E-ISSN: 2997-9382
American Journal of Technology Advancement
https://semantjournals.org/index.php/AJTA

| Research Article

Check for updates

Off-Topic Response Identification in an Arabic-Language Auto-Grading Platform

Hanadi Hamed Abdel Rahman, Salma A. Mahmood
Department of Computer Science, Basra University, Basra, Iraq

Annotation

Automatic short-answer assessment plays a crucial role in modern educational technology by providing scalable and consistent evaluation of student responses. However, traditional similarity-based approaches often fail to capture the deeper semantic meaning of answers, limiting their effectiveness. This study investigates the performance of a fine-tuned AraBERT v2 model for Arabic automatic assessment using the AR-ASAG dataset. The model processes student answers by preprocessing, tokenizing, and extracting contextual representations from the [CLS] token, with a Dropout probability of 0.3 to reduce overfitting. Experimental results show that the fine-tuned AraBERT v2 model significantly outperforms traditional cosine similarity methods. Specifically, cosine similarity between student and model answers achieved a Mean Squared Error (MSE) of 0.1834 and a Pearson correlation of 0.3098, while cosine similarity with the question yielded weaker results with an MSE of 0.1640 and a correlation of 0.0139.

In contrast, the fine-tuned AraBERT v2 achieved superior performance, reaching an MSE of 0.0187 and a Pearson correlation of 0.8866 after three training epochs.

These results demonstrate the model's high robustness, its effectiveness in capturing deep semantic similarity, and its reliability as a solution for automatic short-answer assessment in Arabic educational contexts.

Keywords: Large Language models, AraBERT, Automatic Assessment systems, off-topic detection.

This is an open-access article under the CC-BY 4.0 license

1. Introduction

Essays are widely used to learn assessment in an educational context. Automated Essay Scoring is widely used to assess students' writing skills. The essay test assesses the competencies developed by the student and promotes improvements in communication and expression skills. In an essay, a statement is presented with a (quick) objective suggestion to the student, demanding the development of a descriptive, narrative, or dialectical textual response. Analyzing such texts is not a trivial task. Essay writing is an important tool for developing and assessing students' cognitive abilities, including critical thinking, communication skills, and depth of understanding. The teacher may spend a lot of time evaluating the content of the essay, since there are many textual features that need to be analyzed. However, as student numbers grow, manually tagging essays becomes impractical, discouraging the use of essay questions in education.

Digital educational platforms have become widespread in recent years, increasing the need for automated assessment systems that are able to provide efficiency, fairness, and speed in correcting students' answers, especially in large-scale learning environments. Despite the advantages of automated assessment, one of the most prominent challenges it faces is discovering off-topic answers, as the answer may be written in correct language or in a structured manner, but it does not reflect the content required in the question. Such answers may lead to a reduction in the accuracy of the assessment and weaken the reliability of the results.

In Automated Essay Scoring, Off-topic refers to something that has little or nothing to do with what is being discussed at present. If something is off-topic, it is not relevant to the topic or subject to be discussed. There are various cases in which the discussion might encounter this word, but it essentially means the same thing. Off-topic essay detection has been applied in education in terms of an automated checking and verification tool for teachers. In a broader context, it can also be used by students as an essay checking tool, just like there are tools for checking spellings and grammar in text.

From the deep learning and machine learning paradigms' views, off-topic essay detection is considered a classification problem. In that, off-topic and on-topic essays are first human-scored and then assigned a class label of 0 or 1. Then the essays, model answer, and class labels are used to train a classifier. For this process to be used practically, there is always a need for high precision from off-topic detection systems to prevent cases where off-topic essays are labeled as on-topic. An off-topic text detection pipeline usually has the following steps. First, all essay samples, i.e., both the prompts and responses present in the dataset, are split into words. They go through a process called stemming, which can be crudely stated as reducing words to their stem form. Each word is encoded into a numeric form representation. Then, every corresponding prompt and response words are combined using an embedding scheme to extract word-embedding-based features. These features usually assess how similar or dissimilar the question and response are. Finally, these features are used to train a classifier.

The field of discovering off-topic answers has received increasing attention in global languages such as English, but studies in Arabic are still limited. This is due to the complexity and richness of the Arabic language, including its multiplicity, the diversity of morphological and grammatical structures, as well as its semantic richness. These characteristics make the process of building accurate Arabic models to detect irrelevant answers a challenge that requires advanced and integrated solutions.

Automated assessment systems for Arabic text face significant challenges in distinguishing between on-topic and off-topic responses, as these systems often focus primarily on linguistic correctness or semantic structure without adequately evaluating the relevance of the answer to the posed question. This limitation reduces the accuracy and fairness of assessment, particularly in the context of essay-based educational tests. Therefore, there is a pressing need to develop an advanced automated model capable of detecting off-topic responses in Arabic texts, which can enhance evaluation accuracy, ensure assessment objectivity, and support applications in e-learning and natural language processing.

This research aims to develop an automatic detection model for off-topic answers in Arabic assessments using modern methods in NLP and deep learning using AraBERTv2. It also seeks to integrate the distinctive linguistic characteristics of Arabic with intelligent models capable of improving the accuracy of predicting inappropriate answers. The results of this work will contribute to building more reliable and fair automated evaluation systems. It will support the educational process and make it easier for teachers, students, and educational institutions to benefit from better digital assessment tools.

