Feature Engineering is not Dead: A Step Towards State of the Art for Arabic Automated Essay Scoring

Marwan Sayed, Sohaila Eltanbouly, May Bashendy, Tamer Elsayed
Computer Science and Engineering Department, Qatar University, Doha, Qatar
{me2104862, se1403101, ma1403845, telsayed}@qu.edu.qa

Abstract

Automated Essay Scoring (AES) has shown significant advancements in educational assessment. However, under-resourced languages like Arabic have received limited attention. To bridge this gap and enable robust Arabic AES, this paper introduces the *first publicly-available* comprehensive set of engineered features tailored for Arabic AES, covering surface-level, readability, lexical, syntactic, and semantic features. Experiments are conducted on a dataset of 620 Arabic essays, each annotated with both holistic and trait-specific scores. Our findings demonstrate that the proposed feature set is effective across different models and competitive with recent NLP advances, including LLMs, establishing the state-of-the-art performance and providing strong baselines for future Arabic AES research. Moreover, the resulting feature set offers a reusable and foundational resource, contributing towards the development of more effective Arabic AES systems.

1 Introduction

Automated Essay Scoring (AES) has emerged as a promising solution for efficient evaluation of written essays, offering scalable support for educational assessment. AES systems typically adopt either holistic scoring, which assigns a single overall writing quality score (Xie et al., 2022; Yang et al., 2020; Zhang et al., 2025), or trait-specific scoring, which evaluates distinct writing traits of the essay (Kumar et al., 2022; Ormerod, 2022). Recent AES research follows two paradigms: prompt-specific and cross-prompt. *Prompt-specific* AES involves training and testing models on essays written in response to the same prompt, often achieving high performance due to the model's specialization (Taghipour and Ng, 2016; Dong et al., 2017). In contrast, *cross-prompt* AES seeks to develop models that generalize across different prompts, enabling realistic and broader applicability but presenting greater challenges due to increased topical variability (Ridley et al., 2021). Despite progress in English AES, research on Arabic remains relatively underdeveloped, leaving a critical gap in the development of robust Arabic AES systems.

A key insight from English AES research is the critical role of engineered features in enhancing model performance. Several studies have demonstrated that combining linguistic features, particularly the set proposed by (Ridley et al., 2020), with different approaches, such as neural representations or language models, results in significant improvements in generalization and scoring performance (Ridley et al., 2021; Do et al., 2023; Li and Ng, 2024; Xu et al., 2025; Eltanbouly et al., 2025). Crucially, feature-based models have been shown to outperform embedding-based approaches, with hybrid approaches achieving the best results (Li and Ng, 2024; Lohmann et al., 2024). These findings highlight the value of feature engineering for English AES, motivating the need to bring a similar feature-driven perspective to Arabic.

In this work, we introduce the *first publicly-available* comprehensive list of engineered features for Arabic AES, covering surface-level, readability, lexical, syntactic, and semantic categories. Effectiveness of these features is evaluated across multiple *cross-prompt* models for holistic and trait scoring. Specifically, we benchmark their impact in standalone feature-based models and in hybrid architectures where features are integrated with language representations in encoder-based models.

Our contributions are: (1) introducing and releasing the **first** publicly-available feature set for Arabic AES[1], (2) evaluating the effectiveness of the features in cross-prompt setup across different modeling paradigms, (3) benchmarking the performance of the cross-prompt models against Large Language Models (LLMs), and (4) performing category-wise analysis of the feature importance.

[1] https://github.com/Maroibo/AES_features

Proceedings of The Third Arabic Natural Language Processing Conference, pages 231–245

The remainder of this paper is organized as follows: Section 2 outlines the related work. Section 3 discusses the categories of the extracted features. Section 4 details the different cross-prompt scoring models. Section 5 discusses our experimental setup, and Section 6 presents and analyzes the results. Finally, Section 7 concludes with suggested directions for future work.

2 Related Work

Despite advancements in English AES, Arabic research remains limited due to the scarcity of public datasets and the complexities of the language. Existing Arabic studies focus on prompt-specific setups and follow one of 3 approaches: feature-based, neural network-based, or language model-based.

Traditional approaches to Arabic AES have relied on rule-based methods and feature engineering (Alqahtani and Alsaif, 2020; Alsanie et al., 2022). In addition, several studies have utilized text similarity techniques to measure alignment between student essays and reference answers (Abdeljaber, 2021; Alobed et al., 2021a; Al Awaida et al., 2019; Alobed et al., 2021b; Azmi et al., 2019). These approaches have shown effectiveness, but, they often fail to capture deeper semantic understanding and remain unexplored in cross-prompt Arabic AES.

Other approaches leveraged neural networks and language models. Gaheen et al. (2020, 2021) utilized optimization algorithms to train a neural network. More recently, Ghazawi and Simpson (2024) fine-tuned AraBERT, achieving robust performance, while Machhout and Zribi (2024) introduced an improved AraBERT-based model with handcrafted features to evaluate essay relevance. The latest effort by Mahmoud et al. (2024) explored parameter-efficient fine-tuning strategies to further enhance AraBERT. Concurrently, Ghazawi and Simpson (2025) were pioneers in employing LLMs for Arabic AES, assessing models such as ChatGPT and LLaMA in various prompting setups.

The development of Arabic AES remains limited compared to English. Although some studies have explored feature-based methods, this area is not as well-established for Arabic. In contrast, engineered features have played a significant role in English AES, as demonstrated by their effectiveness across various state-of-the-art (SOTA) models (Do et al., 2023; Xu et al., 2025). Moreover, two recent studies (Li and Ng, 2024) and (Lohmann et al., 2024) have demonstrated that feature-based models outperform embedding-based models, reinforcing the importance of engineered features. Motivated by the superior performance of such features in English AES, this work aims to develop a comprehensive feature set tailored to Arabic and examine its effectiveness across different models. To the best of our knowledge, this is *also* the first study to investigate Arabic *cross-prompt* AES.

3 Feature Engineering

Motivated by the success of the engineered features in English AES in both feature-based models (Li and Ng, 2024) and hybrid approaches (Do et al., 2023; Xu et al., 2025), this study explores their potential in Arabic AES, with the goal of developing a comprehensive set of features tailored to Arabic.

We adopted features from three sources: a prior feature-based Arabic AES study (Alqahtani and Alsaif, 2020) as it provides a large set of features designed for Arabic AES, the widely used English AES features (Ridley et al., 2020), and the feature set proposed in a recent AES SOTA study (Li and Ng, 2024), bringing the total number of features to 816. To bring coherence to this diverse feature set, we organize the features into five main categories that capture writing characteristics at different levels. *Surface-level* features quantify basic structural essay properties. *Readability* measures estimate the complexity of the text. *Lexical* features analyze word choice and usage patterns. *Semantic* features assess similarity, relevance, and tone. Finally, *syntactic* features describe grammatical and structural organization. The categories are detailed next.

3.1 Surface-Level Features

Surface-level features focus on fundamental aspects of writing by quantifying measurable writing patterns that provide insights into writing quality at the character, word, sentence, and paragraph levels.

