Phantoms at BAREC Shared Task 2025: Enhancing Arabic Readability Prediction with Hybrid BERT and Linguistic Features.

Ahmed Alhassan
Carnegie Mellon University Africa
aalhassa@andrew.cmu.edu

Asim Mohamed
African Institute for Mathematical Sciences
amohamed@aimsammi.org

Moayad Elamin
Carnegie Mellon University Africa
melamin@alumni.cmu.edu

Abstract

This paper describes our system for the BAREC 2025 Shared Task on Arabic Readability Assessment. Our approach is centered on a hybrid model that combines the deep contextual representations of a pre-trained transformer (AraBERTv02) with a rich set of engineered linguistic features. We extracted over 200 lexical, morphological, syntactic, and semantic features, which were refined to the 100 most informative ones through a multi-stage selection process. Our final model demonstrates significant effectiveness, achieving a **Quadratic Weighted Kappa (QWK) of 82.7%** and an exact accuracy of **57.6%** on the official blind test set. These results highlight the powerful synergy between transformer-based embeddings and explicit linguistic signals for the nuanced task of assessing Arabic text readability.

1 Introduction

Automatic Readability Assessment (ARA) aims to predict the difficulty level of a given text for a target audience. Although extensively studied for English, ARA for Arabic remains a developing field, presenting unique and significant challenges for modern Natural Language Processing (NLP) models (Liberato et al., 2024). The complexity of Arabic, which comes from its rich derivational morphology, optional diacritization, and the widespread phenomenon of diglossia, complicates the extraction of reliable readability features. Traditional readability formulas, often translated into English, do not capture these linguistic nuances. More recent machine learning and deep learning models have shown promise (Hazim et al., 2022), yet their performance is often constrained by the scarcity of large, high-quality, and fine-grained annotated corpora for Arabic.

The BAREC Shared Task 2025 on Arabic Readability Assessment (Elmadani et al., 2025a) directly addresses this gap by introducing a new, large-scale, and balanced corpus designed for this purpose (Elmadani et al., 2025b) . This initiative provides a crucial benchmark for the development and evaluation of sophisticated Arabic ARA systems. The task challenges participants to move beyond surface-level features and explore more complex linguistic and semantic representations to accurately predict readability scores.

In this paper, we present our system for the BAREC Shared Task. Our approach is novel in its hybrid architecture, which synergistically combines deep contextual embeddings from a pre-trained Arabic transformer model with a rich set of handcrafted linguistic features. These features are specifically designed to capture the morphological, syntactic, and psycholinguistic dimensions of Arabic text that influence reading comprehension. By integrating these diverse feature sets, our model aims to create a more holistic and accurate representation of text complexity. We hypothesize that this multi-faceted approach will outperform models that rely solely on either deep learning or traditional feature engineering, thereby setting a new standard for Arabic readability assessment.

2 Background

The BAREC Shared Task 2025 (Elmadani et al., 2025a) focuses on fine-grained, sentence-level readability assessment for Modern Standard Arabic. The primary goal is to predict a readability score for a given Arabic sentence on a continuous scale. The task is structured into three main tracks:

- **Open Track:** Participants are allowed to use any external data, resources, or pre-trained models to build their systems.

- **Constrained Track:** Participants are restricted to using only the provided training set of BAREC Corpus (Elmadani et al., 2025b) and specific, pre-approved external resources,

Proceedings of The Third Arabic Natural Language Processing Conference, pages 357–361
November 8-9, 2025 ©2025 Association for Computational Linguistics

namely the SAMER Corpus (Alhafni et al., 2025) and the SAMER Lexicon (Al Khalil et al., 2020).

- **strict Track:** Participants are restricted to using only the provided training set of BAREC Corpus.(Elmadani et al., 2025b)

We participated in the **strict track**.

The task utilizes BAREC (Balanced Arabic Readability Corpus) (Elmadani et al., 2025b), a comprehensive dataset containing sentences sourced from diverse genres and annotated according to detailed guidelines (Habash et al., 2025). Each sentence in the corpus is assigned a readability score derived from expert human annotations, which reflects the cognitive effort required for a reader to understand it. An example of an input sentence and its corresponding output score is shown below:

Input: بين طعن القَنا وخَفْق البُنودِ
(Translation: Between the thrust of spears and the fluttering of banners.)
Output: 17

Prior work in Arabic readability has evolved significantly. Early studies focused on adapting the classic readability formula, such as the Flesch-Kincaid index, which mainly uses shallow features such as word and sentence length. Later research incorporated more sophisticated and Arabic-specific linguistic features, including morphological complexity and syntactic structures, into machine learning frameworks like Support Vector Machines (SVM) and Random Forests (Cortes and Vapnik, 1995) (Breiman, 2001). With the advent of deep learning, researchers began to leverage neural networks and, more recently, large pre-trained language models like AraBERT (Antoun et al., 2020) and CAMeLBERT (Inoue et al., 2021). These models have demonstrated strong performance by learning rich semantic representations directly from text. Our work builds upon these advances by proposing a hybrid system that leverages the strengths of both feature-based and deep learning paradigms, a strategy we believe is crucial for capturing the multifaceted nature of text readability in Arabic.