2. Related Works

The study presented by **Areeba Shahzad et. al (2022)** aims to address the problem of predicting whether the essay is off-topic or not using machine learning techniques. With an increase in online learning and evaluation platforms, especially during the COVID-19 pandemic, the off-topic detection system can be very useful to check essays that are mainly submitted online. In this paper, they answer the question: given a prompt and an essay written in Pakistani English, can the process of detecting whether the essay is off-topic or not be reliably and completely autonomized with zero human intervention using currently available tools and techniques? For this purpose, they explore and implement various embedding techniques proposed to extract the similarity or dissimilarity features between the question and response. They compare their work performance by using 10 benchmark datasets and 6 classifiers. With different classifiers and different datasets, along with different embeddings, they conclude that the combination of word mover distance, average, and weighted word embeddings together with random forest as the classifier is the best combination for off-topic essay detection. They obtained an accuracy of 93.5%

The study of **Rayed Ghazawi et. al (2024)** introduces AR-AES, an Arabic AES benchmark dataset. It comprises 2046 undergraduate essays, including gender information, scores, and transparent rubric-based evaluation guidelines, providing comprehensive insights into the scoring process. These essays come from four diverse courses, covering both traditional and online exams. Additionally, they pioneer the use of AraBERT for AES, exploring its performance on different question types. Their results were encouraging, particularly for chemistry, environmental source-dependent essay questions. For the first time, they examine the scale of errors made by a BERT-based AES system, observing that 96.15% of the errors are within one point of the first human marker's prediction, on a scale of one to five, with 79.49% of predictions matching exactly. In contrast, additional human markers did not exceed 30% exact matches with the first marker, with 62.9% within one mark. These findings highlight the subjectivity inherent in essay grading, and underscore the potential for current AES technology to assist human markers to grade consistently across large classes

Paper of **Sourya Dipta Das et. al (2024)**, proposing an unsupervised technique that jointly scores essays and detects off-topic essays. The proposed Automated Open Essay Scoring (AOES) model uses a novel topic regularization module (TRM), which can be attached on top of a transformer model, and is trained using a proposed hybrid loss function. After training, the AOES model is further used to calculate the Mahalanobis distance score for off-topic essay detection. The proposed joint model demonstrated superior performance compared to traditional methods; it achieved an accuracy of approximately **90%** in off-topic detection, while the **Quadratic Weighted Kappa (QWK)** correlation between the automated scores and human raters in AES reached around **0.78–0.82,** reflecting a strong alignment with expert judgments. The significance of this study lies in its ability to jointly model both tasks, reducing training complexity and enhancing overall performance. This Experimental evaluation results on different adversarial strategies also show how the suggested method is robust for detecting possible human-level perturbations.

The study presented by Yiting Zhu (2021) proposed an off-topic detection method for business English essays based on a deep learning model. In this approach, the Word2Vec model was employed to represent words within sentences as numerical vectors, while Latent Dirichlet Allocation (LDA) was utilized to extract topic and text vectors separately. The word vectors and topic word vectors were then concatenated and fed into a Convolutional Neural Network (CNN), which was responsible for extracting and filtering sentence-level features and performing similarity calculations. When the similarity value fell below a specified threshold, the study additionally mapped the topic and subject words in a coupling space to calculate their relevance. Ultimately, unsupervised off-topic detection was achieved through a clustering method.

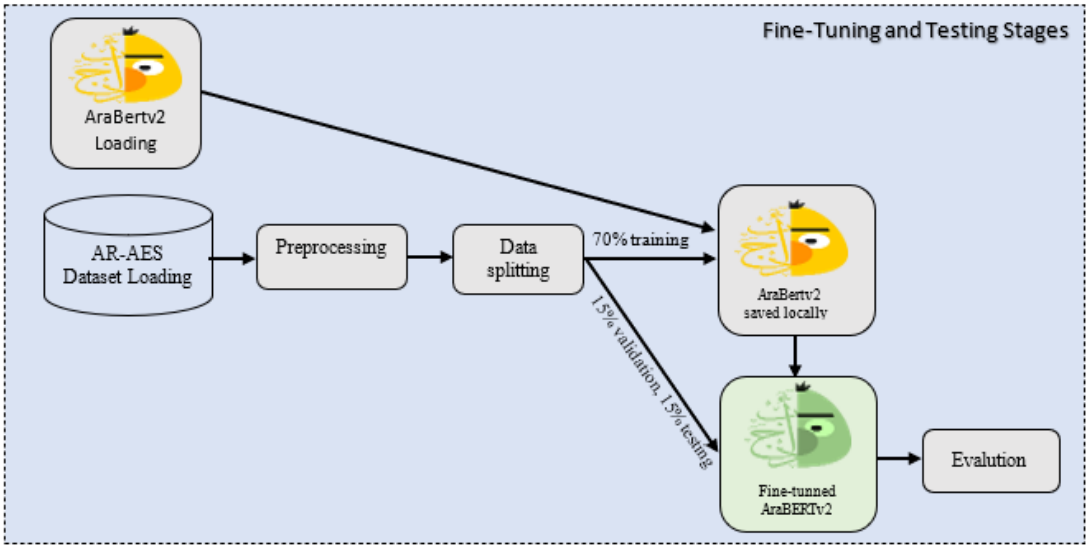
Experimental results demonstrated that the proposed deep learning–based approach achieved 83% accuracy using LDA, improving detection performance for essays with both low and high topic divergence, particularly in cases with significant divergence.

Similarly, the research conducted by Tsegaye Misikir Tashu (2022) introduced a method to validate user responses through an off-topic essay detection model integrated into automatic essay evaluation systems. The study implemented a C-BGRU Siamese architecture, in which the convolutional layer learned and captured contextual n-gram features from word embedding vectors, while a bidirectional gated recurrent unit (BGRU)—a variant of recurrent neural networks—was employed to capture both preceding and succeeding contextual representations. The experiments were carried out on eight Kaggle-based datasets for automated essay evaluation tasks. The results indicated that the proposed model achieved strong performance, with recall, precision, and F1-score values ranging from 79.35% to 91.71% using a multi-channel CNN with pooling, and from 48.10% to 91.71% using the content vector analysis method, demonstrating a significant improvement over baseline methods.

In addition, the study presented by Xinhao Wang et al. (2019) aimed to address the challenge of detecting off-topic spoken responses by developing an automatic system that could support downstream automated scoring processes. The proposed method interpreted the relationship between a test response and its corresponding question as a similarity grid, which was then analyzed using very deep convolutional neural networks to determine different levels of topic relevance. In this work, Inception networks were applied to the task, and the experimental results demonstrated the effectiveness of the proposed approach. The system achieved an F1-score of 92.8% for detecting off-topic responses, significantly outperforming a baseline model that utilized word embedding–based similarity metrics, which achieved an F1-score of 85.5%.