Character-level features: Orthographic precision is assessed through character-level features, including counts of misspellings and "همزة" usage, providing insight into the writer's attention to detail and writing accuracy.

Word-level features: Word-level characteristics are captured through various features, including measures of lexical diversity, such as the ratio of unique words, indicators of morphological complexity, such as average lemma length, and word count distribution across the essay's paragraphs.

Sentence-level features: Structural variation is

quantified by analyzing sentence length statistics (e.g., average, minimum, maximum, and variance), while capturing sentence counts across paragraphs. This subset of features sheds light on how sentence construction changes across essay segments.

Paragraph-level features: This subset of features assesses the essay structure at the paragraph level through measures such as paragraph counts and paragraph length statistics, including average, minimum, and maximum lengths.

3.2 Readability Metrics Features

These features estimate the essay's reading difficulty using established readability formulas.

Arabic-based metrics: Arabic readability measures range from simple metrics such as Heeti, considering only the average word length (Al-Heeti, 1984), to more comprehensive measures such as OSMAN (Open Source Metric for Measuring Arabic Narratives), which integrates multiple linguistic factors (El-Haj and Rayson, 2016).

English-adopted metrics: English readability measures, such as the SMOG-Index (Mc Laughlin, 1969) and Flesch–Kincaid (Kincaid et al., 1975), provide indications about the text's complexity and the comprehension level required to understand the content. Most of these measures rely on basic statistical properties of the text. For instance, Linsear Write formula (O'Hayre, 1966) estimates the reading level based on sentence and word lengths, and Flesch–Kincaid evaluates readability using sentence length and syllable counts. In this study, we apply these formula-based measures to Arabic text.

3.3 Lexical Features

This group focuses on analyzing word choice, phrase usage, punctuation, and recurring lexical patterns throughout the text.

N-gram features: This group of features is computed based on the top N unigrams identified in the dataset, including the counts of the most common words in the dataset, the number of sentences that contain these frequent words, and the proportion of sentences in which they occur.

Punctuation features: Punctuation usage is measured through quantitative counts and rule-based accuracy checks, including the presence of specific punctuation marks, individual punctuation mark counts, and assessments of correct usage, missing usage, and incorrect usage based on the rules defined by Alqahtani and Alsaif (2020).

Paragraph keyword features: This group detects phrases with religious or structural significance within designated essay sections. Notable examples include traditional openings like "بسم الله" and "الحمد لله" appearing in early paragraphs, as well as binary detection of introductory phrases in openings such as "في البداية" and "أولاً" as well as concluding terms in endings like "أخيراً".

Dialect features: Assessment of dialect usage evaluates the degree to which essays deviate from Modern Standard Arabic (MSA). This group includes the number of dialects in the essay quantified at the sentence level and their proportion relative to MSA sentences. These features are newly proposed, as Arabic AES is intended for MSA-based scoring, the consistent use of the standard language is a key indicator of writing proficiency.

3.4 Semantic Features

This category focuses on features related to the overall meaning and relevance of the essay content, as well as the relations between the essay's parts.

Prompt adherence features: Adherence to the prompt is quantified using embedding similarity scores. This includes computing the maximum, minimum, and average dot product between the embeddings of the essay sentences and the prompt, providing insight into how well the essay stays focused and relevant.

Sentiment features: Sentiment analysis captures the emotional tone and its spread across the essay. The features cover positivity, negativity, and neutrality at the sentence level, with the essay-level features representing the average sentiment scores across all sentences.

Text similarity features: These features assess the degree of similarity between different parts of the essay. They capture lexical overlap and semantic alignment through measures such as matched word counts and embedding similarity on the sentence and paragraph levels.

3.5 Syntactic Features

This category analyzes the grammatical structure and organization of sentences and phrases.

POS Tag features: These features capture the grammatical patterns through the frequency of part-of-speech tags throughout the essay.

POS bi-gram features: These features encode the count of POS bi-grams in the dataset, such as noun–verb and adjective–noun bi-grams.

Arabic grammatical features: This group targets grammatical constructs unique to Arabic, highlighting distinctive sentence structures and usage. These features include counts of auxiliary verbs, the presence of particles like "إِنَّ" and "كان", and occurrences of "الجزم" particles.

Pronoun features: This feature group caters to the use of pronouns and their distribution. Key features include individual pronoun counts, pronoun groupings such as demonstrative, interrogative, and relative pronouns, and the proportion of sentences that contain specific pronouns.

Discourse connectives features: The diversity of discourse connectives help in evaluating the essay's logical flow and cohesion. The group includes total conjunction counts, ratios of unique connectives, average spacing between connectives, and connective density relative to essay length.

Sentence structure features: These features characterize the complexity of sentence construction and syntactic depth, including features such as the average number of clauses per sentence, the maximum clause count, parse tree depths, and the frequency of nominal and verbal sentences.

4 Cross-prompt Scoring Models

The cross-prompt AES problem requires training a model on essays written in response to a set of *source* writing prompts, with the goal of scoring essays from a different unseen *target* prompt. During training, the model has access to the source prompts and their corresponding essays, along with scores for different essay traits. At inference time, only the target prompt and essays are available to the model. This setup challenges the model to generalize beyond the specific training prompts.

To evaluate the effectiveness of the proposed engineered features, we conduct a comparison across various cross-prompt models. These include purely feature-based and encoder-based models, also covering SOTA English models. For all models, we adopt a *multi-task* learning approach, where all the trait scores are predicted simultaneously.

Feature-based Models We select 3 traditional machine learning algorithms, namely Linear Regression (LR) (Galton, 1886), Random Forest (RF) (Breiman, 2001), and Extreme Gradient Boosting (XGB) (Chen and Guestrin, 2016). Moreover, following the SOTA model of English AES for holistic cross-prompt scoring (Li and Ng, 2024), we also select a simple feedforward Neural Network (NN).

Source	Prompt	Type	Essays	Len.
TAQEEM	1	Expl.	215	137
TAQEEM	2	Pers.	210	150
QAES	3	Pers.	115	500
QAES	4	Pers.	80	473

Table 1: *TAQAE* dataset statistics. "Expl." and "Pers." mean explanatory and persuasive, respectively. Length is indicated in average number of words.

Encoder-based Models Additionally, we select two Encoder-based models. The first is **ProTACT**, one of the current SOTA for trait scoring in English AES (Do et al., 2023). It constructs essay representations using CNNs and LSTMs over POS embeddings, while prompt representations combine POS and pre-trained GloVe embeddings (Mohammad et al., 2017). A multi-head attention mechanism obtains prompt-aware essay representations. These are concatenated with engineered features and fed into a linear layer for scoring. The same architecture has been adapted for Arabic, using AraVec[2] instead of GloVe.

Since pretrained language models have been widely adopted for AES in both English (Wang et al., 2022; Do et al., 2024) and Arabic (Ghazawi and Simpson, 2024; Mahmoud et al., 2024), we also fine-tune **AraBERT** (Antoun et al.), with a regression head for trait scoring, exploring two architectures. The first approach uses max pooling over token embeddings with trait-specific dense layers, while the second adds an attention layer to model dependencies between traits. More details are provided in Appendix A.

5 Experimental Setup

In this section, we outline the setup used to conduct our experiments, including the dataset, the implementation details, and the training setups.