3 System Overview

Our system is designed to address the multifaceted challenge of Arabic text readability by integrating deep contextual understanding with explicit linguistic knowledge. The core of our approach is a hybrid neural architecture that leverages a pre-trained transformer model alongside a curated set of engineered features.

Design Rationale: The primary challenge in readability assessment is to capture a wide range of signals, from syntactic complexity and lexical choice to semantic coherence. While pre-trained models like BERT excel at learning contextual representations, they may not explicitly capture specific linguistic phenomena known to influence readability. Our design decision to fuse BERT with handcrafted features is motivated by this; we provide the model with both implicit, learned representations and explicit, targeted linguistic cues, creating a more robust and informed system.

Algorithmic Framework: Our model, implemented in PyTorch and the Hugging Face `transformers` library, consists of two main components: a text encoding module and a feature fusion classifier.

1. **Textual Representation:** We use the `aubmindlab/bert-base-arabertv02` model to generate contextualized embeddings for the input text. For a given sentence, the final hidden state of the special `[CLS]` token is used as its aggregate semantic representation. Let this be denoted as $\mathbf{e}_{\text{text}} \in \mathcal{R}^{768}$.

2. **Linguistic Feature Representation:** The 100 features selected from our feature engineering pipeline are compiled into a numerical vector, $\mathbf{f}_{\text{raw}}$. This vector is standardized using a `StandardScaler` (fit on the training data) to ensure zero mean and unit variance, resulting in the final feature vector $\mathbf{f}_{\text{num}}$.

3. **Hybrid Feature Fusion:** The textual and linguistic representations are combined through concatenation to form a unified feature vector:

$$\mathbf{c} = [\mathbf{e}_{\text{text}} \oplus \mathbf{f}_{\text{num}}]$$

   where $\oplus$ denotes the concatenation operation. This vector $\mathbf{c} \in \mathcal{R}^{768+100}$ serves as input to the final classification layer.

4. **Classification Head:** The combined vector $\mathbf{c}$ is passed through a multi-layer perceptron (MLP) to predict the readability level. This layer is trained to classify the input into one of the 19 ordinal readability classes.

Training Configuration: The model is trained for a maximum of 10 epochs using the AdamW optimizer with a learning rate of 5×10^{-5} and a linear learning rate scheduler. To counteract class imbalance, we employ a weighted Cross-Entropy Loss, with weights inversely proportional to class frequencies. We utilize mixed-precision training for efficiency. The model's performance is monitored on the validation set using the Quadratic Weighted Kappa (QWK) score, and we apply early stopping with patience of 2 epochs to prevent overfitting.

4 Experimental Setup

Dataset: We utilized the BAREC sentence-level dataset for our experiments. The data is partitioned into three distinct sets: a training set for model development, a validation set for hyperparameter tuning, and a test set for final evaluation. The respective sizes and characteristics of these splits are determined by the original dataset providers. In addition to the standard splits, we also process the sentence-level blind test set.

Preprocessing and Feature Engineering: To prepare the data for our models, we implement a comprehensive pre-processing and feature engineering pipeline.

Text Normalization: Each sentence undergoes a series of normalization steps using the `camel-tools` library(Obeid et al., 2020). This includes Unicode normalization, normalization of Alef (أ, إ, آ to ا), Alef Maksura (ى to ي), and Teh Marbuta (ة to ه), followed by the elimination of all diacritics.

Feature Extraction: We extract a rich set of more than 200 features from the normalized text, leveraging the capabilities of `camel-tools`. These features can be categorized as follows:

- **Surface Features:** Basic statistics such as word count, average and standard deviation of word length, and the ratio of long ($>=7$ characters) and short ($<=3$ characters) words.
- **Character-level Features:** Ratios of non-Arabic characters, punctuation, numbers, mathematical operators, and other symbols within each sentence.
- **Morphological Features:** Proportions of various parts of speech (POS), gender, number, aspect, case, and other morphological characteristics derived from the top analysis of an MLE disambiguator. We also compute morphological richness, verb-to-noun ratio, and affix ratios (prefix, suffix) based on morphological tokenization.
- **Semantic Features:** We include the count and ratio of Named Entities (NER), a sentiment score (positive, neutral, negative) and dialect identification scores, particularly the confidence score for Modern Standard Arabic (MSA).
- **Lexical Features:** The ratio of stop words in a sentence and the stem diversity, calculated as the ratio of unique stems to the total number of stems.

Feature Selection: To reduce dimensionality and mitigate multicollinearity, we apply a three-stage feature selection process to the training data:

1. **Variance Thresholding:** Features with variance below a threshold of 0.01 are removed.
2. **Correlation Filtering:** Highly correlated features are filtered out. We compute the Pearson correlation matrix and remove one feature from any pair with a correlation coefficient greater than 0.95.
3. **Tree-based Selection:** A Random Forest classifier is trained on the remaining features to rank their importance. The top 100 most informative features are selected for the final feature set.