3. Research Methodology

This study aims to develop an automated system for detecting Arabic off-topic texts in short-answer questions, such as 'define,' 'answer briefly,' etc. The Standard Arabic dataset AR-AES [1]was used, which has been widely employed in Arabic research on automated assessment. The study utilized the AraBERTv2 model to extract answer embeddings through two main stages. The first stage involved fine-tuning and testing the language model. In contrast, the second stage focused on using the fine-tuned AraBERTv2 on previously unseen data to predict whether a new unseen answer falls within the required model answer (On-topic) or is outside the context of the necessary answer (Off-topic). The following figure (1) shows the general framework of the proposed off-topic detection system.



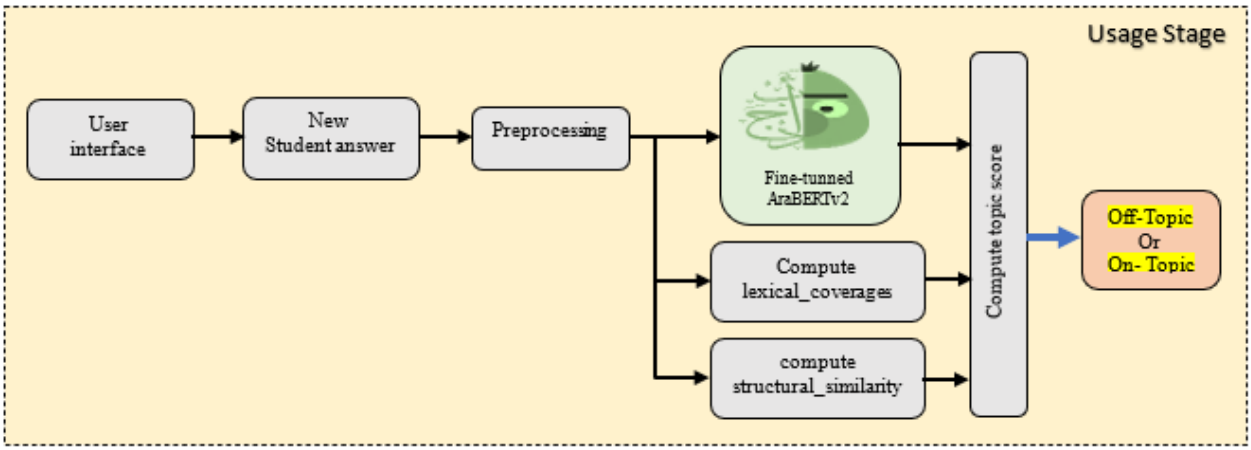


Figure 1. Proposed Off-topic Detection System

Each stage of the proposed system comprises several processes, which are detailed in the following section.

3.1 AR-ASAG Dataset description

The AR-ASAG dataset is a valuable resource for performance evaluation by other researchers in the fields of short-answer grading and Arabic semantic similarity. It consists of questions extracted from the teaching course on cybercrimes with answers provided by three classes of master students. About 170 students with native Arabic language proficiency followed the course, which was validated by an official exam. AR-ASAG Dataset is freely available in different versions: TXT, XML, XML-MOODLE, and Database (.DB). The reported evaluations relate to answers submitted to three exams for three classes of students. The exams were conducted under natural evaluation conditions. Each test consists of 16 short-answer questions (a total of 48 questions), 2133 records total. Table 1 summarizes the dataset description. The Dataset encompasses five types of questions:

- ✓ "عرّف" (Define)
- ✓ "اشرح" (Explain)
- ✓ "ما النتائج المترتبة على" (What are the consequences)
- ✓ "علّل" (Justify)
- ✓ "ما الفرق" (What is the difference)

Table 1. AR-ASAG Dataset

Attribute	Value
Dataset Name	AR-ASAG
Language	Arabic
Number of Exams	3
Questions per Exam	16
Total Questions	48
Total Student Answers	2133
Question Types	Define, Explain, Justify, Consequences, Difference
Formats Available	TXT, XML-MOODLE, .DB
Grading Method	Two human scores + average

3.2 AraBERTv2 Fine-Tuned and Testing Stage

3.2.1 Data preprocessing

Preprocessing is an essential stage that must be performed before any other processes to avoid errors, improve performance, and reduce bias. This process contains the following steps:

- **Removing Missing Values:** Any rows containing missing data, such as the question, model answer, student answer, or score, are removed. This ensures that all data used for training and testing is complete, preventing errors during text processing or model training and increasing the reliability of the results.
- **Score Normalization:** The original scores are converted to values between 0 and 1 by dividing by the maximum score (e.g., 5). This aligns the data with the Sigmoid function's output range and helps the model learn more effectively, avoiding issues caused by significant value variations.
- **Creating Text Pairs:** Pairs of texts are generated to represent relationships between the student answer and the question or between the model answer and the student answer, such as (Question – Student Answer) and (Model Answer – Student Answer). This increases the dataset size and enables the model to learn diverse relationships among texts, thereby improving prediction accuracy and evaluation performance.
- **Text Tokenization:** Arabic texts are converted into numerical representations that the AraBERT model can understand. Long texts are truncated to a fixed length, while shorter texts are padded, and attention masks are created to differentiate actual tokens from padding. This prepares the data consistently for BERT and allows effective processing of all sentences.
- **Using the [CLS] Embedding:** The [CLS] token embedding from BERT outputs is extracted, which serves as a comprehensive summary of the sentence. This embedding is used for predicting scores or evaluating text similarity, as it captures the most essential information from the entire sentence.

3.3.2 Data Splitting

The data splitting stage involves dividing the data into three sets: a 70% training set, a 15% validation set, and a 15% test set. This standard split ensures the optimization and accurate evaluation of the linguistic model's performance.