Dataset The absence of standardized Arabic essay corpora has significantly slowed down progress in Arabic AES. In this study, we use a newly-formed dataset, denoted as ***TAQAE***, of 620 Arabic essays over 4 prompts drawn from two sources. The first source includes 425 essays for 2 prompts (corresponding to prompts 1 and 2) recently provided by TAQEEM 2025 shared task (Bashendy

[2] https://github.com/bakrianoo/aravec

et al., 2025) as the training set.[3] These essays were written by native Arabic first-year university students. The second source is the Qatari Corpus of Argumentative Writing (QCAW) (Ahmed et al., 2024),[4] which provides 195 essays for 2 prompts (corresponding to prompts 3 and 4), leveraging their publicly available QAES annotations (Bashendy et al., 2024).[5] Table 1 provides a breakdown of the prompts featured in our *TAQAE* dataset.

Essays from both sources have the same scoring procedures. Each essay is annotated across seven traits: Relevance (REL, 0–2), Organization (ORG: 0–5), Vocabulary (VOC, 0–5), Style (STY, 0–5), Development (DEV, 0–5), Mechanics (MEC, 0–5), and Grammar (GRM, 0–5), in addition to a Holistic score (HOL, 0–32) computed as the sum of all trait scores. Annotation follows institution-developed standardized rubrics from the Core Academic Skills Test (CAST) by Qatar University Testing Center (QUTC).[6]

Data Splits Due to the limited size of the dataset, we adopt a *leave-one-prompt-out* cross-validation setup in which each experiment holds out one prompt (out of the four available prompts) as the unseen target prompt, while the remaining three are used for training.

Evaluation To evaluate our models, we use Quadratic Weighted Kappa (QWK) (Cohen, 1968), a common measure for AES that assesses the agreement between the scores of two raters.

Feature Extraction We extract a total of 816 features using a combination of rule-based methods and Arabic NLP tools. The implementation details are provided in Appendix B, and we release the full list of features, including their categorization, descriptions, and implementation.

Feature Selection Given the large and diverse feature set, we employ a model-independent feature selection method in which a single selected set is shared across all traits, based on Pearson and Spearman coefficients. Correlations are computed between each feature value and the score of each trait. Features are then selected if their absolute correlation for either correlation metric with any trait exceeds a predefined threshold. This threshold is considered a hyperparameter and optimized during training, with candidate values in [0.1, 0.2, 0.3, 0.4, 0.5]. In cases where no features surpass the threshold, the top 10 most correlated features are selected.

Hyperparameter Tuning To tune the hyperparameters of each model, for each target prompt, we perform an inner 3-fold cross-validation, where for each fold, one of the three prompts is used as validation set, and the other two for training. The best configuration is selected based on the average QWK across the folds and is then used to evaluate the model on the unseen target prompt. To explore the hyper-parameter space, we used Bayesian hyperparameter optimization with the Tree-structured Parzen Estimator algorithm (Bergstra et al., 2011), using the TPESampler from the optuna library.[7] We set the number of trials to 20, with 5 startup trials. More details about model-specific hyperparameters are provided in Appendix C.

Training Setups We trained the selected models under various setups to evaluate the effectiveness of the engineered features across different scenarios. For the feature-based models, we consider two variants. In the first variant, models are trained using all the 816 features, denoted as LR, RF, XGB, and NN. In the second variant, feature selection is applied and the models are denoted as LR_{fs}, RF_{fs}, XGB_{fs}, and NN_{fs}, respectively.

For ProTACT and AraBERT, we consider two main training setups. In the first, models are trained without considering the features, relying only on the embedding of the essay and the prompt. We refer to these models as ProTACT^{-f} and AraBERT^{-f}. In the second variant, the features are concatenated with the embeddings, and feature selection is applied. We refer to these models as ProTACT_{fs} and AraBERT_{fs}. Also, we introduce a third variant of AraBERT that incorporates an attention layer, referred to as $\text{AraBERT}_{fs}^{+att}$.

Additionally, we evaluate the performance of three Arabic-centric LLMs under two different prompting scenarios. The motivation behind this comparison is to assess how common AES methods perform relative to recent LLM-based approaches. In the zero-shot (0) setting, the LLM is prompted to directly score the essay given the prompt text

[3] https://sites.google.com/view/taqeem-2025

[4] https://catalog.ldc.upenn.edu/LDC2022T04

[5] https://gitlab.com/bigirqu/qaes

[6] https://www.qu.edu.qa/sites/en_US/testing-center/TestDevelopment/cast

[7] https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html

and the essay. The few-shot (2-shot) setting provides the LLM with two example pairs of prompt texts and essays from prompts other than the target, as two examples strike a balance between offering sufficient scoring context and staying within the context length limit. In all scenarios, the LLM is required to provide scores for all traits. We selected the top three LLMs, at the time of the experiments, based on the Open Arabic LLM Leaderboard:[8] Fanar,[9] Command R7B Arabic,[10] and ALLaM.[11] The details of the LLM experiments are provided in Appendix D.

6 Experimental Results

In this section, we discuss the results of our experiments addressing 4 research questions *in the context of Arabic AES*: **RQ1**: How effective are engineered features? (6.1), **RQ2**: Do they provide significant contributions to more complex models? (6.2), **RQ3**: Which model achieves the best performance? (6.3), and **RQ4**: Which engineered features play the most significant role? (6.4).

6.1 Purely Feature-based Models (RQ1)

We first evaluate the effectiveness of the feature set using purely feature-based models. Table 2a shows the results of the models under two training settings: using all features and with feature selection.

Without feature selection, NN and XGB achieve the best and comparable performance, while LR performs significantly worse. After applying feature selection, LR_{fs} shows a substantial improvement, followed by RF_{fs}, indicating the effectiveness of feature selection. Conversely, NN_{fs} and XGB_{fs} exhibit minimal differences. Overall, RF_{fs} achieves the highest average performance across traits with a QWK of 0.294. However, each model excels on different traits: NN performs best on 3 traits, followed by RF_{fs} and XGB_{fs} with 2 traits each, and LR_{fs} with 1 trait.

Notably, across all models, feature selection resulted in varying impacts on individual traits. In some cases, there were significant performance drops, such as a decrease of approximately 6 points in the mechanics and grammar with NN_{fs}, and a 5-point drop in the style with XGB_{fs}. These results highlight that different traits have different characteristics, and certain features may not hold equal relevance or significance across all traits. A similar performance decline is observed across some prompts, as shown in Table 4a. This drop in QWK after feature selection could be attributed to the fact that feature selection is based on training data that is limited in both size and prompt diversity. Consequently, it may fail to capture prompt- and trait-specific variability.

Moreover, the number of selected features varies significantly across models, as shown in Table 3, ranging from 12 to 73 features on average. This is *considerably much lower-dimensional feature space* compared to the original 816 dimensions, while either enhancing average performance or having no discernible impact.

6.2 Effect of Incorporating Features (RQ2)

We examine the effect of incorporating the features into two encoder-based models: ProTACT, one of the SOTA models for English AES, and AraBERT, a widely adopted transformer-based model for Arabic AES. Both models are trained under two settings: with and without the addition of the feature vector. Table 2b presents the results of both configurations.

