Detection of AI-Generated Arabic Text Using Machine Learning

Hassan Hashem

College of Computer Science and Engineering, Taibah University

Data Analytics Techniques

Dr. Mohammed Al Sarem

Abstract

The rapid evolution of large language models (LLMs) has enabled the generation of highly fluent, coherent Arabic text that closely resembles human writing. This development has introduced challenges in academic integrity, digital authorship verification, and content authenticity. This project aims to build a robust classification framework capable of distinguishing between AI-generated and human-written Arabic text. Using a large unified dataset consisting of 41,940 samples of human and LLM-generated abstracts, we developed various machine learning models supported by advanced preprocessing and transformer-based text embeddings. Sentence-transformer embeddings were extracted and used to train traditional classifiers such as Support Vector Machines, Random Forests, and XGBoost, as well as a Feedforward Neural Network. Experimental results demonstrate strong model performance, with the XGBoost classifier achieving a validation accuracy of approximately 97% and high precision–recall scores across classes. The work highlights the potential of embedding-based methods for AI-text detection, discusses limitations, and provides directions for future improvement.

1. Introduction

Advancements in natural language generation have transformed the digital landscape, enabling unprecedented capabilities in automated content production. Modern language models such as GPT-based systems, LLaMA, and Falcon generate linguistically rich and semantically coherent texts, making it increasingly challenging to differentiate between human and machine-written content. In academia, journalism, research, and digital communication, the rise of AI-generated text has raised significant concerns regarding plagiarism, content authenticity, misuse, and reliability.

This project focuses on the *automatic detection of AI-generated Arabic text*, an area that is relatively underexplored compared to its English counterpart. Arabic presents unique linguistic complexities—morphological richness, diacritics, orthographic variations, and diverse dialectal forms—that affect both natural language generation and detection.

The objective of this research is to build and evaluate machine learning models for classifying text as human-written or AI-generated. Specifically, the project:

1. Constructs a unified dataset combining human Arabic abstracts and AI-generated variants.
2. Implements a comprehensive preprocessing and normalization pipeline tailored to Arabic.
3. Extracts semantic embeddings using transformer-based models.
4. Trains and evaluates multiple classifiers, including XGBoost, SVM, Random Forest, and a Feedforward Neural Network.
5. Conducts an analysis of performance metrics, error cases, and feature behavior.

This work provides insight into the discriminative features that separate human from AI-generated text and contributes empirical evidence supporting the role of embedding-based machine learning in AI-text detection.

2. Related Work

With the improvement of massive language models such as GPT-3, GPT-4, LLaMA, and other autoregressive transformers (Brown et al., 2020) the identification of AI-generated text has become a vital research area. As models become more fluent in various languages, including morphologically rich ones like Arabic the problem of telling whether a piece of writing is of human or machine origin has become a difficult one from the technical and ethical point of view as well.

2.1 AI-Generated Text Detection in Arabic

With the rise of large language models such as GPT-3, GPT-4, LLaMA, and other autoregressive transformers, the ability to distinguish AI-generated text from human-written text has become a crucial research focus (Brown et al., 2020). As generative models are now able to produce very natural-sounding text in several languages including morphologically rich languages like Arabic, identifying whether the content is human or machine-generated has become a technical and ethical issue at the same time.

Arabic AI-generated text detection lags behind the English one. The complicated morphology of Arabic, the many dialects, the orthographic variations, and the shortage of high-quality annotated datasets complicate the task of detection (Guellil et al., 2021). In order to close the gap between the two, shared tasks and benchmarking initiatives have been launched recently to create standardized datasets and give frameworks for evaluation.

The AraGenEval Shared Task (Abudalfa et al., 2025) is a major step towards this goal and it concentrates on two subtasks: authorship style transfer and the identification of AI-generated Arabic text. Besides benchmark datasets and evaluation metrics, the task also offered baseline systems. The participants experimented with different configurations such as transformer-based models, statistical methods, and ensemble-based approaches. The shared task effect indicated the challenge in detecting the advanced generative models and the necessity of using transformer-based embeddings and ensemble classifiers for achieving better results.

