscores the efficacy of lemmatization in enhancing the model's linguistic comprehension and predictive accuracy.

The embedding models, which represent words or phrases in vectors of real numbers, were meticulously selected and combined to optimize performance. The high correlation score indicates a strong relationship between the models' predictions and the expected outcomes, demonstrating the success of this approach in capturing semantic relationships and contextual meanings.

This experiment serves as a compelling case for the strategic integration of lemmatization in the pre-processing stage of NLP tasks. It highlights the significance of pre-processing techniques in refining input data, thereby maximizing the potential of machine learning models to deliver more accurate and reliable outcomes.

Table 3. Experimental results after applying lemmatization

Applied Models	Train Loss	Testing Loss	Train Correlation	Testing Correlation
6	0.1036	0.5960	0.9808	0.8870
4–6	0.1005	0.5536	0.9833	0.8937
1–5–6	0.0572	0.5247	0.9898	0.9008
1–2–3–6	0.0512	0.5051	0.9912	0.9083
1–2–4–5–6	**0.0522**	**0.5185**	**0.9907**	**0.9085**
0–1–2–3–5–6	0.0461	0.5036	0.9915	0.9072
0–1–2–3–4–5–6	0.0570	0.5049	0.9908	0.9070

B. Stemming for Text Standardization

As shown in Table 4, Experiment 2 pivots from lemmatization to stemming. The goal was to explore how this difference impacts the correlation scores when combining pre-trained models. The structured analysis focused on the top seven correlation scores obtained from various combinations of embedding models, with an emphasis on understanding the influence of stemming. The highest correlation score recorded was 90.56%, achieved with a combination of four embedding models.

This slightly lower score, compared to the lemmatization approach, might reflect the limitations of stemming in accurately capturing the semantic nuances of language.

Despite this, the achievement of a high correlation score close to 90.56% is indicative of the robustness of the selected embedding models and their capability to work effectively even when preprocessing introduces some level of approximation.

Table 4. Experimental results after applying stemming

Applied Models	Train Loss	Testing Loss	Train Correlation	Testing Correlation
6	0.1067	0.5939	0.9802	0.8919
1–6	0.0594	0.5746	0.9897	0.8947
1–4–6	0.0678	0.5403	0.9905	0.9044
1–4–5–6	0.0629	0.4990	0.9907	0.9052
0–2–4–5–6	**0.0772**	**0.5571**	**0.9887**	**0.9056**
0–1–2–4–5–6	0.0590	0.5305	0.9905	0.9021
0–1–2–3–4–5–6	0.0612	0.5540	0.9900	0.8967

C. Normalization Impact on Grading Accuracy

As shown in Table 5, This experiment shifts the focus to normalization, which can significantly influence the input data quality for NLP models. The objective was to assess how normalization, when used as the sole preprocessing step, affects the correlation scores of combined pre-trained models.

Table 5. Experimental results after applying normalization

Applied Models	Train Loss	Testing Loss	Train Correlation	Testing Correlation
1	0.1575	0.5558	0.9724	0.9005
1–6	0.0776	0.5068	0.9878	0.9082
1–4–6	0.0692	0.5100	0.9876	0.9076
1–2–4–6	**0.0680**	**0.4870**	**0.9881**	**0.9093**
0–1–2–5–6	0.0712	0.5260	0.9896	0.9074
0–1–2–4–5–6	0.0567	0.5139	0.9896	0.9071
0–1–2–3–4–5–6	0.0522	0.5771	0.9903	0.8935

The analysis detailed the top seven correlation scores, highlighting the achievement of a 90.93% correlation score through the combination of four embedding models. This result is slightly higher than those achieved with lemmatization and stemming, suggesting that normalization, by standardizing text data, may facilitate more effective learning and interpretation by the models.

Normalization's strength lies in its simplicity and efficiency in preparing text data, which can lead to improved model performance by reducing the variability and complexity of the data.