Overall, adding the features *significantly* improves the performance of almost all traits by an average of 20 and 10 points for ProTACT_{fs} and AraBERT_{fs}, respectively. Notably, ProTACT^{-f} performs substantially worse, highlighting that the contribution of engineered features outweighs the other components in the model architecture. Although AraBERT^{-f} outperforms ProTACT^{-f} in the absence of features, their performance becomes comparable once features are included. Furthermore, incorporating an attention layer in $\text{AraBERT}^{+att}_{fs}$ leads to improvements across all traits except the relevance, with an average increase of 3.4 points.

The number of features selected for the encoder-based models is considerably higher than that of the feature-based models, as shown in Table 3. This is expected, as the embedding dimensions are 100 for ProTACT and 768 for AraBERT, requiring a *large enough* feature dimensionality to contribute meaningfully to the model.

These results show the value of the engineered features, highlighting their predictive power and effectiveness in representing essay content and quality. These findings align with the work on English AES, where feature sets are commonly incorporated and have been shown to enhance model per-

[8] Open-Arabic-LLM-Leaderboard
[9] Fanar-1-9B-Instruct
[10] Command-R7b-Arabic
[11] ALLaM-7B-Instruct-preview

Model	REL	VOC	STY	DEV	MEC	GRM	ORG	HOL	Avg.
LR	-0.026	0.079	0.082	0.110	0.086	0.103	0.046	0.100	0.072
RF	0.056	0.350	0.281	0.255	0.243	0.240	0.312	0.412	0.269
XGB	0.064	0.356	0.315	0.267	0.281	0.241	0.335	0.392	0.282
NN	0.044	0.353	0.323$^\bullet$	0.241	**0.324**$^\bullet$	**0.317**$^\bullet$	0.299	0.348	0.281
LR$_{fs}$	0.070$^\bullet$	0.318	0.296	0.263	0.287	0.265	0.347	0.374	0.277
RF$_{fs}$	0.057	0.375	0.310	0.284$^\bullet$	0.269	0.262	0.376	**0.420**$^\bullet$	**0.294**$^\bullet$
XGB$_{fs}$	0.058	**0.383**$^\bullet$	0.269	0.281	0.294	0.249	**0.382**$^\bullet$	0.371	0.286
NN$_{fs}$	0.037	0.334	0.305	0.283	0.255	0.253	0.343	0.393	0.275

(a) Feature-based Models

Model	REL	VOC	STY	DEV	MEC	GRM	ORG	HOL	Avg.
ProTACT^{-f}	0.000	0.066	0.093	0.000	0.081	0.048	0.093	0.099	0.060
AraBERT^{-f}	0.096$^\bullet$	0.168	0.207	0.162	0.189	0.178	0.119	0.181	0.162
ProTACT$_{fs}$	0.082	0.309	0.300$^\bullet$	0.268$^\bullet$	0.276	0.269	0.286	0.324	0.264
AraBERT$_{fs}$	0.066	0.279	0.278	0.230	0.308	0.225	0.322	0.370	0.260
AraBERT$_{fs}^{+att}$	0.034	0.380$^\bullet$	0.291	0.262	0.322$^\bullet$	0.285$^\bullet$	0.375$^\bullet$	0.403$^\bullet$	**0.294**$^\bullet$

(b) Encoder-based Models

Model	REL	VOC	STY	DEV	MEC	GRM	ORG	HOL	Avg.
Fanar (0)	0.052	0.285$^\bullet$	**0.337**$^\bullet$	0.208	0.229	0.297$^\bullet$	0.345$^\bullet$	0.345	0.262
Fanar (2)	**0.149**$^\bullet$	0.278	0.313	**0.319**$^\bullet$	0.286$^\bullet$	0.291	0.259	0.348$^\bullet$	0.280$^\bullet$
R7B (0)	0.058	0.149	0.254	0.130	0.077	0.153	0.184	0.186	0.149
R7B (2)	0.136	0.279	0.296	0.274	0.227	0.278	0.289	0.337	0.265
ALLaM (0)	0.111	0.180	0.228	0.171	0.172	0.209	0.121	0.230	0.178
ALLaM (2)	0.075	0.127	0.099	0.124	0.115	0.141	0.098	0.148	0.116

(c) LLMs

Table 2: Comparison of the cross-prompt models, showing the average QWK performance per trait across all prompts. **Bold** values indicate the best performance per trait, and underlined values represent the second best. Values annotated with $^\bullet$ refer to the top model per trait within the model category.

formance (Ridley et al., 2020; Li and Ng, 2024).

Model	1	2	3	4	Avg.
LR$_{fs}$	10	10	8	22	12.5
RF$_{fs}$	10	80	58	86	58.5
XGB$_{fs}$	10	80	116	86	73
NN$_{fs}$	10	10	58	86	41
ProTACT$_{fs}$	165	10	575	22	193
AraBERT$_{fs}$	573	193	225	176	292
AraBERT$_{fs}^{+att}$	165	193	225	86	167

Table 3: *Tuned* number of selected features per model.

6.3 SOTA for Arabic AES (RQ3)

Table 2c presents the performance of LLMs, allowing a full comparison between all models of different categories reported in Table 2.

LLMs Among the evaluated LLMs, Fanar consistently outperforms the others, followed by Command R7B, while ALLaM demonstrates considerably lower performance. In general, the 2-shot setting yields notable improvements over zero-shot for both Fanar and Command R7B.

LLMs vs. Other Models For individual traits, LLMs, particularly Fanar, perform best on traits that require a broad understanding of essay content. This is most evident in *relevance*, which measures alignment with the prompt; *development*, which reflects the progression of ideas; and *style*, which captures structural cohesion. As for the remaining traits, the best LLM configuration still trails the strongest feature-based model by at least 2 points. The gap is most pronounced in vocabulary and holistic, where the top LLM performance lags by 9.8 and 7.2 points, respectively. Notably, the top two scores for relevance are achieved by LLMs. In contrast, simpler models outperform LLMs on traits that can be better captured through quantifiable features, e.g., mechanics and vocabulary.

Overall Comparison Overall, RF_{fs} and $AraBERT^{+att}_{fs}$ achieve the best average performance across all traits. However, there is no single model that excels at all traits, suggesting that more targeted trait-specific modeling or feature selection could offer further improvements. While LLMs demonstrate strengths in capturing higher-level aspects of content and structure, the best-performing LLM scenario still lags behind the simpler RF model by an average of 1.4 points. Finally, it is worth noting that the top three models, in terms of average performance, are either purely feature-based or incorporate engineered features into their architecture.

There are key differences between LLMs and traditional learning models in terms of their training. First, LLMs, pre-trained on vast data, benefit from a deeper comprehension and understanding of language. In contrast, the other models are either trained from scratch or utilize a smaller training set during the pre-training phase. Second, it is worth noting that, in our setup, LLMs are not fine-tuned for AES and rely solely on their pre-trained knowledge for scoring. Nevertheless, all traditional models have the advantage of being trained directly on AES. However, their performance is likely constrained by the relatively small training set in *TAQAE*, which consists of about 460 essays. We expect that their performance could improve significantly with access to a larger dataset.

