Applied Computing Journal   https://doi.org/10.52098/acj.20255341
ISSN: 2788-9688 Volume 5, Issue 3, pp 399-413, April 2025
Received: 9/4/2025   Revised: 18/4/2025   Accepted: 19/4/2025

Auto-Grading Arabic Short Answer Question

Marwa Ali Abdulsamad[1]*, Salma Abdulbaki Mahmood[1]
[1] Department of Computer Science, Basra University, Basra, Iraq
* Correspondence Author Marwa Ali Abdulsamad, pgs.marwaa.ali@uobasrah.edu.iq

Abstract

Automated Essay Grading Systems (AEGS) have become the main tools to address the challenges associated with manual essay grading, especially in Arabic. These systems use advanced NLP and Machine Learning techniques to support and enhance grading, rating efficiency, and equality. Despite significant improvements in automated grading systems, studies on Arabic essay evaluation are limited due to the Arabic language's unique morphological and syntactic complexities. This paper presents a unique automated grading system for short-form Arabic essay questions. The system applies text representation techniques, such as Word2Vec and TF-IDF, and text similarity techniques, such as LSA, LCS, Cosine Text Similarity, and Jaccard Text Similarity measurements. Furthermore, a stacking-based machine learning model supplies these estimates to attain a coherent and reliable grading system. The system's strength is determined by using metrics such as mean absolute error (MAE), Root Mean Square Error (RMSE), Pearson correlation, and Spearman correlation. Experimental results establish the validity of the unified technique in achieving high accuracy and strong correlations with human raters' ratings. The stacking model (TF-IDF + Jaccard + LCS + LSA) showed outstanding performance, resulting in negligible errors (MAE = 0.81, MSE = 0.96) and significant correlations (Pearson = 0.73, Spearman = 0.76). The advanced approach is very accurate and strongly correlates with human ratings, providing a scalable and economical solution for correcting Arabic text.

Keywords: TF-IDF, latent semantic analysis; AI-powered systems; longest common subsequence; natural language processing; machine learning

1. Introduction

Automated Essay Grading Systems (AEGS) Production Promising machine engineered to mirror human correcting procedures by examining essay content, grammatical precision, structural attachment, and overall qualitative estimate.

These systems employ NLP techniques with machine learning algorithms to assess accurate, steady, and efficient grading.

Considering the escalating demand for scalable educational solutions, AEGS presents a pragmatic strategy for mitigating the challenges inherent in traditional grading practices, such as time constraints and subjective biases. These applications are using NLP techniques and machine learning algorithms on their own to evaluate and automatically assign grades to student essays (Al-Shargabi et al., 2021).

The issue of human grading, as the most essential factor for time which university professors spend on grading, represents the pragmatic standpoint of the duration that educators could have used in different pedagogical activities more productivity. The number of assignments that need to be marked is getting bigger since schools and colleges have recorded a big increase in the number of students enrolled in their courses. In fact, the timeframe for the marking of essay questions makes their process more complex in comparison to other forms of assessment. The reason is that they are an open-book, open-note, and online resource that provides thus students with hints from the examiners so that students would not struggle with the given tasks. As a result, an e-Grading System could fully replace manual grading for teachers causing the latter to engage in various other tasks such as providing better lesson plans, giving feedback, and class management after developing innovative ideas. Consequently, it is expected that institutions of higher education would initiate the use of automated grading systems to make the process of measuring the growth of a student's performance more efficient. (Azmi et al., 2019).

Besides, the use of technology, i.e., Automated Essay Grading Systems (AEGS), as a method of grading writing tasks, is not only effective for the academic sector but it has also proven to be one of the most practical ways of assessment. The meaning is that an essay grading program can evaluate and give grades for written essays automatically, without human touch, by relying on the scientific and computational discipline that is commonly called Computer Science. It is known to all that computer grading is of students’ writing assessments which is of great benefit in that it is quick, efficient, accurate, and cost-effective. Recent studies revealed that it has been reported as a major source that institutions and teachers recognize as useful for their work and that the system has several core advantages making it an indispensable software tool (Al-Awaida, S. 2019).

The development of an automated essay question grading application necessitates the application of Natural Language Processing (NLP) techniques, which commence with the aggregation of a corpus of student responses. The responses require comprehensive annotation and cleansing prior to the extraction of significant features for digital conversion to facilitate analysis in comparison to the instructor-provided prototypical responses. Furthermore, it necessitates the employment of machine learning algorithms to derive the evaluation scores assigned.to each student’s response (Shehab et al., 2018).

The short answers configuration, as well as the morphological, syntactic and semantic complexity of short answers, as well as the complication of identifying the semantics of the sentences and the words inside its semantic contexts correctly are the most significant problems noticed in developing automatic essay question evaluation systems (AEGS). And this is even more true when the AEGS systems are trained on Arabic questions. However, it introduced new challenges for Arabic text clarification which has a complex morphology, resulting in a complex word definition and a high number of meanings for a specific word form (Larkey et al., 2002). Syntactic and semantic ambiguities are widespread in Arabic text. Stopping words is based on significant context; removing them sometimes requires a deep understanding of grammar and syntax (Farghaly & Shaalan, 2009). Arabic stemming poses significant challenges due to the language's inflectional characteristics and the use of diverse prefixes, infixes, and suffixes (Al-Shalabi et al., 2004). Finally, and most importantly, there is the scarcity of authentic Arabic datasets (corpora), which may be appropriate to contribute to building more reliable and trustworthy computer systems (Omran & Ab Aziz, 2013).