D. Combined Lemmatization and Normalization

As shown in Table 6, Experiment 4 explores the combined effect of lemmatization and then normalization as preprocessing techniques. This dual approach aimed to leverage the benefits of both techniques: the semantic precision of lemmatization and the data consistency provided by normalization. The hypothesis was that this combination could lead to a synergistic improvement in the models' performance.

Table 6. Experimental results after applying lemmatization and normalization

Applied Models	Train Loss	Testing Loss	Train Correlation	Testing Correlation
1	0.1425	0.6273	0.9737	0.8927
1–5	0.1192	0.6211	0.9782	0.8963
1–2–5	0.0720	0.6588	0.9876	0.8978
0–1–4–5	0.0872	0.6305	0.9852	0.8977
1–2–3–5–6	0.0576	0.6028	0.9905	0.8969
0–1–2–3–5–6	**0.0596**	**0.5472**	**0.9900**	**0.8985**
0–1–2–3–4–5–6	0.0616	0.5668	0.9903	0.8943

However, the experiment recorded the highest correlation score of 89.85%, realized through the combination of four embedding models. This score, slightly lower than those achieved in the single-preprocessing technique experiments, suggests that the interaction between lemmatization and normalization might not always produce additive benefits. It highlights the complexity of NLP tasks and the need for careful consideration of how different preprocessing techniques interact with each other.

E. Re-Evaluation of Lemmatization Performance

As shown in Table 7, In a revisited approach towards leveraging lemmatization, experiment 5 aimed to further investigate its impact on the performance of combined pre-trained models, possibly under different conditions or with slight modifications to the model combinations. This experiment mirrors the structure of Experiment 1 but with an outcome that produced a correlation score of 90.59%.

This slight variation in the achieved score could be attributed to differences in the experimental setup, such as modifications in the model combinations or adjustments in the parameters used. The consistency of high correlation scores in experiments utilizing lemmatization underscores its value in preprocessing for NLP tasks.

Table 7. Experimental results after applying stemming and normalization

Applied Models	Train Loss	Testing Loss	Train Correlation	Testing Correlation
6	0.1067	0.6063	0.9803	0.8874
1–6	0.0632	0.5484	0.9888	0.8971
1–5–6	0.0566	0.5196	0.9897	0.9008
1–4–5–6	0.0688	0.5285	0.9899	0.9031
1–2–4–5–6	**0.0592**	**0.5828**	**0.9905**	**0.9059**
0–2–3–4–5–6	0.0595	0.5537	0.9891	0.9049
0–1–2–3–4–5–6	0.0646	0.5587	0.9913	0.8945

F. Effectiveness of Raw Data without Preprocessing

As shown in Table 8, Experiment 6 presents the correlation results using raw data without preprocessing. Even though the results obtained after data pre-processing were good, working with raw data yielded the best results.

The observation that working with raw data yields better results than working with processed data is a pivotal finding that merits a detailed exploration in our discussion. This conclusion challenges conventional wisdom in the field of Natural Language Processing (NLP) and data science, where preprocessing techniques like lemmatization, stemming, and normalization are traditionally employed to clean and standardize data before analysis. These techniques are designed to reduce noise, minimize complexity, and standardize linguistic variations, ostensibly to enhance the performance of machine learning models.

Our experiments, particularly Experiment 6, which demonstrated a remarkable correlation score of 92.34% by analyzing raw data with a combination of five pre-trained models, underscore the potential advantages of directly leveraging the inherent complexity and richness of raw data. This approach contrasts with the outcomes observed in Experiments 1 through 5, where various preprocessing techniques were applied, resulting in slightly lower correlation scores ranging from 89.85% to 90.93%.

The superior performance observed when using raw data suggests that pre-processing, despite its intentions to clarify and simplify, might inadvertently filter out subtle nuances and critical information encoded in the raw form of text. This information could be crucial for the models to capture complex patterns, contextual nuances, and the diversity of linguistic expressions, which are essential for tasks requiring

a deep understanding of language. One more finding is that preprocessing might enhance results, and we can claim that the best pipeline could be problem-dependent. Our recommendation, based on the experiments, for the Arabic short answer grading is explained in Fig. 7.