Model	P1	P2	P3	P4	Avg.
LR	0.114	0.192	0.032	-0.048	0.072
RF	0.307	0.433	0.120	0.215	0.269
XGB	0.426	0.417	0.121	0.162	0.282
NN	0.386	0.448	0.061	**0.229**	0.281
LR_{fs}	0.377	0.404	0.115	0.213	0.277
RF_{fs}	0.347	**0.510**	0.135	0.186	**0.294**
XGB_{fs}	0.362	0.451	**0.143**	0.187	0.286
NN_{fs}	0.360	0.442	0.115	0.167	0.271

(a) Feature-based Models

Model	P1	P2	P3	P4	Avg.
$ProTACT^{-f}$	0.244	0.002	-0.003	-0.002	0.060
$AraBERT^{-f}$	0.467	0.191	-0.008	0.000	0.162
$ProTACT_{fs}$	0.369	0.414	0.079	0.196	0.264
$AraBERT_{fs}$	**0.493**	0.336	0.090	0.121	0.260
$AraBERT^{+att}_{fs}$	0.485	0.433	0.073	0.186	**0.294**

(b) Encoder-based Models

Model	P1	P2	P3	P4	Avg.
Fanar (0)	0.453	0.369	0.030	0.198	0.262
Fanar (2)	0.469	0.488	0.013	0.151	0.280
R7B (0)	0.133	0.296	0.047	0.120	0.149
R7B (2)	0.477	0.341	0.059	0.181	0.265
ALLaM (0)	0.302	0.320	0.025	0.064	0.178
ALLaM (2)	0.147	0.171	0.043	0.102	0.116

(c) LLMs

Table 4: Average QWK performance per prompt across all traits. **Bold** indicates best performance per prompt, and underlined values represent the second best.

Performance Per Prompt Table 4 illustrates the performance across various prompts, highlighting significant differences in prompt difficulty. For the feature-based models, the decline in QWK for some prompts after feature selection may be attributed to the distinct characteristics of the writing prompts, particularly P1, which is the only explanatory prompt in the dataset. Similarly, for the encoder-based models, P1 shows the least improvement when the features are added. This can be attributed to the fact that feature selection is conducted based on training data that is limited in both size and prompt diversity, which may not adequately capture this variability. As a result, features that are important for a specific type of prompt might be excluded if they are not relevant to other prompts in the training set, potentially harming performance. For the other prompts, P3 and P4 are generally more challenging to score with all the models, likely due to their higher essay length. In contrast, P2 appears to be the easiest to score, likely due to the strong representation of persuasive essays in the training set.

For the LLMs, Command-R7B shows consistent improvement across all prompts with the 2-shot setup, whereas ALLaM exhibits the opposite trend. Fanar, on the other hand, demonstrates an inconsistent pattern, where the 2-shot performs better on P1 and P2, while the zero-shot outperforms on both P3 and P4.

6.4 Feature Importance Analysis (RQ4)

We analyze the correlation between the extracted feature set and each target trait, focusing on three traits: holistic, relevance, and organization. These traits either illustrate patterns that are repeated across different traits or display unique properties. As shown in Figure 1, surface features consistently achieved the highest correlations overall, ranking as the top category for all traits except relevance. Character-based features were particularly prominent within this group, frequently appearing among

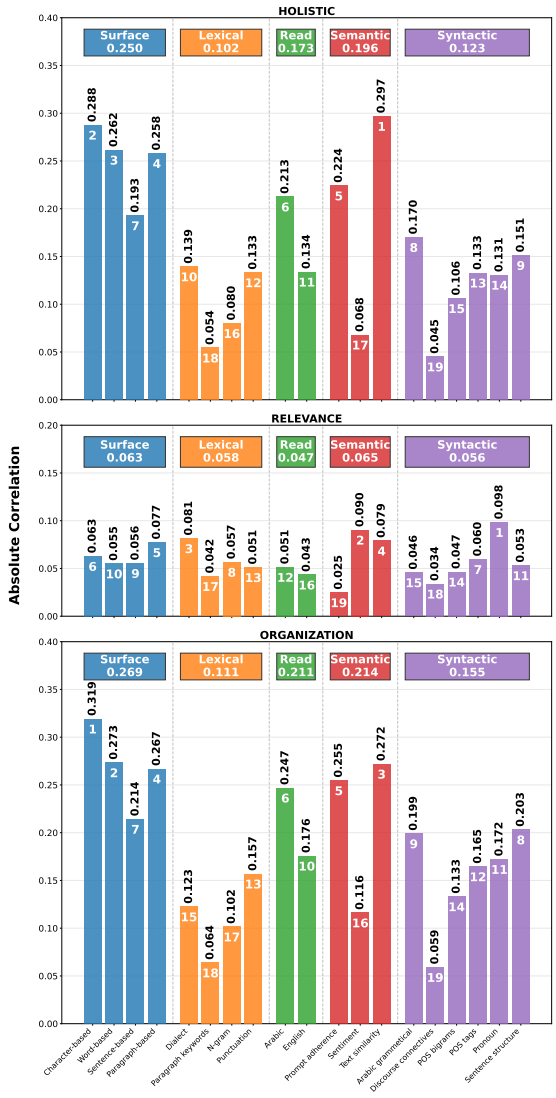

Figure 1: The maximum absolute correlations of features for the Holistic, Relevance, and Organization traits. Numbers inside the bars indicate the subcategory's rank.

the top two most correlated subcategories. Notably, all subcategories within the surface features were found to be highly predictive, with each one ranked in the top half across all the feature subcategories. Semantic features generally ranked second behind surface features. Within this category, the text similarity subcategory exhibited the highest correlations, appearing among the top four subcategories across all traits. On the other hand, the relevance trait exhibited a clear variation in this pattern, with semantic features emerging as the highest-ranking category and pronoun features identified as the most predictive subcategory.

The readability features ranked third across all other traits except relevance, with Arabic-based readability metrics consistently outperforming English-based ones. This aligns with expectations for an Arabic dataset.

Overall, the results indicate that combining surface features with semantic measures provides strong predictive signals across most traits. Traits were generally most correlated with simple, granular features, as reflected in the consistently lower correlations observed for most syntactic subcategories other than pronoun features. More analysis for the other traits is provided in Appendix E.

7 Conclusion and Future Work

In this study, we developed a comprehensive set of engineered features tailored for Arabic AES and systematically evaluated their effectiveness on a range of cross-prompt models, besides benchmarking their performance against SOTA Arabic-centric LLMs. Our findings indicate that features remain important and capture aspects of writing quality that remain underrepresented in encoder-based models and LLMs. Simple feature-based models are on par with, and in some cases outperform, more complex models, indicating that higher model capacity alone does not guarantee improved performance across all traits. Moreover, the varying importance of feature categories across traits suggests that Arabic AES could benefit from trait-specific models or specialized scoring modules for traits with similar characteristics.

In future work, we plan to explore the effectiveness of the proposed feature set in trait-specific models with alternative selection methods. While LLMs demonstrate strengths in capturing higher-level aspects of content and structure, fine-tuning and integrating engineered features offer promising directions to improve scoring performance.