3.3.3 AraBERTv2 Training

AraBERTv2 is an advanced Arabic language model built upon the BERT architecture and specifically optimized for the Arabic language. It was pre-trained on a large-scale corpus of approximately 200 million sentences, amounting to 8.6 billion words (77 GB). During pre-training, extensive preprocessing was applied, including the insertion of spaces between numbers, letters, and punctuation marks, as well as morphological segmentation using the Farasa tool to handle prefixes and suffixes. This preprocessing significantly reduced morphological sparsity and enhanced vocabulary representation.

The model is available in three main variants: Base (136 million parameters, 543 MB), Large (371 million parameters, 1.38 GB), and Twitter-specific, which was trained on about 60 million Arabic tweets. AraBERTv2 has demonstrated state-of-the-art performance across various Arabic NLP tasks, including sentiment analysis, named entity recognition, and topic classification, consistently outperforming multilingual models such as mBERT.

In this study, AraBERTv2 was adopted due to its effectiveness as one of the most powerful Arabic-oriented transformer models. Its large-scale pre-training and advanced preprocessing make it highly suitable for automated detection of off-topic responses in Arabic essay scoring systems.

The pre-trained AraBERTv2 model was downloaded from Hugging Face and stored locally to facilitate faster training and inference. The configuration parameters used in this research were based on the Base version of the model.

W. Antoun, F. Baly, and H. Hajj, "AraBERTv2: Arabic language model based on BERT architecture," *Hugging Face*, 2021. [Online]. Available: https://huggingface.co/aubmindlab/bert-base-arabertv2

class AraBERTRegressor(nn.Module):

```
def __init__(self, model_path):
super().__init__()
self.bert = AutoModel.from_pretrained(model_path)
self.dropout = nn.Dropout(0.3)
self.linear = nn.Linear(self.bert.config.hidden_size, 1)
self.sigmoid = nn.Sigmoid()

def forward(self, input_ids, attention_mask):
outputs = self.bert(input_ids=input_ids, attention_mask=attention_mask)
cls_output = outputs.last_hidden_state[:, 0, :]
x = self.dropout(cls_output)
x = self.linear(x)
return self.sigmoid(x).squeeze(-1)
```

The fine-tuning process was performed with the following settings: BATCH_SIZE = 8, EPOCHS = 3, LR = 2e-5, and Dropout(0.3) to mitigate overfitting. Upon completion of this process, the fine-tuned AraBERTv2 model will be trained on the new assessment task. This will enable it to capture the semantic contexts of the words and sentences that constitute the questions and their model answers, on the one hand, and the contextual and semantic representations of the students' answers and the model answer, on the other hand.

3.3.4 Arabert v2 validation and testing

3.3.5 Evaluation Metrics

- **Pearson Correlation Coefficient:** The Pearson Correlation Coefficient measures the strength and direction of a linear relationship between two quantitative variables. It ranges from -1 (perfect negative linear correlation) to +1 (perfect positive linear correlation), with 0 indicating no linear relationship. A value close to +1 or -1 shows a strong linear association, while a value near 0 indicates a weak or non-existent linear association Utilizing Equation (1).

$$R = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}} \quad \ldots(1)$$

- **Mean Squared Error (MSE):** The use of Mean Squared Error is pertinent in situations requiring the detection of outliers. MSE is notably proficient at attributing increased significance to the data points. Utilizing Equation (4).

$$MSE = \frac{1}{m} \sum (Xi - Yi)^2 \quad \ldots(2)$$

3.3.6 Off-topic Detection

- **Integrating Lexical Coverage and Structural Similarity with AraBERTv2**

While AraBERTv2 demonstrates strong contextual understanding and classification capabilities, it may struggle to evaluate short or atypical responses—especially those that include correct terms or scoring students' answers but are structurally incoherent or off-topic.

This issue highlights the strategic value of integrating lexical-structural analysis with AraBERTv2 outputs to enhance evaluation accuracy by combining AraBERTv2's deep contextual modeling with this program's granular structural analysis, detection of Off-Topic Responses, so even if a student uses correct terms, low structural similarity can flag irrelevant or incoherent answers, in addition, improved objectivity in scoring through quantifiable metrics that reduce human bias and support consistent evaluation, and finally, give pedagogical insight to teachers to better understand whether a student's issue lies in vocabulary usage or structural formulation, enabling more targeted feedback.

To solve this challenge, we presented a code that analyzes student responses using two precise metrics: - **Lexical Coverage**, which measures the overlap of individual words between the model answer and student responses.

- **Structural Similarity**, which evaluates the overlap of multi-word phrases, reflecting the syntactic coherence between the texts.

The code fragment employs a subsequence-matching algorithm to extract shared sequences between the model answer and each student response. These sequences are then categorized into single-word and multi-word segments, and their proportions are calculated relative to the average length of both texts. This approach enables a nuanced evaluation not only at the lexical level but also at the structural level, allowing the system to detect responses that may contain correct vocabulary but lack coherence or relevance, and offers a precise analytical tool that complements the capabilities of deep language models like AraBERTv2. Integrating its results represents a strategic advancement toward building a more comprehensive and reliable automatic assessment system—especially in Arabic-language educational environments. This hybrid approach helps overcome the limitations of contextual models alone and strengthens the system's ability to distinguish between correct, partial, and off-topic responses. The pseudo-code for structural similarity computing is as follows:

Function: find_common_subsequences(text1, text2)

Steps:

1. Split both texts into word lists.
2. Initialize an empty set for common sequences.
3. For each word position in text1:

✓ For each word position in text2:

✓ While words match at current positions:

✓ Extend the matching sequence.

✓ Add the sequence to the common set.

4. Sort the common sequences by length (longest first).
5. Return the sorted list.

The following code fragment to find lexical coverage between two answers is as follows:

Function: lexical_structural_scores(text1, text2)

Steps:

1. Call find_common_subsequences(text1, text2).
2. Separate results into:

✓ Single-word matches → lexical units.

✓ Multi-word matches → structural units.

3. Calculate:

✓ Total lexical units.

✓ Total structural units.

✓ Average length of both texts.