This study focuses on developing an automated system for grading essay questions, particularly those with short answers written in Arabic. A novel hybrid approach has been proposed, combining the contextual representation of word2vec with the positional and semantic representation of both teacher and student answers, using various methods to measure their textual similarity. Furthermore, a machine learning model has been introduced to unify these diverse evaluations into a single grading score that is fairer to students and closer to the teacher's assessment. This methodology and its results will be detailed and presented within the framework of this study.

2. Literature Survey

The following section introduces a comprehensive review of the most notable studies addressing the automated assessment of short response questions, the simplest form of essay questions, and the various approaches used in each study. We will classify these studies based on the approaches used, focusing on the merits and drawbacks of each, thereby enabling the identification of existing deficiencies and possible opportunities for improvement in current systems. The progressive development of Automated Essay Grading (AEG) systems highlights their importance in modern education, enhances grading procedures and maintains uniformity. This section examines significant AEG research using text similarity algorithms, emphasizing techniques and principal findings.

2.1. English Language AEG Systems

Al-Awaida created an automated essay evaluation framework for Arabic texts through the combination of Support Vector Machine (SVM) methods and text similarity algorithms. The research of Al-Awaida examined the assessment of brief Arabic responses through supervised machine learning techniques. The system enabled the SVM classifier training and assessment through Arabic student texts that received linguistic processing and similarity analysis. The framework showed promising results through its Pearson correlation coefficient of 0.756 but faced two main limitations due to its restricted dataset size and insufficient semantic processing depth. The authors suggested that future systems should use word embeddings while expanding their dataset size.

The system developed by Ramalingam (Ramalingam et al., 2018) united Bayes' Theorem with e-Rater functionalities to create a machine learning-based essay evaluation system. The framework used multiple linguistic attributes to measure lexical richness together with grammatical precision and sentiment orientation. The system received training from a Kaggle dataset that was divided into eight separate scoring categories.

The model achieved more than 80% accuracy following stemming and stop-word removal preprocessing when compared to human evaluators. The researchers suggested future work to enhance syntactic and semantic features and to use neural network architectures to increase accuracy.

2.2. Arabic Language AEG Systems

The author Shehab and his team (Shehab et al., 2018) developed an automatic Arabic essay grading system that uses text similarity algorithms based on the Bag of Words (BOW) model. The research study employed 210 sociology responses from secondary-level students who received grades ranging from 0 to 5 from multiple evaluators. The character-based N-gram approach generated the highest correlation value of 0.803 when used with the Damerau-Levenshtein (DL) distance. Stop-word processing proved beneficial for the system to improve its precision. The research failed to include semantic methods in its analysis. The future work should focus on merging string-based and corpus-based approaches together with synonym-based techniques and extending the system to different datasets.

Badry (Badry et al., 2023) created an automatic Arabic grading system for short answer questions, which utilized the AR-ASAG dataset that included 2,133 students and model answer pairs. The system evaluated two different weighting approaches, which included local weighting alone and a combination of local and global weights. The hybrid approach performed better than the local-only approach by lowering RMSE to 0.798 while reaching an F1-score of 82.82%. The system demonstrates high accuracy, but its performance can be enhanced by adding neural networks and Arabic WordNet integration for improved semantic representation.

Al Awaida (Al-Awaida et al., 2019), developed an Arabic essay grading system that combined Arabic WordNet with Support Vector Machines (SVM) and cosine similarity. The research used 40 computer science and social studies questions with student answers, which followed Hewlett Foundation guidelines. Their model reached a minimum mean absolute error (MAE) of 0.117 and enhanced accuracy by 2.648%. The study faces limitations in scalability because it depends on lexical features. The authors suggested that machine learning models, together with neural networks and bigger datasets, should be integrated to boost system performance.

Another study by Ouahrani (Ouahrani & Bennouar, 2020) introduced the AR-ASAG dataset, a resource designed for Arabic short answer grading evaluation, containing 2,133 pairs of student and model answers in both .txt and .xml formats. They proposed an unsupervised grading approach based on the COALS algorithm for semantic similarity and a summation vector model with term weighting. Their experiments investigated the effects of domain specificity, semantic space dimensions, and stemming, showing promising results in automatic Arabic grading. However, the model lacks human guidance. The researchers plan to develop a supervised version that incorporates teacher insights and addresses practical grading challenges.

Gomaa and Fahmy (Gomaa & Fahmy, 2020) proposed Ans2vec as an unsupervised framework to evaluate short answers through pre-trained sentence embeddings without requiring NLP preprocessing or linguistic resources. The system underwent validation using three datasets which included Texas with 80 questions and 2,273 responses and Cairo University with 61 questions and 610 responses and SCIENTSBANK with five response categories. The method produced semantic vectors from answers which resulted in a Pearson correlation of 0.63. The system does not combine supervised learning methods with feature engineering techniques. Future work should investigate Sent2Vec as an alternative to sentence embedding models and implement machine learning classifiers while extending the system's support for Arabic and multiple languages.