Acknowledgment

The work of Sohaila Eltanbouly was supported by GSRA grant# GSRA12-L-0413-250111, and the work of the other authors was supported by NPRP grant# NPRP14S-0402-210127, both from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Limitations

Several limitations should be acknowledged in this work. First, the dataset used is relatively small with

limited diversity in prompt types, limiting the generalizability of the findings across different writing scenarios. The cross-prompt setting explored in this work is particularly sensitive to such limitations, as performance may vary with greater variability in prompt structure or student populations.

Second, we tried one model-independent feature selection method based on correlation thresholds. While it has shown effectiveness in the English SOTA model (Li and Ng, 2024), this approach might not be optimal in capturing the nuanced needs of individual traits. Different traits may benefit from tailored selection strategies or specialized modeling components.

Third, while we explored two prompting strategies for LLMs, we did not explore more advanced techniques such as the chain of thought or finetuning. These approaches may offer further performance gains worth investigating in future work.

Finally, we assumed that the scoring rubrics are not explicitly accessible to any model at inference time. Future work could explore methods that incorporate rubrics directly into the models.

References

Hikmat A Abdeljaber. 2021. Automatic arabic short answers scoring using longest common subsequence and arabic wordnet. *IEEE Access*, 9:76433–76445.

Abdelhamid M. Ahmed, Xiao Zhang, Lameya M. Rezk, and Wajdi Zaghouani. 2024. Building an Annotated L1 Arabic/L2 English Bilingual Writer Corpus: The Qatari Corpus of Argumentative Writing (QCAW). *Corpus-based Studies across Humanities*, 1(1):183–215.

Saeda A Al Awaida, Bassam Al-Shargabi, and Thamer Al-Rousan. 2019. Automated arabic essay grading system based on f-score and arabic worldnet. *Jordanian Journal of Computers and Information Technology*, 5(3).

Khalaf Al-Heeti. 1984. *Judgment Analysis Technique Applied to Readability Prediction of Arabic Reading Material*. Ph.D. thesis, ProQuest Dissertations and Theses. Copyright ProQuest LLC. ProQuest does not claim copyright in the individual underlying works. Last updated 2023-02-19.

Mohammad Alobed, Abdallah M M Altrad, and Zainab Binti Abu Bakar. 2021a. A comparative analysis of euclidean, jaccard and cosine similarity measure and arabic wordnet for automated arabic essay scoring. In *2021 Fifth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pages 70–74.

Mohammad Alobed, Abdallah MM Altrad, Zainab Binti Abu Bakar, and Norshuhani Zamin. 2021b. Automated arabic essay scoring based on hybrid stemming with wordnet. *Malaysian Journal of Computer Science*, pages 55–67.

Abeer Alqahtani and Amal Alsaif. 2020. Automated Arabic essay evaluation. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 181–190, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Waleed Alsanie, Mohamed I Alkanhal, Mohammed Alhamadi, and Abdulaziz O Alqabbany. 2022. Automatic scoring of arabic essays over three linguistic levels. *Progress in Artificial Intelligence*, pages 1–13.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Aqil M Azmi, Maram F Al-Jouie, and Muhammad Hussain. 2019. Aaee–automated evaluation of students' essays in arabic language. *Information Processing & Management*, 56(5):1736–1752.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Walid Massoud, Houda Bouamor, and Tamer Elsayed. 2025. TAQEEM 2025: Overview of the First Shared Task for Arabic Quality Evaluation of Essays in Multi-dimensions. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Eman Zahran, Hamdo Elhuseyin, Tamer Elsayed, Walid Massoud, and Houda Bouamor. 2024. QAES: First publicly-available trait-specific annotations for automated scoring of Arabic essays. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 337–351, Bangkok, Thailand. Association for Computational Linguistics.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32. Accessed: YYYY-MM-DD.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 785–794, New York, NY, USA. ACM.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Heejin Do, Yunsu Kim, and Gary Lee. 2024. Autoregressive score generation for multi-trait essay scoring. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1659–1666, St. Julian's, Malta. Association for Computational Linguistics.

Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023. Prompt- and trait relation-aware cross-prompt essay trait scoring. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1538–1551, Toronto, Canada. Association for Computational Linguistics.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

Mahmoud El-Haj and Paul Rayson. 2016. Osman—a novel arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255.

Sohaila Eltanbouly, Salam Albatarni, and Tamer Elsayed. 2025. TRATES: Trait-specific rubric-assisted cross-prompt essay scoring. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20528–20543, Vienna, Austria. Association for Computational Linguistics.

Marwa M Gaheen, Rania M ElEraky, and Ahmed A Ewees. 2020. Optimized neural network-based improved multiverse optimizer algorithm for automated arabic essay scoring. *INTERNATIONAL JOURNAL OF SCIENTIFIC TECHNOLOGY RESEARCH*, 9:238–243.

Marwa M Gaheen, Rania M ElEraky, and Ahmed A Ewees. 2021. Automated students arabic essay scoring using trained neural network by e-jaya optimization to support personalized system of instruction. *Education and Information Technologies*, 26:1165–1181.

Francis Galton. 1886. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263. Accessed: YYYY-MM-DD.

Rayed Ghazawi and Edwin Simpson. 2024. Automated essay scoring in arabic: a dataset and analysis of a bert-based system. *arXiv preprint arXiv:2407.11212*.

Rayed Ghazawi and Edwin Simpson. 2025. How well can llms grade essays in arabic? *arXiv preprint arXiv:2501.16516*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Rahul Kumar, Sandeep Mathias, Sriparna Saha, and Pushpak Bhattacharyya. 2022. Many hands make light work: Using essay traits to automatically score essays. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1485–1495, Seattle, United States. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024. Conundrums in cross-prompt automated essay scoring: Making sense of the state of the art. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7661–7681, Bangkok, Thailand. Association for Computational Linguistics.

Julian F Lohmann, Fynn Junge, Jens Möller, Johanna Fleckenstein, Ruth Trüb, Stefan Keller, Thorben Jansen, and Andrea Horbach. 2024. Neural networks or linguistic features?-comparing different machine-learning approaches for automated assessment of text quality traits among l1-and l2-learners' argumentative essays. *International Journal of Artificial Intelligence in Education*, pages 1–40.

Rim Aroua Machhout and Chiraz Ben Othmane Zribi. 2024. Enhanced bert approach to score arabic essay's relevance to the prompt. *Communications of the IBIMA*, 2024.

Somaia Mahmoud, Emad Nabil, and Marwan Torki. 2024. Automatic scoring of arabic essays: A parameter-efficient approach for grammatical assessment. *IEEE Access*.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Abu Bakr Mohammad, Kareem Eissa, and Samhaa El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

John O'Hayre. 1966. *Gobbledygook Has Gotta Go*. U.S. Department of the Interior, Bureau of Land Management, Denver, Colorado. A style manual that helped inspire the Plain Language movement.

Christopher Michael Ormerod. 2022. Mapping between hidden states and features to validate automated essay scoring using deberta models. *Psychological Test and Assessment Modeling*, 64(4):495–526.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. In *Proceedings of the AAAI*

conference on artificial intelligence, volume 35, pages 13745–13753.