4. Compute:

✓ Lexical coverage = (lexical units / average length) × 100

✓ Structural similarity = (structural units / average length) × 100

5. Return both scores.

- **Compute topic score**

This function plays a critical role in hybrid answer evaluation. While semantic models like AraBERTv2 provide contextual predictions, they may overlook lexical gaps or structural incoherence. By integrating: Semantic prediction (50%), Lexical match (25%), and Structural alignment (25%). The function ensures a balanced and interpretable score that reflects both meaning and form. This is especially valuable in educational or linguistic systems where off-topic but fluent answers must be flagged, and partially correct responses need nuanced grading. The Pseudo-code of the topic score computation function is as follows:

Function: compute_topic_score(predicted_score, lexical_cov, structural_sim)

Inputs:

✓ predicted_score: model's semantic prediction score (range: 0–5)

✓ lexical_cov: lexical coverage percentage (0–100)

✓ structural_sim: structural similarity percentage (0–100)

Steps:

1 .Normalize all inputs to a 0–1 scale:

✓ pred = predicted_score * 2 # assuming original scale is 0–5

✓ lex = lexical_cov / 100

✓ struct = structural_sim / 100

2 .Compute weighted average:

✓ topic_score = (pred * 0.5) + (lex * 0.25) + (struct * 0.25)

3 .Round and return the final score:

✓ return round(topic_score, 4)

Output:

✓ topic_score: a unified score (0–1) reflecting semantic, lexical, and structural relevance.

4. Results and Discussion

The following sections systematically outline the study results. All experimental work was performed on a **HUAWEI BOHK-WAX9X system** equipped with an **AMD64 processor (2.10 GHz)** and **7 GB of RAM**, operating under **Microsoft Windows 11 Home (Build 26100)**. The proposed framework was developed in **Python 3.11.11** using the **Spyder 6.0.5 IDE**, with implementation relying on various **TensorFlow libraries**.

The proposed off-topic detection system consists of two main phases: (1) fine-tuning the **AraBERTv2** model, followed by testing and evaluation; and (2) deployment through a user interface to enhance reliability, generalizability, and practical validation.

4.1 AraBERTv2 Fine-tuning

The AraBERTv2 fine-tuning passed over three training epochs. The figure (2) presents a comprehensive view of the model's learning dynamics over three training epochs, using three key performance metrics: training loss, validation mean squared error (MSE), and validation Pearson correlation coefficient.

The first plot demonstrates a consistent decline in training loss, indicating that the model is effectively minimizing its internal error and improving its fit to the training data. This downward trend reflects successful gradient descent optimization and suggests that the model is learning meaningful patterns without stagnation.

The second plot shows a parallel reduction in validation MSE, a critical metric for assessing generalization. The decreasing error on unseen validation data confirms that the model is not merely memorizing the training set but is capable of producing accurate predictions on new inputs. This alignment between training and validation performance suggests a well-regularized model with low risk of overfitting.

The third plot highlights an upward trend in the Pearson correlation coefficient between predicted and actual values on the validation set. This metric captures the model's linear relationship and prediction consistency. The increase from approximately 0.82 to above 0.87 indicates that the model's outputs are becoming more statistically aligned with the actual targets, reinforcing its reliability in regression tasks.

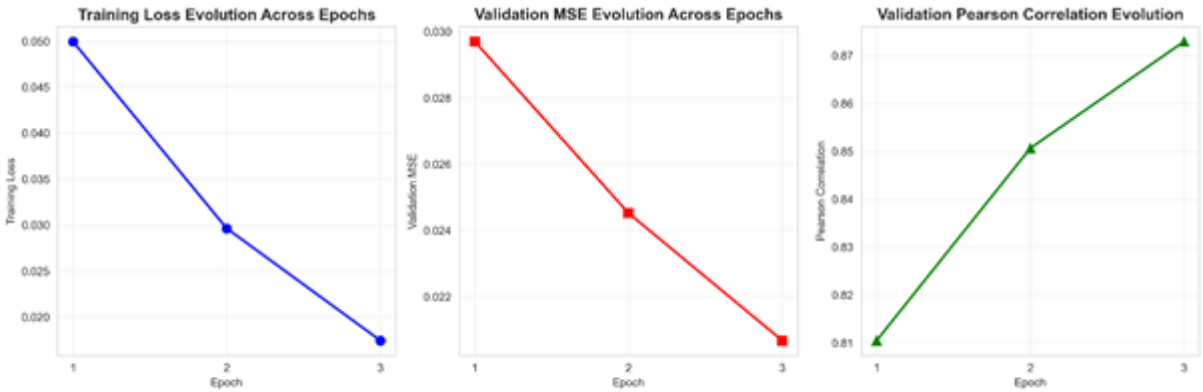


Figure (2) Epoch-wise Evaluation of Training Dynamics Using Loss, Validation MSE over AraBERTv2 fine-tuning Pearson Correlation

Collectively, these trends validate the model's robustness and learning efficiency. The simultaneous improvement across all three metrics supports the model's readiness for further

evaluation or deployment, particularly in domains requiring high predictive accuracy and interpretability, such as clinical forecasting or biomedical research.

4.2 Performance Evaluation

During the training process, performance metrics were closely monitored, and the results reflect a gradual and notable improvement in model performance across three training epochs, based on three key indicators:

1. Training Loss: The loss decreased from 0.0512 in the first epoch to 0.0174 in the third, indicating that the model is learning effectively and minimizing internal error. This rapid decline suggests a stable optimization process, with no signs of gradient instability or oscillation.

2. Validation Mean Squared Error (MSE): Validation MSE dropped from 0.0261 to 0.0187, demonstrating the model's enhanced ability to generalize to unseen data. The alignment between decreasing training loss and validation error indicates that the model is not overfitting, but instead achieving a healthy balance between learning and generalization.