Table 8. Experimental results on the raw data

Applied Models	Train Loss	Testing Loss	Train Correlation	Testing Correlation
1	0.1957	0.5120	0.9655	0.9053
1–6	0.0778	0.4920	0.9860	0.9146
1–4–6	0.0716	0.4742	0.9878	0.9172
0–1–4–6	0.0708	0.4223	0.9889	0.9192
1–2–4–5–6	**0.0682**	**0.4476**	**0.9887**	**0.9234**
0–1–2–4–5–6	0.0703	0.4632	0.9888	0.9190
0–1–2–3–4–5–6	0.0689	0.4896	0.9887	0.9120

G. Optimal Pipeline Configuration for Highest Accuracy

Fig. 7 depicts the most effective pipeline, which yielded the highest Pearson correlation score, namely, 92.34%.

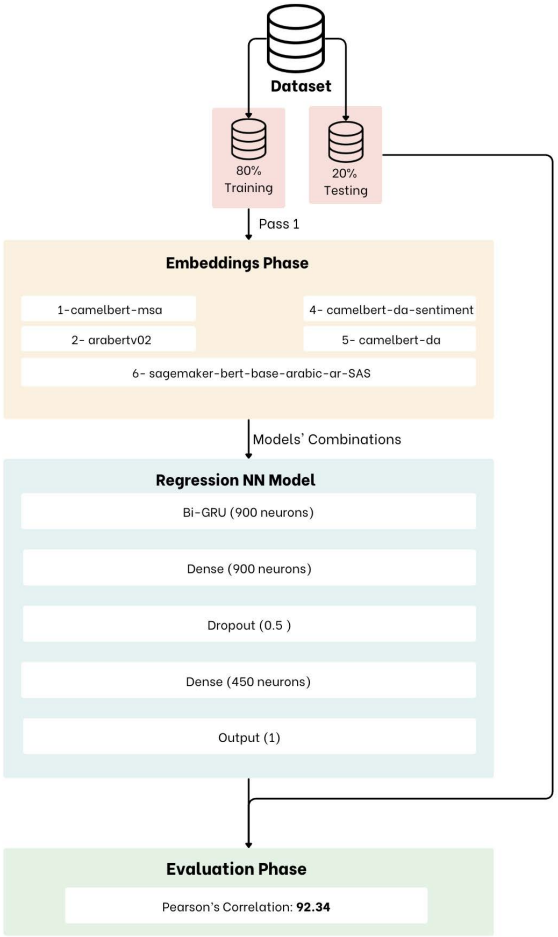

Fig. 7. The proposed approach pipeline.

Table 9. Pearson correlation comparison on the CUD

Applied Models	Pearson Correlation
Proposed Approach	**0.9234**
[31]	0.84
[27]	0.79
[34]	0.86

This pipeline is initiated by dividing the data into an 80% training set and a 20% testing set. During the embedding phase, we employ five models: model 1, model 2, model 4, model 5, and model 6, as referenced in Table 3. Then, a neural network architecture was utilized, followed by testing and evaluation based on the Pearson correlation score.

Table 9 presents a comparison between the proposed approach and the previous related work.

H. Proposed Approach Validation

To further substantiate the effectiveness and versatility of the proposed approach, we conducted additional validation on a different dataset, namely the AR-ASAG dataset [14]. As presented in Table 10, the proposal surpassed the current state-of-the-art and achieved an outstanding Pearson correlation of 80%. The superiority of the proposal clearly confirms our findings.

Table 10. Pearson correlation comparison on AR-ASAG dataset

Applied Models	Pearson Correlation
Proposed Approach	**0.80**
[14]	0.70
[18]	0.78
[22]	0.745

What is particularly noteworthy is how this performance underscores the proposal's ability to generalize effectively to unseen data, which is critical in real-world applications of automated short answer grading. By excelling in the context of Arabic short answer grading, our approach addresses a niche yet important domain, highlighting the model's adaptability across various datasets and its potential for deployment in broader educational and linguistic applications. These results strengthen our confidence in the approach as a state-of-the-art solution capable of scaling and maintaining performance across diverse tasks within the field.