Robert Ridley, Liang He, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2020. Prompt agnostic essay scorer: a domain generalization approach to cross-prompt automated essay scoring. *arXiv preprint arXiv:2008.01441*.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Yongjie Wang, Chuang Wang, Ruobing Li, and Hui Lin. 2022. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3416–3425, Seattle, United States. Association for Computational Linguistics.

Jiayi Xie, Kaiwei Cai, Li Kong, Junsheng Zhou, and Weiguang Qu. 2022. Automated essay scoring via pairwise contrastive regression. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2724–2733, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jiangsong Xu, Jian Liu, Mingwei Lin, Jiayin Lin, Shenbao Yu, Liang Zhao, and Jun Shen. 2025. Epcts: Enhanced prompt-aware cross-prompt essay trait scoring. *Neurocomputing*, 621:129283.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Mustafa Zeki, Othman O. Khalifa, and A. W. Naji. 2010. Development of an arabic text-to-speech system. In *International Conference on Computer and Communication Engineering (ICCCE'10)*, pages 1–5.

Chunyun Zhang, Jiqin Deng, Xiaolin Dong, Hongyan Zhao, Kailin Liu, and Chaoran Cui. 2025. Pairwise dual-level alignment for cross-prompt automated essay scoring. *Expert Systems with Applications*, 265:125924.

A AraBERT-based Model Architecture

This section describes two setups based on the AraBERT model. In the first setup, max pooling is applied over the output token embeddings to obtain an overall essay representation. This pooled representation is then passed separately for each trait through a trait-specific dense layer followed by a sigmoid output, producing eight predictions corresponding to the target traits. In the second setup, an attention layer is inserted between the dense layer and the sigmoid layer to operate on the trait representations, enabling the model to capture potential dependencies and interactions among them. This additional mechanism allows information sharing across traits. The architecture of the second variant is illustrated in Figure 2.

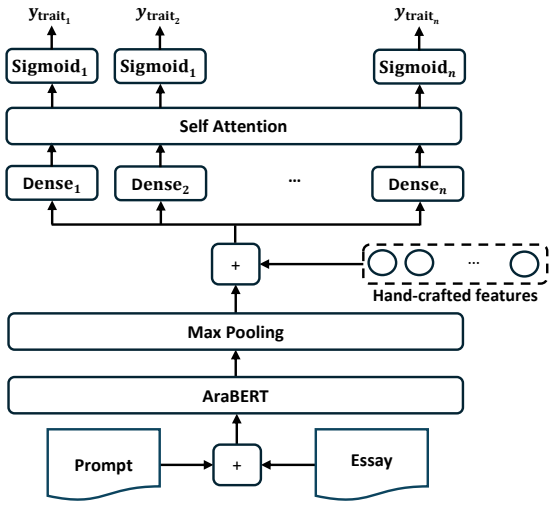

Figure 2: AraBERT$^{+att}_{fs}$ Architecture

B Feature Extraction

For feature extraction, we relied primarily on Camel Tools[12] as one of the main Arabic NLP processing frameworks (Obeid et al., 2020). Besides, we utilize other tools, including NLTK[13] for stopword removal, `pyspellchecker`[14] for spelling error detection, and CAMeL Parser[15] for the clause-based syntactic features.

For rule-based features, syllable counts followed the text-to-speech approach by Zeki et al. (2010), which are used by several readability measures. The other rule-based features are implemented based on the description provided by Alqahtani and Alsaif (2020) and Li and Ng (2024). For the features that require matching expressions from predefined lists, we applied fuzzy string matching implemented using SequenceMatcher function[16] with similarity thresholds of 0.93 or 0.95. These

[12] https://camel-tools.readthedocs.io/
[13] https://pythonspot.com/nltk-stop-words/
[14] https://pypi.org/project/pyspellchecker/
[15] https://github.com/CAMeL-Lab/camel_parser
[16] https://docs.python.org/3/library/difflib.html

threshold values are determined based on some preliminary experiments. This approach was used primarily for features related to paragraph keywords, e.g., detecting introductory phrases in the first paragraph or identifying concluding expressions in the final paragraph. For grammatical features, instead of fuzzy matching, we relied on morphological analysis to identify function words and particles.

For the semantic features, we used CAMeLBERT model.[17] To ensure consistency when calculating sentiment and prompt adherence features, the essay was segmented into batches of eight sentences to accommodate the model's limited context window. For dialect detection, we used the CAMeLBERT variant that is finetuned for dialect identification.[18] We consider only the number of dialects detected without any further categorization beyond distinguishing MSA and non-Standard Arabic, as more detailed classification was assumed to be irrelevant in the context of essay scoring.

C Hyperparameters Tuning

For all the considered fs models, we perform hyperparameter tuning for the feature selection threshold with candidate values in [0.1, 0.2, 0.3, 0.4, 0.5]. We also used a fixed random seed of 42 to ensure reproducibility. For the feature-based models, LR, RF, and XBG, we used the sklearn library[19] and the XGBoost library[20]. For NN-based models, all are trained for up to 50 epochs with early stopping based on the QWK score on the dev set, using a patience of 10, and a batch size of 16.

The hyperparameters used for each model are summarized in Table 5. The NN model is tuned over different hidden layer widths and learning rates, with a fixed dropout rate of 0.3. For AraBERT configurations, the learning rate values were different from those of other models, with the encoder and the dense layer tuned separately but using the same values. ProTACT settings included fixed embedding dimensions, maximum input limit for the essay and prompt, the number of attention heads, and convolutional parameters.

[17] https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-mix

[18] https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-mix-did-madar-corpus26

[19] https://scikit-learn.org/

[20] https://xgboost.readthedocs.io/en/stable/

سيتم إعطاؤك مقال كتب ردًا على الموضوع المعطى. مهمتك هي تقييم جميع المعايير التالية للمقال.

—

الموضوع: موضوع المقال
المقال: المقال المراد تقييمه.
الدرجات: الرجاء إعطاء الدرجات لجميع المعايير بهذا الشكل:
الصلة بالموضوع: 0-2 ، الهيكل العام: 0-5 ، المفردات: 0-5 ، الأسلوب والتماسك البنائي: 0-5 ، الأفكار والمضمون: 0-5 ، الإملاء والترقيم والشكل: 0-5 ، البناء والتراكيب: 0-5

—

الموضوع : هل تتفق أو تختلف جعلت الهواتف ورسائل البريد ...
المقال : إن مصطلح التكنولوجيا ...
الدرجات : الهيكل العام: 3.0 ، المفردات: 3.0 ، الأسلوب والتماسك البنائي: 3.0 ، الأفكار والمضمون: 3.0 ، الإملاء والترقيم والشكل: 3.0 ، البناء والتراكيب: 3.0 ، الصلة بالموضوع: 2

—

الموضوع : على الرغم من أهمية وسائل التواصل الاجتماعي ...
المقال : لا شك ان الافراط في استخدام وسائل التواصل ...
الدرجات : الهيكل العام: 1.0 ، المفردات: 2.0 ، الأسلوب والتماسك البنائي: 2.0 ، الأفكار والمضمون: 2.0 ، الإملاء والترقيم والشكل: 1.0 ، البناء والتراكيب: 1.0 ، الصلة بالموضوع: 1

—

الموضوع: باتَ اِهْتمام وحماس المراهقين لِتعلُّم رِياضةٍ جديدة ...
المقال: الصحة والجسم السليم من نعم الله على الإنسان ...
الدرجات:

Figure 3: An example of the LLM-prompt, containing the base instructions, the input format, the 2-shot examples, and the input essay for scoring. For zero-shot, the same prompt is used without the 2-shot examples.