Azmi (Azmi et al., 2019) developed an Arabic automated essay grading system through a two-stage approach that combined training with evaluation. Researchers tested the system using 350 handwritten Arabic essays that originated from four educational institutions which covered different academic levels and topics. The training consisted of 300 essays while the evaluation set contained 50 essays. The grading system operated from 0 to 10 points and used a third human evaluator to resolve grading disputes between the first two raters. The system reached a Pearson correlation of 0.756 which exceeded the typical 0.709 Arabic essay correlation and matched the 0.85 English essay grading correlation. The system delivered promising scores with 90% accuracy but failed to include sophisticated semantic features. The system can be improved through Word2Vec representation integration and dataset expansion, and additional human rates to enhance grading precision.

Table 1: summarizes some studies in the literature survey.

Author(s), Year	Method	Findings
Al-Awaida S., 2019	LCS, Common Words (COW), Semantic Distance (SD)	82% correlation with human grading
Ramalingam, et al., 2018	e-Rater, Bayes' Theorem, lexical & grammatical analysis	Accuracy >80%, reduced errors
Shehab et al., 2018	N-gram, Damerau-Levenshtein (DL), stop-word removal	Haigh accuracy with N-gram (0.803 with DL)
Badry et al., 2023	Hybrid local & global weighting (AR-ASAG dataset)	F1-score: 82.82%, RMSE: 0.798
Al Awaida et al., 2019	SVM, cosine similarity, Arabic WordNet	2.648% improvement in grading accuracy
Ouahrani & Bennouar, 2020	COALS algorithm, term weighting	Positive results for Arabic grading
Gomaa & Fahmy, 2020	Sent2vec embeddings, no NLP preprocessing	Pearson correlation: 0.63
Azmi et al., 2019	Two-stage Arabic essay grading, 350 handwritten essays	Pearson: 0.756, 90% accuracy

3. Research Methodology

The main aim of this study is to introduce a comparative study using different techniques for text representation, text similarity, and enameled ML, namely stacking, to find the best AEGS model for Arabic short-answer questions. The following conceptual framework illustrates the general diagram of this study's overall work, as shown in Figure 1. It shows the stages of the proposed short answer assessment system, from preprocessing model answers and students' answers through text representation and similarity calculation to predicting scores using the Stacking technique and conducting the statistical evaluation of the model.

3.1. Data Set

For developing the grading system, the Arabic AR-ASAG dataset contained 2133 student answers. This dataset was available in the GitHub repository (Manning et al, 2008) in three formats: text (.TXT), (. XML-MOODLE) XML, and database (.DB). This dataset includes the reported scores linked to three types of student replies from three separate tests. The evaluations were placed in an environment that mimics real life. 48 questions were administered, with 16 short-answer questions for each exam. A model response (teacher answer) was provided for each question. In response to these questions, students turned in their work. From question to question, the number of responses fluctuated. 2133 pairs (sample response, student answer) make up this dataset. Five different kinds of questions contained in this dataset include:

- " عرف": Define?
- "اشرح": Explain?
- "ما النتائج المترتبة على": What consequences?
- "علل": Justify?
- "ما الفرق": What is the difference