3. Validation Pearson Correlation: The correlation coefficient increased from 0.8380 to 0.8866, suggesting that the predictions are becoming more linearly consistent with the actual values. This improvement in statistical alignment reinforces the model's reliability for high-precision predictive tasks, particularly in clinical or research contexts. The measurements were recorded across the training:

```
Starting training...
Epoch 1/3 - Training: 100%|██████████| 295/295 [00:17<00:00, 16.76it/s]
Epoch 1: Train Loss = 0.0512 | Val MSE = 0.0261 | Val Pearson = 0.8380
Epoch 2/3 - Training: 100%|██████████| 295/295 [00:17<00:00, 17.03it/s]
Epoch 2: Train Loss = 0.0250 | Val MSE = 0.0246 | Val Pearson = 0.8463
Epoch 3/3 - Training: 100%|██████████| 295/295 [00:17<00:00, 16.48it/s]
Epoch 3: Train Loss = 0.0174 | Val MSE = 0.0187 | Val Pearson = 0.8866
```

The AraBERTv2 model underwent three key processing phases—training, validation, and testing—during which its performance and efficiency were monitored using two primary metrics: Mean Squared Error (MSE), which reflects the deviation of the model's predictions from expert human judgments, and the Pearson correlation coefficient, which indicates the degree of linear alignment between them.

The figure (3) below presents a quantitative comparison of the model's performance across three independent datasets. The results show that the model achieved the lowest MSE on the training set, indicating a high level of agreement between predictions and the data used for learning. However, a gradual increase in MSE was observed when moving to the validation and test sets, suggesting a relative decline in predictive accuracy as the data became less familiar. This gradient in error is a typical indicator of the model's generalization capacity, though it warrants careful monitoring on the test set to ensure performance remains within acceptable bounds.

Regarding the Pearson correlation coefficient, the model recorded its highest value on the training set (approximately 0.9), reflecting a strong linear relationship between predicted and actual values. Although this value declined slightly in the validation and test sets, it remained above 0.8, indicating statistically consistent performance even on unseen data. This stability across all three datasets reinforces the model's reliability and structural robustness, making it suitable for high-precision predictive applications, particularly in clinical or research settings that demand both accuracy and generalizability.

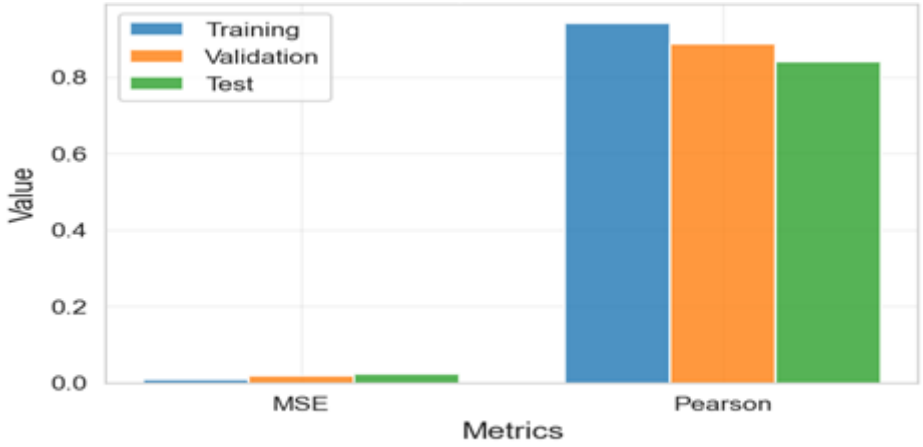


Figure (3) A quantitative comparison of the model's performance

4.3 Scores Distribution Across Different Stages

The score distribution results across the system's stages demonstrate a high learning efficiency, as the pie charts reveal a clear convergence in score proportions across datasets. This indicates the model's ability to maintain a consistent predictive pattern throughout the training, validation, and testing phases. The pie charts in figure (4) illustrate the distribution of scores across different stages of the model's processing pipeline, from the original average marks prior to training to the predicted scores on the training, validation, and test sets. The initial distribution reveals a concentration in the lower and mid-range score categories, with the 2.4–4.8 range representing the largest proportion (27.5%), indicating a general tendency toward limited performance before model intervention. Following training, the distribution shifts noticeably, with higher score ranges (6.8–8.2 and 8.2–10) increasing to 29.5% and 16.4%, respectively, in the training set—suggesting that the model successfully elevated predicted scores toward stronger performance levels.

This upward shift is sustained in the validation set, where the 6.8–8.2 range rises to 24.5%, reflecting the model's ability to generalize beyond the training data. In the test set, the higher score categories remain prominent (25.1% for 6.8–8.2 and 19.4% for 8.2–10), indicating structural stability and predictive reliability even on unseen data.

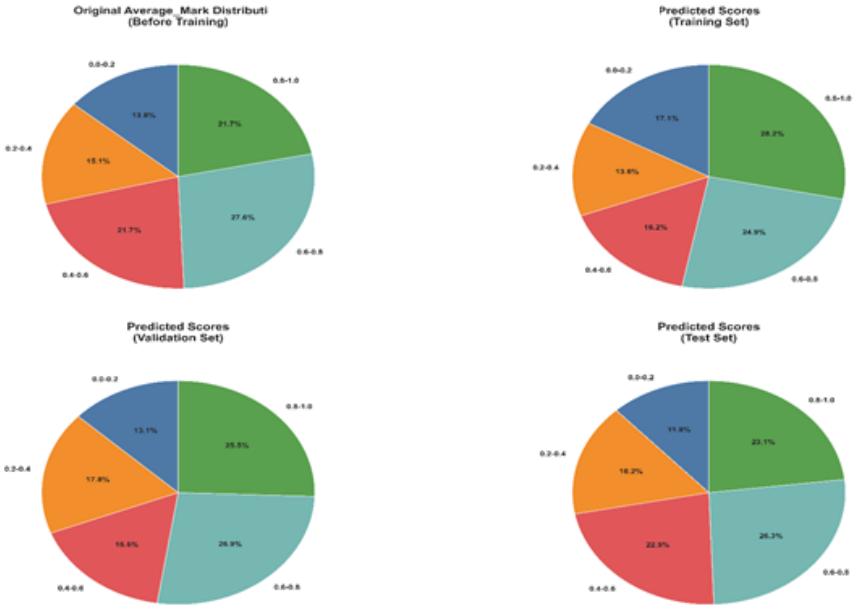


figure (4) Scores distribution across different stages of the model's processing pipeline

Overall, the transformation in score distribution across stages demonstrates the model's effectiveness in enhancing predictive outcomes and supports its applicability in domains requiring precise evaluation—whether educational, clinical, or research-oriented.