V. Discussion

In the context of automatic short-answer grading, it is particularly noteworthy that using raw textual data often results in the highest correlation scores when compared to preprocessed forms such as stemmed, lemmatized, or normalized data. A plausible explanation for this observation is that raw data preserves the original linguistic context and subtle nuances essential for understanding and accurately evaluating student responses. Preprocessing techniques, while helpful for standardization, can sometimes strip away important information necessary for accurate assessment making raw input more effective in capturing the depth of meaning in student answers.

To enhance semantic and syntactic understanding, this study explores the integration of multiple Arabic sentence transformer models. Each of these models is fine-tuned for a particular NLP task, and their combination allows for a more enriched representation of the input.

For example, one model might be particularly adept at capturing Arabic syntactic patterns, while another may excel at understanding semantics or dialectal subtleties. Their collective use enables a more comprehensive analysis of student answers, thereby increasing grading accuracy.

This multi-model approach is especially valuable in handling the complexity and richness of the Arabic language. Furthermore, combining diverse model outputs reduces the potential bias or limitations of individual models, leading to a more stable and accurate scoring system.

However, this strategy does come with a trade-off incorporating multiple embedding models into the pipeline significantly increases computational requirements, demanding greater memory and processing resources during

both training and inference.

During experimentation with Arabic sentence transformer ensembles for short-answer grading, we identified that the most effective model combination excluded two specific models:

- Intfloat/multilingual-e5-base
- ChaimaaBouafoud/arabicSent-ChamaBert

These exclusions were made for two primary reasons. The Intfloat/multilingual-e5-base model, although supporting 94 languages, was not specifically optimized for Arabic and is documented to perform less effectively on low-resource languages, potentially compromising its utility for Arabic-only tasks. Meanwhile, ChaimaaBouafoud/arabicSent-ChamaBert, though Arabic-focused, was primarily trained on Moroccan Arabic. Since our dataset does not center on Moroccan dialectal vocabulary, this mismatch likely contributed to its suboptimal performance.

Ultimately, the proposed model ensemble has proven its generalization capability by achieving state-of-the-art performance on two different and challenging Arabic datasets. This strong performance highlights the robustness and adaptability of our approach, confirming its effectiveness across diverse scenarios in Arabic short-answer grading.

VI. Conclusions

The task of automatically grading Arabic short answers is of great value to both students and educational institutions. In this study, we addressed this challenge using deep learning techniques, including neural networks and Arabic-specific sentence embedding models. Our approach demonstrated strong performance when applied to raw, unprocessed data and further enhanced grading accuracy by integrating multiple pre-trained embedding models. This resulted in a Pearson correlation of 92.34%, setting a new benchmark for the Cairo University Dataset.

To evaluate generalizability, we tested the proposed model on a second, unseen dataset and achieved superior performance over existing methods. These results underscore the robustness, adaptability, and effectiveness of the approach across different datasets.

Future work may focus on integrating more advanced pre-trained models and extending the approach to additional datasets to further validate and enhance its applicability.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Methodology: Emad Nabil, Mostafa Mohamed Saeed, Rana Reda, Safiullah Faizullah, and Wael Gomaa; Software: Mostafa Mohamed Saeed, Rana Reda; Supervision: Emad Nabil, Safiullah Faizullah, and Wael Gomaa; Writing original draft: Mostafa Mohamed Saeed, Rana Reda; Writing review and editing: Emad Nabil, Mostafa Mohamed Saeed, Rana Reda, Safiullah Faizullah, and Wael Gomaa; Administration: Emad Nabil. All authors had approved the final version.

Acknowledgment

This work was supported by the Deanship of Scientific Research, Islamic University of Madinah, KSA.

References

[1] K. Darwish *et al*., "A panoramic survey of natural language processing in the Arab world," *Commun. ACM*, vol. 64, no. 4, pp. 72–81, Apr. 2021.