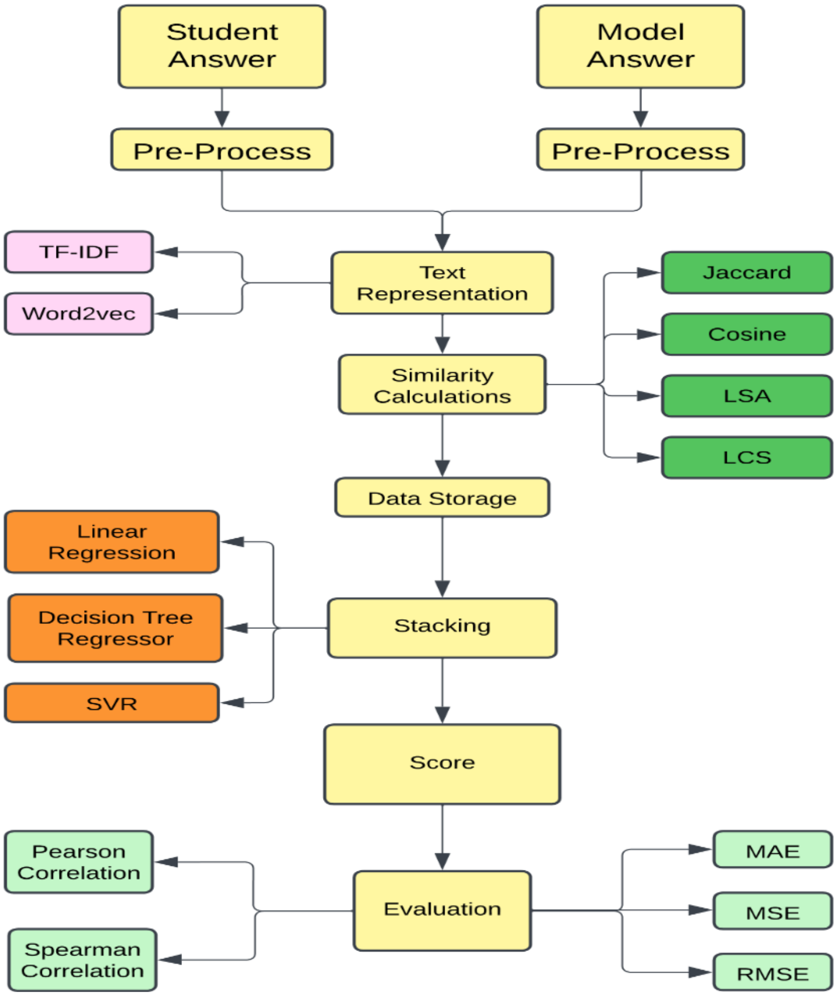

Figure 1: A diagram of comparative analysis.

3.2. Text Preprocessing

Text preprocessing is essential in formulating an automated essay evaluation system, especially concerning Arabic short-answer inquiries. Each preprocessing phase is pivotal in enhancing the precision and efficacy of the subsequent text classification and similarity evaluations as shown in Figure 2.

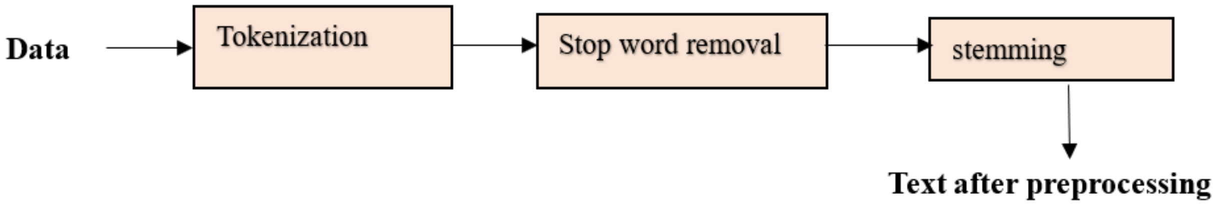

Figure 2: A diagram of Text preprocessing.

3.2.1 Tokenization

Slicing is the process of dividing text into smaller units, called symbols, which typically represent individual words. By breaking down sentences into tokens, the system can perform detailed linguistic analysis and prepare the text for subsequent processing steps, such as similarity assessment.

3.2.2 Stop words removal

Stop words are frequently occurring lexical items that typically lack substantial semantic weight, such as conjunctions, prepositions, and pronouns. Common Arabic stop words include terms like ' 'إيانا', 'إياكم', 'إياكما', 'إياكن', 'أو', 'أنى', 'أنتن', 'أنتم', 'أنتما', 'أنت', 'أيا', 'آها', 'آه', 'إيه', 'أينما', 'أين', 'أي'", among others. Removing these items reduces useless noise in the dataset, permissive the analytical system to focus on lexemes that convey significant semantic information.

3.2.3 stemming

Stemming is an important preprocessing phase in Arabic NLP because of the complexity of its morphology in terms of inflection and derivation. It can enhance the precision and efficiency of information retrieval systems considerably by classifying words into their base forms thus facilitating the clustering of related terms (Yousif & Sembok, 2008). The ISRI Arabic Stemmer is a very valuable tool in Arabic natural language processing (Yousif, J., 2018), especially for stemming. It is part of the NLTK package that cut down words to their root form (Jiwani et al., 2022). The procedure of stemming is stopped when the residual length of the input term is three characters or less. For instance, the word "تحميلها " after stemming will become “حمل ".

3.3. Text Representation

Table 2 summarizes a comparison-test of TF-IDF vs. weight (word2vec) in terms of technical characteristics and functioning procedures with special focus on their unique advantages. This comparison demonstrates the distinct representational phenomena of each approach; hence, their utility across computational and linguistic areas as two competitive candidates for text representation across a broad range of applications, including semantic feature extraction, and lexical similarity analysis (Zhan, Z., 2025).

Table 2: a comparison-test of TF-IDF vs. weight (word2vec)

Feature	TF-IDF	word2vec
Approach	A statistical measure based on term frequency (TF) and inverse document frequency (IDF)	A neural network model that generates continuous vector representations of words based on context
Output	A sparse matrix where each word is a unique dimension, with values representing the TF-IDF score	Dense vectors (embeddings) in continuous space, typically with 100-300 dimensions
Contextual Understanding	Treats each word independently, without considering context, and does not capture semantic relationships	Captures semantic relationships and context, positioning similar words close to each other
Dimensionality	High-dimensional and sparse, which can lead to computational inefficiency with large corpora	Low-dimensional and dense, making it computationally efficient for various NLP tasks
Handling Rare Words	Rare words may have high TF-IDF scores if they are unique to a document, but this may not indicate their importance.	Can learn meaningful embeddings for rare words based on their context
Applications	It is commonly used for document retrieval and ranking, text classification, and information retrieval.	Used for tasks requiring semantic understanding, such as sentiment analysis and machine translation

This research used two distinct text representation methodologies for student brief replies: Word2Vec, which set up context-sensitive word embeddings, and TF-IDF (Term Frequency-Inverse Document Frequency), which relies on term frequency. These methods were used to assess the efficacy of different text representation techniques in accurately representing student responses in automated grading systems concerned with both lexical and semantic types of short texts (Sharma & Singh, 2024; Zhou et al., 2024).