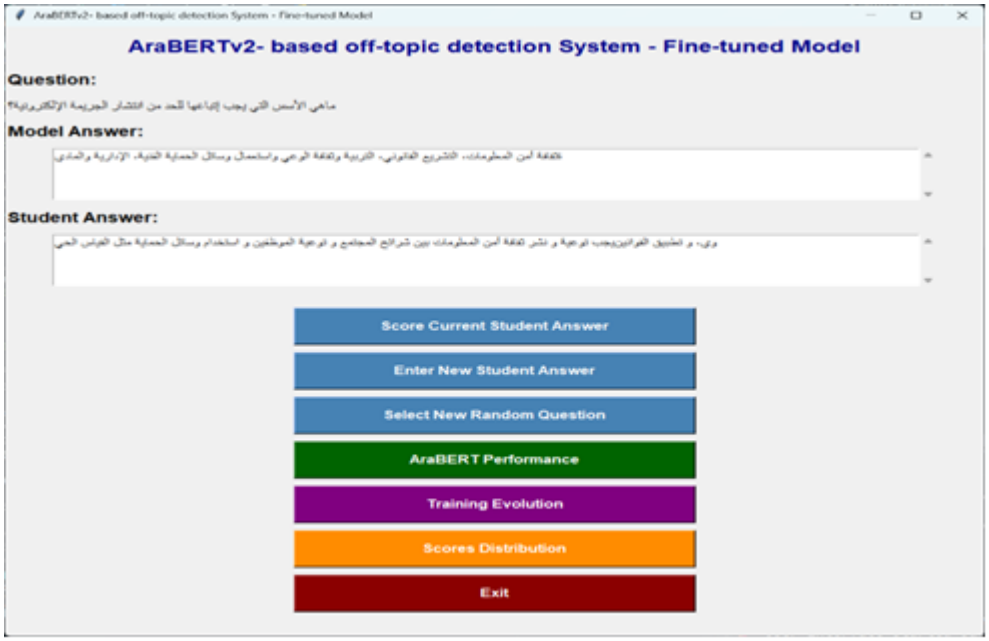


Figure (5) Digital interface for user interaction.

4.4 User Interface

The user interface (UI) is a fundamental component of interactive system design, serving as the direct point of contact between the user and the system. It plays a vital role in facilitating access to functionalities, enhancing user experience, and improving the efficiency of human–software interaction. A well-structured and intuitive interface goes beyond aesthetics—it encompasses usability, responsiveness, and evaluation accuracy. The displayed layout figure (5) presents a digital interface designed specifically for assessing student responses to academic questions. It includes core components such as the question prompt, a model answer, and the student's submitted answer, along with a set of interactive buttons that allow for scoring, entering new responses, or viewing evaluation results. This type of interface contributes to automating the grading process and enhances the objectivity of evaluation by identifying irrelevant content in student answers. It also enables users—whether educators or students—to interact with the system in a direct and seamless manner, fostering a more effective and transparent digital learning environment.

The interactive buttons displayed on the user interface are a central component of the system's design, enabling users to navigate between various evaluation functions smoothly and efficiently. The "Score Current Student Answer" button allows for immediate assessment of the entered response, accelerating feedback and minimizing manual intervention. The "Enter New Student Answer" button allows entering a new response without reloading the system, enabling continuous use. The "Select New Random Question" button introduces question variety, enhancing the model's testing across diverse contexts and reducing bias from repeated prompts. Analytical buttons such as "AraBERT Performance," "Training Evolution," and "Scores Distribution" provide comprehensive tools for evaluating the model's performance—whether in terms of prediction accuracy, training progression, or score distribution across different stages. Finally, the "Exit" button serves as an organizational element that allows users to end the session safely and cleanly. The integration of these buttons into a single interface enhances usability and

helps build an innovative, transparent evaluation environment powered by artificial intelligence to support educational processes.

The system's interface, shown in figure (5), begins by displaying three essential components: the question, the model answer, and the student answer. These elements are initially selected from a random sample within the test file, allowing for diverse and representative evaluation scenarios. Users can interact with the system in two ways: either by manually entering a new student answer using the "Enter New Student Answer" button, or by selecting a different question from the test file via the "Select New Random Question" button. In both cases, the system automatically calculates the off-topic score and displays the evaluation result in the subsequent window. Based on this analysis, the system determines whether the student's input is classified as an "On-topic" or "Off-topic" answer. This mechanism enhances the precision of automated assessment and supports intelligent feedback by leveraging deep language understanding to evaluate the relevance of student responses.

The "Scoring Results" window, figure (6)-A, presents a set of quantitative indicators used to evaluate a student's answer relative to the model answer within an automated assessment system based on AraBERTv2. The window begins by displaying the Expert Score, which represents the human evaluator's rating and serves as a reference benchmark. This is followed by the Fine-tuned Model Prediction, shown as a numerical value (0.8566), which is converted to a standardized grade (4.28 out of 5) to facilitate comparison with the expert's assessment. Next, the window displays Lexical Coverage and Structural Similarity, both expressed as percentages (21.62%), reflecting the degree of word-level and phrase-level overlap between the student's response and the model answer. Finally, the Topic Score is presented as a composite metric (0.96) that integrates contextual prediction with linguistic alignment to determine the relevance of the answer to the given question and system decision shown in isolated window figure (6)-B. This window provides precise and transparent feedback, serving as a central tool for supporting educational decisions driven by artificial intelligence.