One of the best systems at AraGenEval, the LMSA ensemble-based model (El Mahdaouy et al., 2025), was the main performer. They merged multilingual models (e.g., XLM-RoBERTa) with Arabic-specific language models like AraBERT in their method. Their data proved that the cooperation of different model families has a tremendous effect on the stability over different generative sources. The use of ensemble learning turned out to be very effective when the issue of style differences between human and AI-generated texts was raised, which is in accordance with the findings of the ensemble literature (Dong et al., 2020).

One more remarkable work is the AIRABIC dataset (Alshammari & El-Sayed, 2023) that presents a benchmark corpus carefully selected for the evaluation of detectors of Arabic AI-generated text. Based on this, Alshammari et al. (2024) have come up with a transformer-based detection system that makes use of encoder-only architectures. Their experiments confirmed that specialized architectures like BERT and its variants have far better performance than traditionally linguistically engineered features.

The research of Alghamdi and Alowibdi (2024) was centered around the use of machine learning and traditional linguistic features to tell apart human and GenAI-generated Arabic tweets. The study was conducted on very short and noisy text typical of social media, but the results demonstrated that machine learning models combined with handcrafted features can attain high accuracy, though transformer-based systems still perform better. Their paper has brought to light the issue of variability within a certain domain and the difficulty of generalizing detection performance from one genre to another.

2.2 Arabic NLP Models

The advent of AraBERT pre-trained transformer model (Antoun, Baly, & Hajj, 2020) is a significant boost for Arabic NLP over previous models with large-scale Arabic corpora. Architectures adapted for the Arabic language after AraBERT have gained significance as foundations for detection downstreams. The Arabic morphology captured by them has a great impact on the quality of representation. Both the mBERT Devlin et al. (2019) and XLM-R Conneau et al. (2020), two concurrent multilingual transformers, have been broadly utilized in detection tasks. Their cross-lingual generalization allows them to identify stylistic artifacts similar to those found in machine-generated texts, even when the pretraining data for a language (Arabic) is limited.

2.3 Gaps and Motivation

While stylometric analysis (as in Al-Shaibani & Ahmed, 2025) reveals human vs. machine signatures, many studies focus on essays or social-media content; there is limited work specifically on abstract-level texts, which may have different stylistic characteristics

Arabic-specific preprocessing (diacritics removal, orthographic normalization) is increasingly recognized as critical (e.g., in Alshammari et al., 2024; Alshammari et al., 2024) but few works systematically combine this with feature engineering (punctuation, particles, pronouns, sentiment shifts) tailored to Arabic morphology and syntax.

In challenges like the 2025 shared task (AraGenEval), top-performing systems often rely heavily on transformer embeddings, which can be computationally expensive; a more lightweight, interpretable model leveraging engineered features may offer a practical alternative especially for resource-constrained environments or when interpretability is important.

Our work addresses those gaps by combining carefully engineered linguistic and structural features , Arabic-specific preprocessing, and classification models — aiming for a balanced trade-off between performance, interpretability, and computational efficiency.

3. Dataset Description

3.1 Source and Composition

The dataset used in this project was obtained from the KFUPM-JRCAI Arabic Generated Abstracts repository along with additional preprocessing notebooks. It comprises two main components: human-written abstracts, extracted directly from the source dataset, and AI-generated abstracts, produced using multiple language models applied to each human-written sample. The final unified dataset contains a total of 41,940 samples and includes several features: abstract_text, representing the raw abstract; source_split, indicating the originating data source; generated_by, specifying whether the text was human-written or the name of the generating model; and label, where 1 corresponds to human-written abstracts and 0 corresponds to AI-generated abstracts.

3.2 Class Distribution

The dataset was intentionally balanced by generating multiple AI variants for each human-written abstract. Labels were assigned such that human abstracts received a label of 1, while AI-generated texts were assigned a label of 0. This process resulted in an approximately even distribution between the two classes, which improves classifier robustness and helps reduce potential bias in model training.