3.4. Text Similarity

Text similarity is assigned to techniques for definitively determining the similarity between two phrase components. Various NLP applications, such as information retrieval, text categorization, and clustering, adopted these techniques. The most widely used research methods in this study are LSA, LCS, Jaccard Similarity, and Cosine Similarity.

4.3.1. Latent Semantic Analysis

LSA is an unsupervised learning approach that captures and represents the contextual semantics of words corresponding to the relationships among a group of texts and the related lexical elements. It can be implemented in Python (see Appendix A for code).

3.4.2. Long Common Substring

The longest contiguous character sequence common to two text strings is determined using the LCS method. This measure, which highlights universal, continuous wording, is useful in automated grading since it demonstrates the degree of similarity between a student's answer and the model response. For more details about the code, see Appendix B.

3.4.3 Cosine Similarity

Cosine similarity is a prevalent measure in NLP for determining how two textual factors are similar as defined in equation (1). It measures the cosine of the angle between two texts' vector representations in a multifaceted space to determine how similar they are. This measure is beneficial since it can compare documents of different lengths by exacting the direction of the vectors rather than their significance. Text documents are converted to vector representation, frequently using methods like TF-IDF. This method allows for a more nuanced comparison by highlighting the emphasis on terms within the larger corpus (Thongtan & Phienthrakul, 2019; de Vos et al., 2022).

$$Similarity = \cos\theta = \frac{A\ .B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\ \sqrt{\sum_{i=1}^{n} B_i^2}} \quad \ldots (1)$$

The results range from -1 to 1, where -1 illustrates differing papers, 0 illustrates orthogonality (dearth of similarity), and 1 illustrates identical documents (Thongtan & Phienthrakul, 2019; de Vos et al., 2022).

3.4.4 Jaccard Similarity

Jaccard similarity quantifies the closeness between two sets by computing the ratio of the size of their crossing to the size of their union as defined in equation (2). It is a straightforward metric often used to compare collections of words or characters (Leskovec et al., 2014).

$$\text{Jaccard Similarity} = |A \cup B| / |A \cap B| \quad \ldots (2)$$

Where:

- A and B denote the sets, including the words or characters in two distinct texts.
- $|A \cap B|$ represents the cardinality of the intersection of sets A and B, indicating the quantity of shared items.
- $|A \cup B|$ represents the total count of distinct items in the union of both sets.

The Jaccard similarity is often used to evaluate texts at the set level by comparing two sets of words post-tokenization. It is often used for duplication detection, where precise or almost precise matching is crucial. An example of using the Jaccard similarity between the sets of words in sentence A, {dog, cat, fish}, and sentence B, {dog, cat, bird}, would be: Jaccard Similarity = 2/4 = 0.5. The two groups share the terms dog and cat, although fish and bird are not. The size of the sets influences the Jaccard similarity metric and does not include word frequency, which might be significant in some instances. It disregards the sequence or context of the words. Hence, two sets of words with identical parts but divergent meanings might still be comparable (Leskovec et al., 2014).

3.5. Stacking

Stacking (stacked generalization) is a complicated machine learning method that amalgamates predictions from many base models to improve overall predictive veracity. The ability to address the limitations of individual models by integrating the benefits of other methodologies, such as semantic embeddings (Word2Vec) and lexical features (TF-IDF), validates its use in automated grading systems. This integration improves alignment with human bias by including latent semantic (LSA) and structural (LCS) linkages, reducing bias and variation. Consequently, stacking is the optimal method for addressing linguistic challenges in Arabic texts, including syntactic variations and morphological variety as shown in Figure 3.

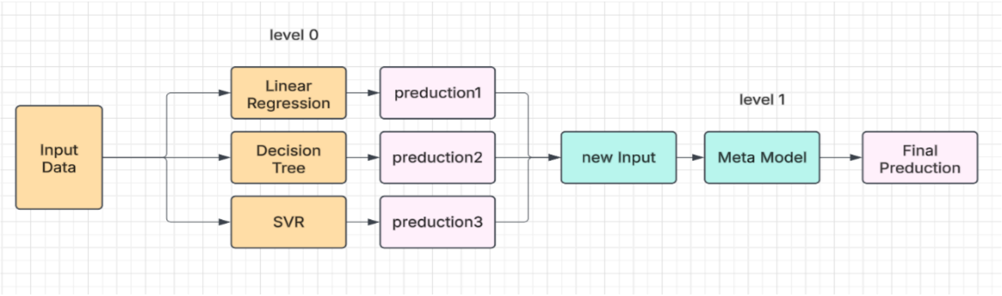

Figure 3: Stacking Ensemble Framework.

3.6. 3. Evaluating

Correlation: Correlation quantifies the link between two variables, often between expected scores and actual values as described in equation (3). In essay evaluation, correlation is crucial in assessing the alignment between the system's ratings and human scores. The predominant correlation metric is Pearson’s correlation coefficient, computed as:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})}^2} \ldots (3)$$

Where:

x_i and y_i represent individual observations, whereas $\bar{x}$ and $\bar{y}$ denote the means of the observed and forecasted values, respectively.

r ranges from -1, indicating perfect negative correlation, to +1, denoting perfect positive correlation, with 0 reflecting a lack of connection.