Implementation Details

Our primary model is a hybrid architecture that combines a pre-trained transformer with the engineered numerical features. The model is built using PyTorch and the Hugging Face `transformers` library.

Model Architecture: We use the `aubmindlab/bert-base-arabertv02` model as our text encoder. The output representation of the `[CLS]` token is extracted and concatenated with the vector of scaled numerical features. This combined vector is then passed through a classification head consisting of a linear layer, a SiLU activation function, a dropout layer ($p = 0.2$), and a final linear layer to produce the output logits for the 19 readability classes. A dropout layer ($p = 0.3$) is also applied to the combined feature vector before it enters the classifier.

Training: The model is trained for a maximum of 10 epochs with a batch size of 16. We use the AdamW optimizer with a learning rate of 5×10^{-5} and a linear learning rate scheduler. To address class imbalance, we employ a weighted Cross-Entropy Loss function, where weights are inversely proportional to class frequencies in the training set. We utilize mixed-precision training to accelerate computation. Early stopping is implemented with a patience of 2 epochs, monitored by the validation Quadratic Weighted Kappa (QWK) score. The best-performing model based on validation QWK is saved for evaluation.

Evaluation Metrics

Given the ordinal nature of the readability labels, we evaluated model performance using a suite of metrics. In addition to standard classification and regression metrics like **Exact Accuracy** and **Mean Absolute Error (MAE)**. We also report **Adjacent Accuracy** (allowing for an off-by-one error), **the 3, 5, and 7 Levels Accuracy**—classifying the sentences as if they are classified into 3, 5, and 7 different classes, respectively—and the **Quadratic Weighted Kappa (QWK)**, which is particularly well-suited for measuring inter-rater agreement on an ordinal scale.

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

$$\text{QWK} = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}}$$

where w_{ij} are the weights, O_{ij} is the observed count, and E_{ij} is the expected count for a label pair (i, j).

5 Results

Our system's performance was evaluated on the official BAREC blind test set. We also conducted internal experiments to compare different configurations of our model's classification head on the development set. The internal comparison results are included in Appendix A. The evaluation focuses on metrics suited for ordinal classification, primarily **Quadratic Weighted Kappa (QWK)**, alongside **Exact Accuracy**, **Adjacent Accuracy (Acc ±1)**, and **Mean Absolute Error (MAE)**.

5.1 Official Blind Test Set Results

On the official competition blind test set, our final model achieved a strong performance, demonstrating its robustness and generalization capabilities. The system attained a **QWK of 82.7%**, confirming a high level of agreement with the gold-standard labels. The exact accuracy was **57.6%**, while the adjacent accuracy (Acc ±1) reached **72.3%**, indicating that most of our model's errors were minor, differing by only a single readability level. The complete results are presented in Table 1.

QWK	Acc	Acc ±1	MAE
82.7%	**57.6%**	**72.3%**	**1.06**

Acc (3)	Acc (5)	Acc (7)
77.2%	71.3%	67.4%

Table 1: Final results of our system on the official sentence-level blind test set.

The high QWK and adjacent accuracy scores validate our hybrid approach, confirming that combining pre-trained language models with carefully engineered linguistic features is highly effective for sentence-level readability assessment in Arabic.

6 Conclusion

In this paper, we present our system for the BAREC 2025 Shared Task on sentence-level Arabic Readability Assessment. Our approach successfully integrated a powerful pre-trained Arabic transformer model with a comprehensive set of linguistic features to create a robust prediction system. The final model achieved an impressive **Quadratic Weighted Kappa of 82.7%** on the blind test set, demonstrating the efficacy of our methodology.

Our key finding is that, while transformers are excellent at capturing semantic context, their performance is significantly enhanced by explicit features that describe lexical complexity, morphological richness, and sentence structure. This hybrid strategy proved crucial for navigating the subtleties of the Arabic language. Future work could involve exploring more advanced transformer architectures, incorporating features from diverse linguistic resources, and conducting a thorough error analysis to better understand the remaining challenges in automatic readability assessment.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2025. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Nizar Habash, Hanada Taha-Thomure, Khalid N. Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

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.

A Internal Model Comparison

To select the best architecture, we compared three variants of our BERT-based model on the development set: one using a SiLU activation function, one using the Swish function, and one employing an ordinal regression head. The results, summarized in Tables 2 3, show that the models with **SiLU** and **Swish** activation functions performed very similarly and slightly better than the ordinal regression approach across most metrics. Based on its marginally higher QWK score, the BERT (SiLU) configuration was selected for the final submission.

Model	Accuracy	Accuracy ±1	MAE
BERT (swish)	56.18%	70.66%	1.0917
BERT (SiLU)	55.87%	69.90%	1.1023
BERT (ordinal)	53.61%	69.78%	1.1473

Table 2: Model Performance Metrics (Part 1)

Model	QWK	Acc (7)	Acc (5)	Acc (3)
BERT (swish)	81.15%	65.45%	69.01%	74.60%
BERT (SiLU)	81.17%	64.41%	67.95%	74.55%
BERT (ordinal)	79.35%	63.38%	67.06%	72.87%

Table 3: Model Performance Metrics (Part 2)