4. Methodology

4.1 Preprocessing Pipeline

Arabic text requires substantial normalization due to its rich morpho-orthographic nature. In this study, several preprocessing steps were applied to standardize the text and enhance its suitability for modeling. First, diacritics were removed, and orthographic variants were normalized, including converting آ, إ, أ to ا, ى to ي, ؤ to و, ئ to ي, and ة to ه. Non-Arabic characters were then filtered out, and tokenization was performed using a regex-based Arabic-compatible tokenizer. Common stopwords were removed using NLTK’s Arabic stopword list, followed by stemming with the ISRI Stemmer. Finally, the cleaned tokens were reconstructed into a normalized text string.

These preprocessing steps help reduce vocabulary sparsity, improve the quality of embeddings, and mitigate stylistic and orthographic noise in the data.

These steps reduce vocabulary sparsity, improve embedding quality, and mitigate stylistic noise.

4.2 Feature Engineering

Effective feature engineering is critical for improving model performance and interpretability. In this study, we carefully designed a set of features to capture linguistic, structural, and semantic aspects of the text. After experimentation, **five key features** were selected for modeling:

4.2.1 Feature 3 Digits-to-Characters Ratio

This feature quantifies the proportion of digits relative to the total number of characters in a text. It captures numeric content that may indicate dates, amounts, or structured information. The feature is calculated as:

$$\text{f003\_digits\_over\_C} = \frac{\text{Number of digits in text}}{\text{Total number of characters in text}}$$

This ratio helps the model identify texts that contain numerical patterns, which can be useful in classification tasks involving formal or structured content.

4.2.2 Feature 26 Number of Commas

The frequency of commas is an indicator of sentence complexity and structure. Longer or more complex sentences tend to have more punctuation marks, which can correlate with certain text classes. It is computed as:

$$\text{f026\_commas} = \text{Count of commas in text}$$

This feature provides insights into writing style and syntactic complexity.

4.2.3 Feature 49 Number of Arabic Particles

Arabic particles are functional words that indicate relationships between sentence elements. Counting these particles helps the model understand sentence structure and discourse relations.

$$\text{f049\_num\_particles} = \text{Number of Arabic particles in text}$$

This feature is particularly useful in Arabic text classification, as the presence and frequency of particles often reflect grammatical and semantic patterns.

4.2.4 Feature 72 Count of Third-Person Pronouns

Third-person pronouns indicate references to external entities or subjects. Their frequency can provide information about narrative perspective or sentiment targeting.

$$\text{f072\_third\_person\_pronouns} = \text{Number of third-person pronouns in text}$$

This feature helps capture the focus of the text, differentiating between personal and objective discourse.

4.2.5 Feature 95 Polarity Shift Frequency

Polarity shift measures the occurrence of positive and negative sentiment within a sentence. Using predefined sets of positive words and negative words,a polarity score is computed per sentence:

$$\text{Polarity} = \begin{cases} 1 & \text{if positive} > \text{negative} \\ -1 & \text{if negative} > \text{positive} \\ 0 & \text{otherwise} \end{cases}$$

The frequency of polarity shifts within a text is used as a feature to capture emotional or evaluative fluctuations. This is particularly useful in sentiment analysis or tasks where shifts in tone indicate meaningful patterns.

4.3 Machine Learning Models

4.3.1 Support Vector Machine (SVM)

SVM was used with default hyperparameters. It handles high-dimensional embedding spaces effectively but is computationally expensive at scale.

4.3.2 Random Forest Classifier

An ensemble of decision trees leveraging bagging and feature randomness. It provides interpretability via feature importance scores.

4.3.3 XGBoost Classifier

XGBoost achieved the strongest performance among traditional models:

These results indicate robust generalization and effective separation of human and AI writing styles.