Correlation evaluates the magnitude and orientation of the linear relationship between anticipated and actual results as described in equation (4). It is often used in regression analyses (Yousif, J. & Yousif, M. 2024). Error (Mean Squared Error, Root Mean Squared Error): Error measures quantify the disparity between expected values and actual values. Frequently used error metrics comprise.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})}^2} \ldots (4)$$

Mean Squared Error (MSE): Quantifies the average of the squares of the discrepancies between expected and actual values as outlined in equation (5) (Kazem et al., 2019).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \ ..(5)$$

Where:

y_i represents actual values, and

$\widehat{y_i}$ denotes anticipated values.

Root Mean Squared Error (RMSE): The square root of the Mean Squared Error (MSE). It produces an error in the same units as the real numbers as identified in equation (6).

$$RMSE = \sqrt{MSE} \ldots (6)$$

Application: These measures are used when the amount of the mistake is of significance (e.g., in regression tasks). RMSE is more interpretable since it has the same units as the projected values (Kazem et al., 2022).

4. Experimental Results and Discussion

This section comprehensively analyzes the proposed models' performance using comparative analysis. The accompanying statistical metrics were used in a comprehensive evaluation. Spearman's correlation is a statistical technique used to assess the degree of agreement between automated ratings and instructor assessments. Error metrics are used to quantify deviations from benchmark scores: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) (Kazem et al., 2019).

Table 3 highlights the model performance across various experiments, providing a systematic framework for identifying the optimal approach based on the balance between grading accuracy and reliability. These analyses form the foundation for the study's final recommendations, which aim to guide future efforts in developing more efficient Arabic automated grading systems capable of addressing the language's linguistic intricacies. From Table 3, we can validate the superiority of Stacking-based TF-IDF models in achieving the highest correlations (Pearson=0.73, Spearman=0.76) and lowest errors (MAE=0.81, MSE=0.96), highlighting the efficacy of combining lexical keyword frequency with stacking ensemble techniques.

While Word2Vec models achieved lower errors when integrated with LSA and LCS (MAE=1.02), they showed weaker alignment with human evaluation compared to TF-IDF. Jaccard methods improved with stemming but remained less efficient than Cosine-based approaches. LCS models differed from representation type (Word2Vec/TF-IDF) due to their reliance on structural alignment alone. Figure 4 presents the summary of results based on different evaluation methods.

5. Conclusion

This section aims to clarify the key findings from the experimental methods used in this study, which sought to identify the best framework for evaluating text similarity to predict students' academic performance. The study examined many text similarity evaluation methodologies in conjunction with LSA, LCS, cosine similarity, and Jaccard similarity. The methodology that united Jaccard's similarity with LSA, LCS using a stacking technique, and TF-IDF for text representation achieved the most promising results; it attained a Mean Absolute Error (MAE) of 0.81, a Mean Squared Error (MSE) of 0.96, and a Root Mean Squared Error (RMSE) of 0.98, indicating decreased error values. The correlation values were high, with Spearman = 0.76 and Pearson = 0.73. In conclusion, the findings of the study suggest that the choice of similarity measuring approach is unpredictable. TF-IDF is better at basic similarity comparison, but Word2Vec is better at complex text relationships; this highlights the need to choose the most appropriate method based on the needs of the study. In the future, the study plans to focus on expanding datasets, integrating advanced deep learning architectures such as BERT, and improving feedback mechanisms to enhance the effectiveness and reliability of assessment systems. This work confirms that integrating lexical, semantic, and contextual representations can enhance assessment accuracy and reliability, providing educational institutions with a scalable framework.

Table 3: Experimental Performance Analysis of Essay Grading Methods

Exp.	**Method**	**MAE**	**MSE**	**RMSE**	**Pearson Correlation**	**Spearman Correlation**
1	Word2Vec + Cosine Similarity + without stemming	1.29	2.71	1.65	0.45	0.46
	TF_IDF + cosine + without stemming	1.73	4.25	2.06	0.52	0.52
2	Word2Vec + Cosine Similarity + with stemming	1.38	2.99	1.73	0.55	0.55
	TF_IDF + cosine + with stemming	1.42	2.93	1.71	0.59	0.6
3	Word2Vec + Cosine Similarity + LSA + LCS + avg	1.02	1.52	1.23	0.61	0.6
	TF_IDF + cosine + LSA + LCS+ avg	1.19	2.07	1.44	0.61	0.61
4	Stacking (Word2Vec Cosine + LSA + LCS)	0.94	1.33	1.15	0.57	0.58
	Stacking (TF_IDF + LSA + Cosine Similarity + Avg)	0.87	1.13	1.07	0.65	0.65
5	Word2Vec + Jaccard Similarity + without stemming	2.3	6.73	2.59	0.48	0.54
	Jaccard + TFIDF + without stemming	2.3	6.73	2.6	0.48	0.53
6	Word2Vec + Jaccard Similarity + with stemming	1.98	5.13	2.26	0.57	0.62
	TFIDF + Jaccard + with stemming	2.71	9.1	3.02	0.57	0.62
7	Word2Vec + Jaccard Similarity + LSA + LCS + avg	1.89	4.72	2.17	0.57	0.6
	TF_IDF + Jaccard + LSA + LCS + avg	1.92	4.88	2.21	0.52	0.54
8	Stacking (Word2Vec + Jaccard Similarity + LSA + LCS)	0.85	1.12	1.06	0.66	0.64
	Stacking (TF_IDF + Jaccard + LCS + LSA)	0.81	0.96	0.98	0.73	0.76
9	word2vec + LSA	1.43	2.97	1.72	0.58	0.59
	TF-IDF + LSA	1	1.55	1.24	0.59	0.6
10	Word2Vec + LCS	2	5.22	2.28	0.55	0.59
	TF-IDF + LCS	2	5.22	2.28	0.55	0.59