[2] W. H. Gomaa and A. A. Fahmy, "Tapping into the power of automatic scoring," presented at 11th Int. Conf. Lang. Eng., Egyptian Society of Language Engineering, 2011.

[3] M. G. Hahn, S. M. B. Navarro, L. D. L. F. Valentín, and D. Burgos, "A systematic review of the effects of automatic scoring and automatic feedback in educational settings," *IEEE Access*, vol. 9, pp. 108190–108198, 2021.

[4] S. Grossberg, "Recurrent neural networks," *Scholarpedia*, vol. 8, no. 2, p. 1888, Feb. 2013.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Dec. 1997.

[6] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2013, pp. 273–278.

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv:1412.3555, Dec. 2014.

[8] W. H. Gomaa, A. E. Nagib, M. M. Saeed, A. Algarni, and E. Nabil, "Empowering short answer grading: Integrating transformer-based embeddings and BI-LSTM network," *Big Data Cogn. Comput.*, vol. 7, no. 3, 2023.

[9] M. M. Saeed and W. H. Gomaa, "An ensemble-based model to improve the accuracy of automatic short answer grading," in *Proc. 2nd Int. Mobile, Intell. Ubiquitous Comput. Conf. (MIUCC)*, May 2022, pp. 337–342.

[10] A. Vaswani *et al*., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[11] A. Magooda *et al.*, "Vector based techniques for short answer grading," in *Proc. FLAIRS Conf.*, 2016, pp. 238–243.

[12] W. H. Gomaa and A. A. Fahmy, "A survey of text similarity approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, Apr. 2013.

[13] M. Awad and R. Khanna, "Support vector regression," in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, M. Awad and R. Khanna, Eds., Berkeley, CA: Apress, 2015, pp. 67–80.

[14] L. Ouahrani and D. Bennouar, "AR-ASAG: An Arabic dataset for automatic short answer grading evaluation," in *Proc. 12th Language Resources and Evaluation Conference (LREC 2020)*, May 2020, pp. 2634–2643.

[15] W. H. Gomaa and A. A. Fahmy, "Short answer grading using string similarity and corpus-based similarity," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 11, Jan. 2012.

[16] W. H. Gomaa and A. A. Fahmy, "Arabic short answer scoring with effective feedback for students," *Int. J. Comput. Appl.*, vol. 86, no. 2, pp. 35–41, Jan. 2014.

[17] K. Taghva, R. Elkhoury, and J. Coombs, "Arabic stemming without a root dictionary," in *Proc. Int. Conf. Inf. Technol.: Coding Comput. (ITCC'05)*, Apr. 2005, pp. 152–157.

[18] O. Nael, Y. Elmanyalawy, and N. Sharaf, "AraScore: A deep learning-based system for Arabic short answer scoring," *Array*, vol. 13, 100109, Jan. 2022.

[19] M. A. Hearst *et al*., "Support vector machines," *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998.

[20] J. Torre, D. Puig, and A. Valls, "Weighted kappa loss function for multi-class classification of ordinal data in deep learning," *Pattern Recognit. Lett.*, vol. 105, pp. 144–154, Apr. 2018.

[21] K. Clark, M.-T. Luong, and Q. V. Le, "ELECTRA: Pre-training text encoders as discriminators rather than generators," arXiv:2003.10555, 2020.

[22] R. M. Badry *et al*., "Automatic Arabic grading system for short answer questions," *IEEE Access*, vol. 11, pp. 39457–39465, 2023.

[23] S. T. Dumais, "Latent semantic analysis," *Annu. Rev. Inf. Sci. Technol.*, vol. 38, pp. 189–230, 2004.

[24] A. S. Hussein, "Arabic document similarity analysis using n-grams and singular value decomposition," in *Proc. 9th IEEE Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, May 2015, pp. 445–455.

[25] M. A. Salam, M. A. El-Fatah, and N. F. Hassan, "Automatic grading for Arabic short answer questions using optimized deep learning model," *PLoS ONE*, vol. 17, no. 8, e0272269, Aug. 2022.