D LLMs Experiments

Figure 3 presents the LLM-prompt template. In the zero-shot setup, the LLM receives the prompt text, the essay, and the score ranges for each trait. The model is instructed to generate scores for all traits following a predefined output format. For few-shot scoring, we adopt a 2-shot configuration, where two example essays, each with its corresponding prompt text and trait scores, are provided as demonstrations. These examples are randomly selected from two prompts that are different from the target. The LLM is then asked to score a new essay from the target prompt. To account for variability in example selection, the experiment is repeated five times using different random seeds: 1, 12, 22, 32, and 42, and we report the average of the 5 runs.

For all LLMs, we used the official checkpoints

Model	Hyperparameter Name	Value
RF	Max depth	[3-10] with a step of 1
	Max features	[0.1-0.9] with a step of 0.1
	Max samples	[0.1-0.9] with a step of 0.1
XGB	Max depth	[3-10] with a step of 1
	Learning rate	[0.01-5] with a step of 0.01
	Subsample	[0.1-0.9] with a step of 0.1
NN	Hidden layer widths	[64, 128, 256]
	Dropout rate	0.3
	Learning rate	[1e-5, 1e-4, 1e-3]
AraBERT	Input length	512 tokens
	Encoder learning rate	[1e-5, 5e-5, 1e-4]
	Dense-layers learning rate	[1e-5, 5e-5, 1e-4]
ProTACT	Learning rate	[1e-5, 1e-4, 1e-3]
	Embedding dimension	100
	Max essay length	500 tokens
	Max prompt length	100 tokens
	LSTM units	32
	Dense layer size	32
	Self-attention heads	4
	CNN filters	100
	CNN kernel size	3
	Dropout rate	0.5

Table 5: Model-specific hyperparameters

available on Hugging Face and conducted inference using the Hugging Face Transformers library.[21] To ensure reproducibility and minimize randomness of the LLMs output, we employed greedy decoding.

E Additional Feature Importance Analysis

Figure 4 shows the features correlation for the other five traits: mechanics, development, grammar, style, and vocabulary. Overall, similar patterns emerge, with surface-level features ranking as the top, and character-level and text similarity features being the two most predictive subcategories. The mechanics trait has higher correlations with readability metrics than any other trait. This aligns with the scoring criteria for mechanics, which emphasize factors related to readability, such as spelling and clarity. Development and grammar display consistently lower correlations across all syntactic subcategories except for Arabic grammatical features. Meanwhile, the lexical features consistently ranked lowest across all the traits.

[21] https://huggingface.co/docs/transformers

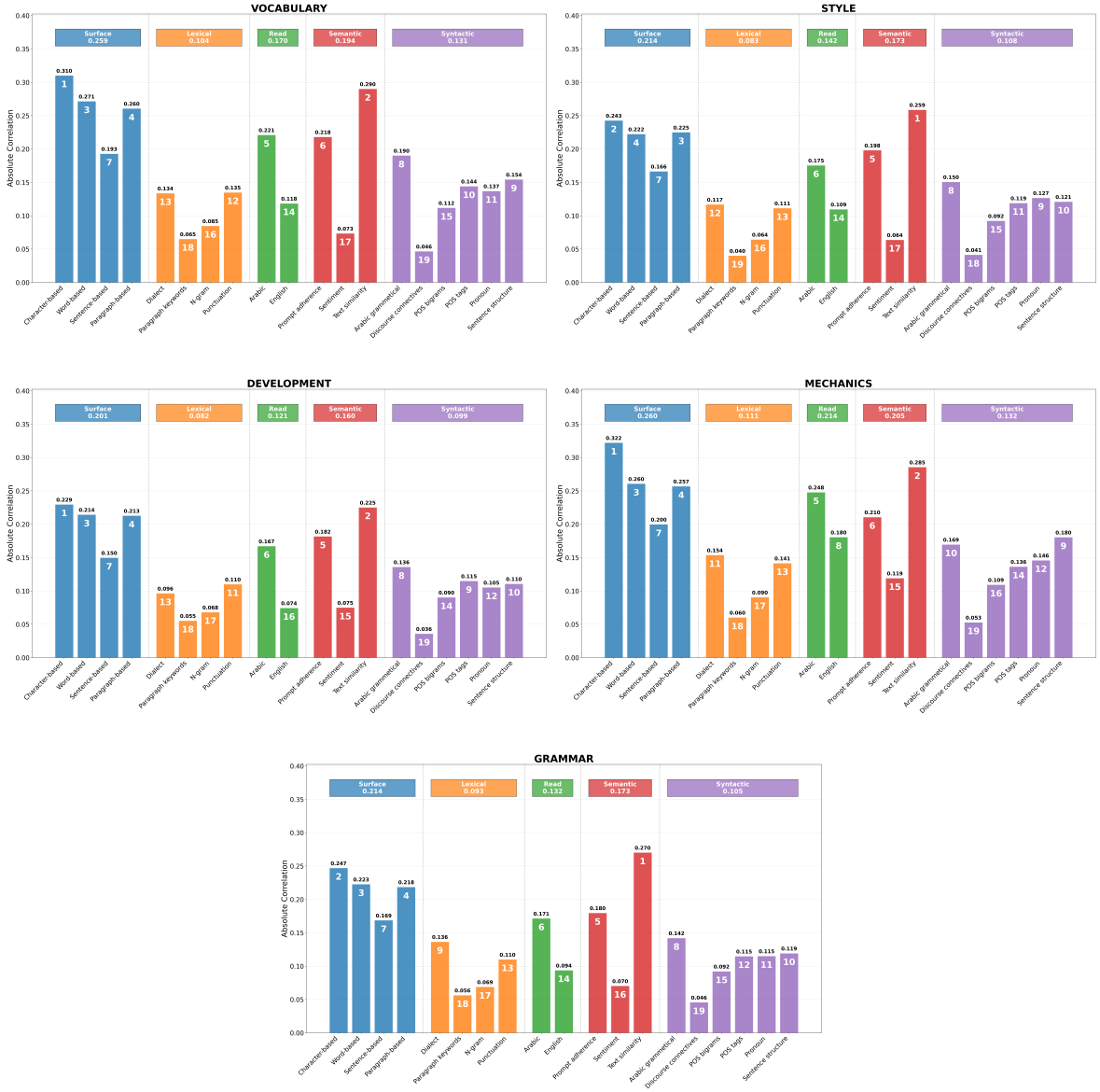

Figure 4: The maximum absolute correlations of features for the vocabulary, style, developments, mechanics, and grammar traits, with the numbers inside the bars indicating each subcategory's rank.