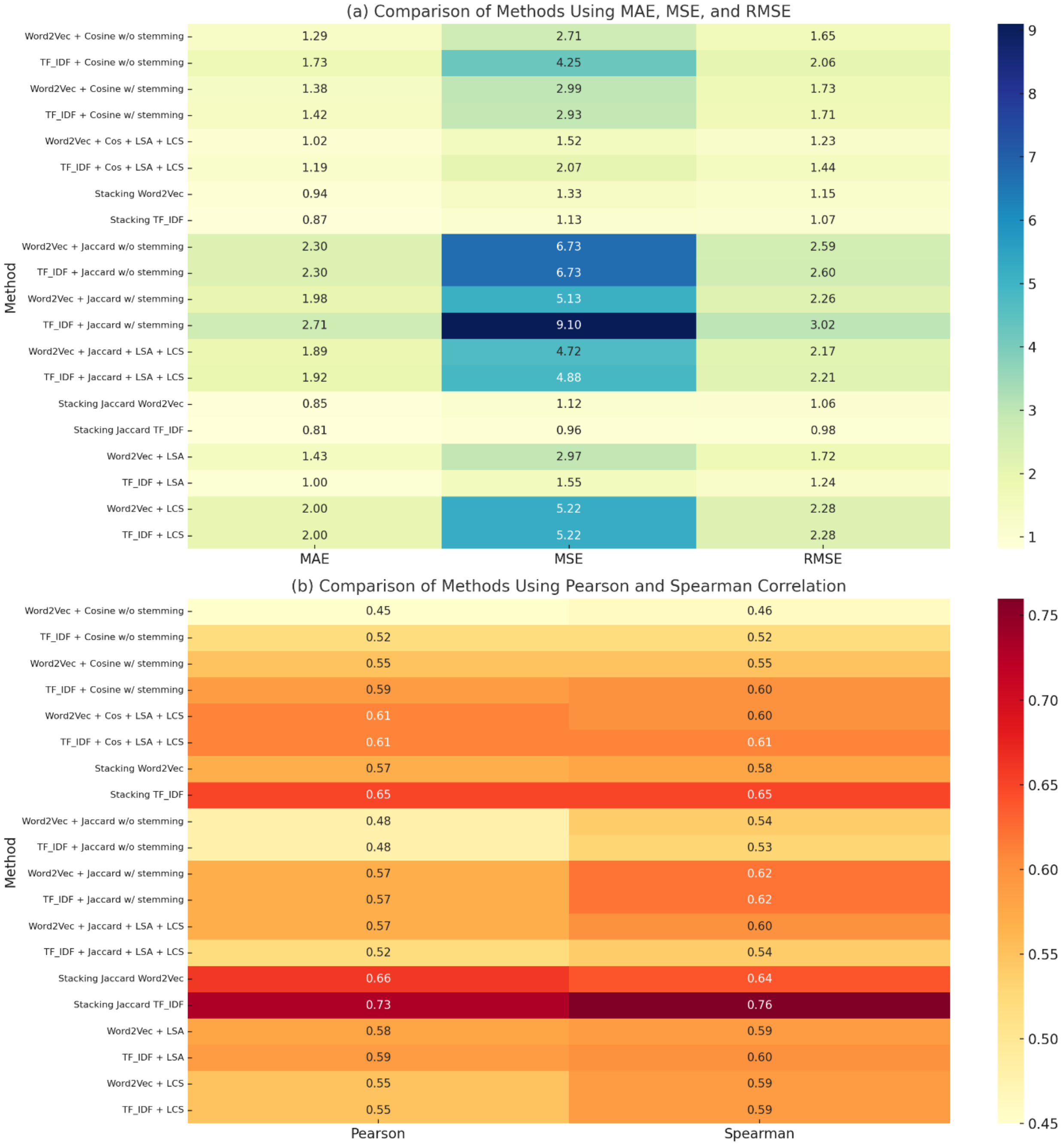

Figure 4: (a) compares different approaches using three performance metrics: MAE, MSE, and RMSE; (b) shows the results of the Pearson and Spearman effect.

Acknowledgment

The research leading to these results has received no Research Grant Funding.

Author contribution: All authors have contributed, read, and agreed to the published version of the manuscript results.

Conflict of interest: The authors declare no conflict of interest.