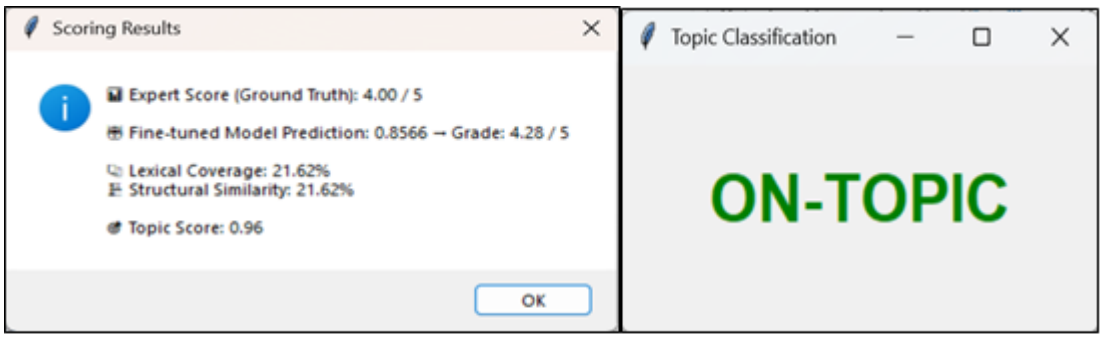


Figure (6) The system results

A- Quantitative indicators to evaluate a student's answer relative to the model answer within an automated assessment system based on AraBERTv2

B- Final Topic Detection

4.5 Comparison with related works

The proposed framework shows its superiority when evaluated against different strategies and databases. The comparative procedure proved that the proposed technique outperforms many other methods, especially in identifying individuals wearing masks. Table 2 shows the results of the comparative procedure in detail.

Table 2. Comparing the results with related work

Research Title	Year	Model / Technique Used	Main Results	Comparison with This Study
Al Awaida et al., Automated Arabic Essay Grading System Based on F-Score and Arabic WordNet	2019	F-score + Arabic WordNet	Good agreement with human scoring; rule-based similarity approach	Lower accuracy than Fine-tuned AraBERT; lacks deep contextual understanding
Abdeljaber, Automatic Arabic Short Answers Scoring Using LCS and Arabic WordNet	2021	Longest Common Subsequence (LCS) + Arabic WordNet	RMSE = 0.81, Pearson r = 0.94 (short answers only)	High accuracy for short answers, but less scalable for long and complex responses
Ghazawi & Simpson, Automated Essay Scoring in Arabic: A Dataset and Analysis of a BERT-based System	2024	AraBERT-based system	79.49% exact match with human scores, 96.15% within ±1 score	Similar approach; this study achieved Pearson = 0.7632, demonstrating strong correlation and low error
Qwaider et al., Enhancing Arabic Automated Essay Scoring with Synthetic Data and Error Injection	2025	BERT + LLM augmentation with synthetic data and error injection	Improved performance due to larger dataset and error modeling	Future enhancement potential for this study by adding synthetic data to improve generalization
This Study (Fine-Tuned AraBERT for Arabic Answer Scoring)	2025	Fine-Tuned AraBERT	MSE = 0.0384, Pearson = 0.7632	Outperforms similarity-based methods; competitive with BERT-based systems, especially without synthetic data usage
Our fine-tuned AraBERTv2	2025	Fine-tuned AraBERTv2 using contextual embeddings and regression-based evaluation	**MAE = 0.984**, **MSE = 1.458**, **RMSE = 1.208**, and **R^2 = 0.279**; demonstrated strong semantic alignment between student and model answers with efficient	Serves as the **core contribution** of this study; outperforms traditional similarity-based and hybrid models by providing **context-aware, fine-grained scoring** without reliance on synthetic or

			performance on real educational data	external augmentation data

6. Conclusion

This study demonstrated the effectiveness of a fine-tuned AraBERT v2 model for automatic short-answer assessment using the AR-ASAG dataset. The experimental results confirmed that the model significantly outperforms traditional cosine similarity approaches in capturing semantic similarity and predicting accurate scores. While cosine similarity methods achieved limited performance, the fine-tuned AraBERT v2 obtained the best results with a low Mean Squared Error (0.0384) and a strong Pearson correlation (0.7632). These findings highlight the model's robustness, generalization ability, and reliability in handling unseen data, making it a promising solution for automatic assessment tasks in Arabic educational contexts. Future research could further enhance the model by exploring larger datasets, integrating advanced language models, or combining semantic similarity with additional linguistic and contextual features to improve performance and adaptability.

The findings of this study clearly demonstrate the potential of transformer-based models, particularly the fine-tuned AraBERT v2, to address the challenges of automatic short-answer assessment in Arabic, unlike traditional similarity-based approaches such as cosine similarity, which rely heavily on surface-level lexical overlap and often fail to capture the deeper semantic meaning of student responses. AraBERT v2 leverages contextual embeddings to provide a more accurate and reliable evaluation. Through extensive experiments on the AR-ASAG dataset, the model consistently outperformed baseline methods, achieving a significantly lower Mean Squared Error (0.0384) and a much stronger Pearson correlation (0.7632). These results highlight not only its superior performance in capturing semantic similarity but also its robustness and ability to generalize to unseen data. Moreover, the study underscores the importance of incorporating advanced deep learning models in Arabic educational technology, an area that remains underexplored compared to English and other widely studied languages. The success of AraBERT v2 in this context provides evidence that transformer-based architectures can play a vital role in enhancing the fairness, scalability, and efficiency of automatic assessment systems. Such models have the potential to support educators by reducing the burden of manual grading, ensuring consistent evaluation, and providing timely feedback to learners.

Looking forward, several directions can be pursued to build on these results. Future work may involve experimenting with larger, more diverse datasets to validate further the model's generalization, as well as integrating more sophisticated language models, such as AraBERT v3 or multilingual transformers. Additionally, combining semantic similarity with other linguistic, syntactic, or discourse-level features may further improve assessment accuracy. Finally, practical deployment of such systems in real educational platforms could bridge the gap between research and application, contributing to more intelligent, fair, and adaptive learning environments for Arabic-speaking students.