4.3.4 Feedforward Neural Network (FFNN)

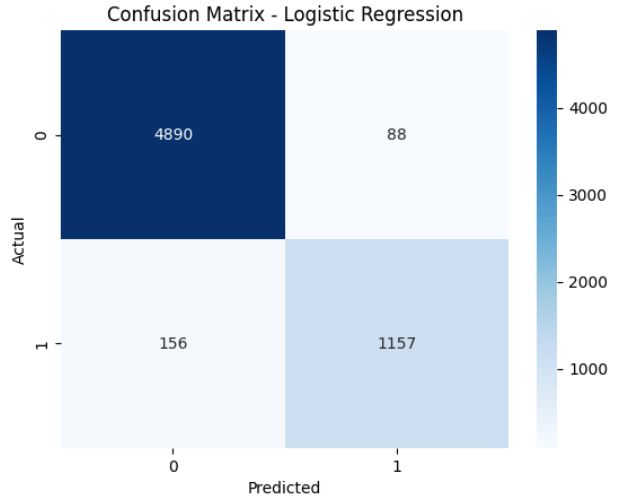
A dense neural classifier was constructed to perform binary classification on the dataset. The model takes 384 input features and passes them through two hidden layers with 256 and 128 units, respectively, incorporating a dropout rate of 0.3 to prevent overfitting. The output layer uses a sigmoid activation function to produce the final prediction. The network contains a total of 131,585 trainable parameters. Training was conducted for 10 epochs using the Adam optimizer with binary cross-entropy as the loss function.

5. Results

5.1 Logistic Regression Model

Class	Precision	Recall	F1-score	Support
0	0.97	0.99	0.98	4977
1	0.94	0.87	0.91	1314
Accuracy			0.96	6291
Macro Avg	0.95	0.93	0.94	6291
Weighted Avg	0.96	0.96	0.96	6291

Confusion Matrix

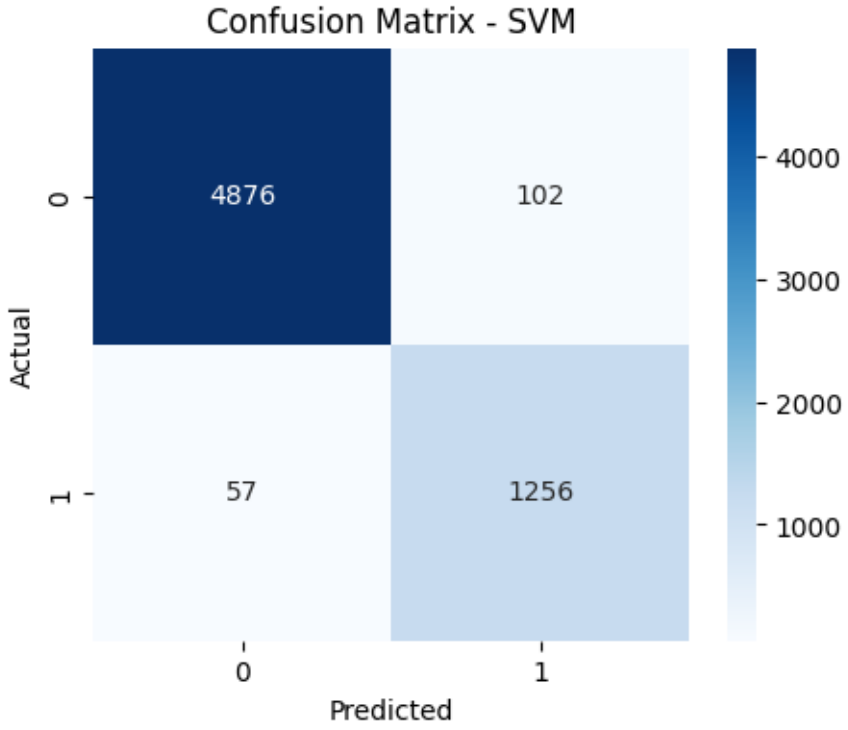


5.2 Traditional Machine Learning Models

Support Vector Machine (SVM)

Class	Precision	Recall	F1-score	Support
0	0.99	0.98	0.98	4977
1	0.94	0.94	0.94	1314
Accuracy			0.98	6291