[26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[27] (2024). Ans2vec: A scoring system for short answers. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-

14118-9_59
[28] H. A. Abdeljaber, "Automatic Arabic short answers scoring using longest common subsequence and Arabic WordNet," *IEEE Access*, vol. 9, pp. 76433–76445, 2021.
[29] Z. Peng and Y. Wang, "A novel efficient graph model for the multiple longest common subsequences problem," *Front. Genet.*, vol. 8, Aug. 2017.
[30] W. Black *et al*., "Introducing the Arabic WordNet project," arXiv:cs/0605061, 2006.
[31] A. E. Magooda, M. A. Zahran, M. A. Rashwan, H. Raafat, and M. B. Fayek, "Vector based techniques for short answer grading," in *Proc. FLAIRS Conf.*, 2016, pp. 238–243.
[32] M. Mohler and R. Mihalcea, "Text-to-text semantic similarity for automatic short answer grading," in *Proc. 12th Conf. Eur. Chapter Assoc. Comput. Linguist.*, 2009, pp. 567–575.
[33] M. Mohler, R. Bunescu, and R. Mihalcea, "Learning to grade short answer questions using semantic similarity measures and dependency graph alignments," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguist.: Human Language Technol.*, 2011, pp. 752–762.
[34] W. H. Gomaa and A. A. Fahmy, "Automatic scoring for answers to Arabic test questions," *Comput. Speech Lang.*, vol. 28, no. 4, pp. 833–857, 2014.
[35] M. A. Zahran, A. Magooda, A. Y. Mahgoub, H. Raafat, M. Rashwan, and A. Atyia, "Word representations in vector space and their applications for Arabic," *Comput. Linguist. Intell. Text Process.*, pp. 430–443, 2015.
[36] R. Duwairi and M. El-Orfali, "A study of the effects of preprocessing strategies on sentiment analysis for Arabic text," *J. Inf. Sci.*, vol. 40, no. 4, pp. 501–513, Aug. 2014.
[37] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of Arabic word embedding models for use in Arabic NLP," *Procedia Comput. Sci.*, vol. 117, pp. 256–265, Jan. 2017.
[38] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Oct. 2014, pp. 1532–1543.
[39] A. Karipbayeva, A. Sorokina, and Z. Assylbekov, "A critique of the smooth inverse frequency sentence embeddings (student abstract)," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 10.
[40] W. H. Gomaa and A. A. Fahmy, "SimAll: A flexible tool for text similarity," in *Proc. the Seventeenth Conference on Language Engineering ESOLEC*, Cairo, Egypt, 2017, pp. 122–127.
[41] W. H. Gomaa, "A multi-layer system for semantic relatedness evaluation," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 23, pp. 3536–3544, 2019.
[42] A. Purandare and T. Pedersen, "Word sense discrimination by clustering contexts in vector and similarity spaces," in *Proc. 8th Conf. Comput. Nat. Lang. Learn. (CoNLL-2004) at HLT-NAACL 2004*, Boston, MA, USA: Association for Computational Linguistics, May 2004, pp. 41–48.
[43] R. Kruse *et al*., "Multi-layer perceptrons," *Computational Intelligence: A Methodological Introduction*, Cham: Springer, 2022, pp. 53–124.
[44] R. Qasim *et al*., "A fine-tuned BERT-based transfer learning approach for text classification," *J. Healthc. Eng.*, vol. 2022, 3498123, 2022.
[45] P. Sedgwick, "Correlation," *BMJ*, vol. 345, e5407, Aug. 2012.
[46] L. Wang *et al*., "Multilingual E5 text embeddings: A technical report," Feb. 2024, arXiv:2402.05672.
[47] G. Inoue *et al*., "The interplay of variant, size, and task type in Arabic pre-trained language models," arXiv:2103.06678, Sep. 2021.
[48] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," arXiv:2003.00104, Mar. 2021.
[49] A. Safaya, M. Abdullatif, and D. Yuret, "KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media," arXiv:2007.13184, Jul. 2020.