References

[1]. Al Awaida, S. A., Al-Shargabi, B., & Al-Rousan, T. (2019). Automated Arabic essay grading system based on F-score and Arabic WordNet. *Jordanian Journal of Computers and Information Technology, 5*(3).

[2]. Al-Awaida, S. E. O. (2019). *Automated Arabic essay grading system based on support vector machine and text similarity algorithm* [Master's thesis, Middle East University]. Middle East University Theses Repository. https://meu.edu.jo/libraryTheses/5d3c0808d27b7_1.pdf

[3]. Al-Shalabi, R., Kanaan, G., Jaam, J. M., Hasnah, A., & Hilat, E. (2004, April). Stop-word removal algorithm for Arabic language. In *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, 2004.* (p. 545). IEEE.

[4]. Al-Shargabi, B., Alzyadat, R., & Hamad, F. (2021). AEGD: Arabic essay grading dataset for machine learning. *Journal of Theoretical and Applied Information Technology, 99*(6).

[5]. Azmi, A. M., Al-Jouie, M. F., & Hussain, M. (2019). AAEE–Automated evaluation of students' essays in Arabic language. *Information Processing & Management, 56*(5), 1736–1752.

[6]. Badry, R. M., Ali, M., Rslan, E., & Kaseb, M. R. (2023). Automatic Arabic grading system for short answer questions. *IEEE Access, 11*, 39457-39465.

[7]. de Vos, I. M. A., Boogerd, G. L., Fennema, M. D., & Correia, A. D. (2022). Comparing in context: Improving cosine similarity measures with a metric tensor. *arXiv preprint arXiv:2203.14996*.

[8]. Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP), 8*(4), 1–22.

[9]. Gomaa, W. H., & Fahmy, A. A. (2020). Ans2Vec: A scoring system for short answers. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019)* (Vol. 4, pp. 586–595). Springer.

[10]. Jiwani, N., Gupta, K., & Whig, P. (2022, November). Analysis of the potential impact of omicron crises using NLTK (natural language toolkit). In *Proceedings of Third Doctoral Symposium on Computational Intelligence: DoSCI 2022* (pp. 445-454). Singapore: Springer Nature Singapore.

[11]. Kazem, H. A., Chaichan, M. T., & Yousif, J. H. (2019). Evaluation of oscillatory flow Photovoltaic/Thermal system in Oman. *Int. J. Comput. Appl. Sci*, *6*(1).

[12]. Kazem, H. A., Yousif, J. H., Chaichan, M. T., Al-Waeli, A. H., & Sopian, K. (2022). Long-term power forecasting using FRNN and PCA models for calculating output parameters in solar photovoltaic generation. *Heliyon*, *8*(1).

[13]. Larkey, L. S., Ballesteros, L., & Connell, M. E. (2002, August). Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 275-282).

[14]. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of massive datasets*. Cambridge University Press

[15]. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

[16]. Omran, A. M. B., & Ab Aziz, M. J. (2013). Automatic essay grading system for short answers in English language. *Journal of Computer Science, 9*(10), 1369.

[17]. Ouahrani, L., & Bennouar, D. (2020). AR-ASAG: An Arabic dataset for automatic short answer grading evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 2634–2643).

[18]. Ramalingam, V. V., Pandian, A., Chetry, P., & Nigam, H. (2018, April). Automated essay grading using machine learning algorithm. In *Journal of Physics: Conference Series* (Vol. 1000, p. 012030). IOP Publishing.

[19]. Sharma, G., & Singh, P. (2024). Comparative study: Word2Vec versus TF-IDF in software defect predictions. In *International Conference on Data Science and Network Engineering* (pp. 95–107). Springer Nature Singapore.

[20]. Shehab, A., Faroun, M., & Rashad, M. (2018). An automatic Arabic essay grading system based on text similarity Algorithms. *International Journal of Advanced Computer Science and Applications*, *9*(3).

[21]. Thongtan, T., & Phienthrakul, T. (2019). Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop* (pp. 407–414).

[22]. Yousif, J. (2018). Neural computing based part of speech tagger for Arabic language: a review study. *International Journal of Computation and Applied Sciences IJOCAAS*, *5*(1).

[23]. Yousif, J. H., & Sembok, T. M. T. (2008, August). Arabic part-of-speech tagger based Support Vectors Machines. In *2008 International Symposium on Information Technology* (Vol. 3, pp. 1-7). IEEE.

[24]. Yousif, J. H., & Yousif, M. J. (2024). Evolutionary Perspectives on Neural Network Generations: A Critical Examination of Models and Design Strategies. *Current Computer Science*, *3*(1), E050424228693.

[25]. Yousif, J. H., & Yousif, M. J. (2024). Evolutionary Perspectives on Neural Network Generations: A Critical Examination of Models and Design Strategies. *Current Computer Science*, *3*(1), E050424228693.

[26]. Zhan, Z. (2025). Comparative analysis of TF-IDF and Word2Vec in sentiment analysis: A case of food reviews. In *ITM Web of Conferences*. EDP Sciences, 02013.

[27]. Zhou, J., Ye, Z., Zhang, S., Geng, Z., Han, N., & Yang, T. (2024). Investigating response behavior through TF-IDF and Word2vec text analysis: A case study of PISA 2012 problem-solving process data. *Heliyon*, *10*(16).