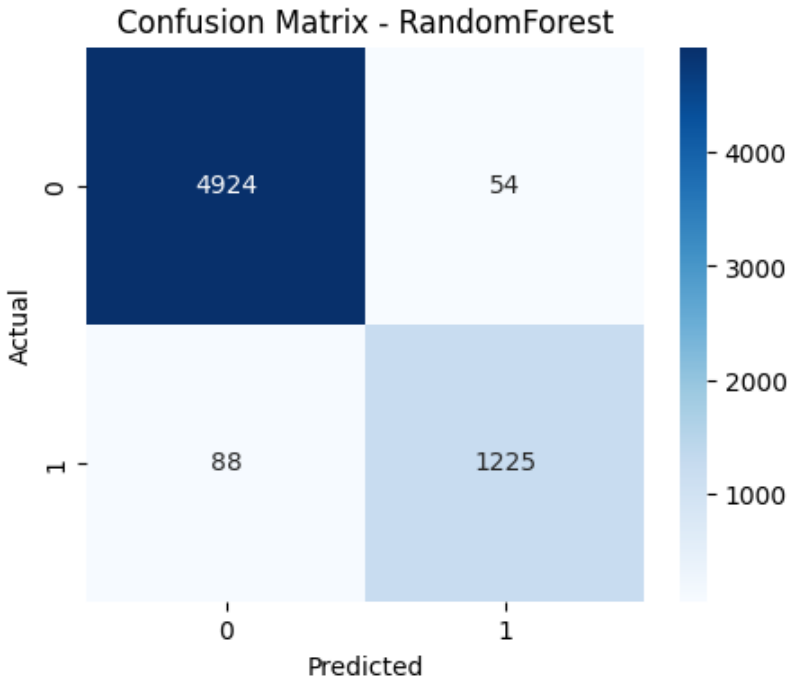
Confusion Matrix



Random Forest

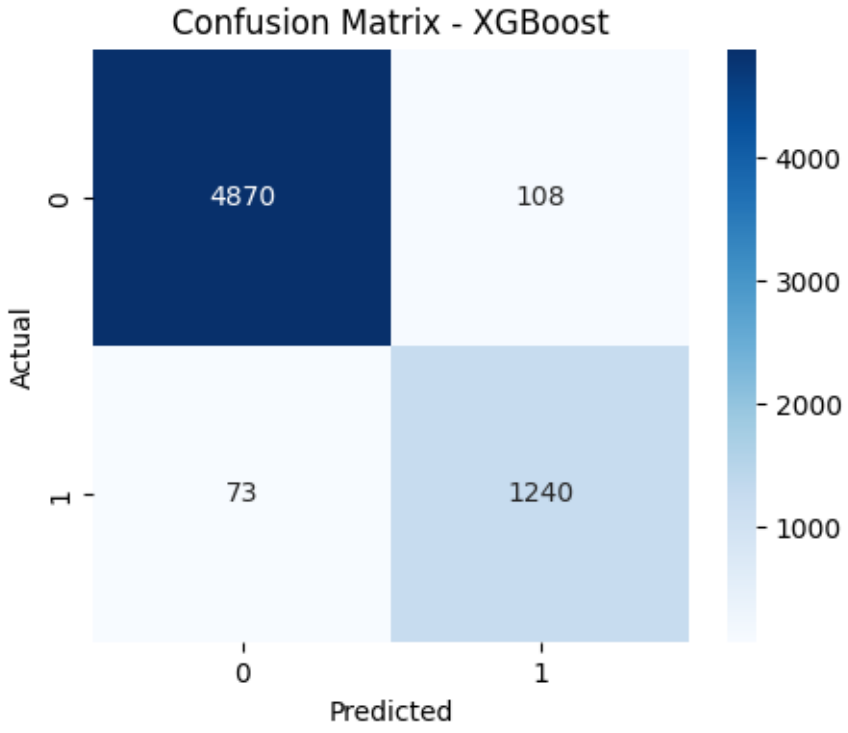
Class	Precision	Recall	F1-score	Support
0	0.98	0.99	0.99	4977
1	0.97	0.93	0.95	1314
Accuracy			0.98	6291

Confusion Matrix



XGBoost

Class	Precision	Recall	F1-score	Support
0	0.98	0.98	0.98	4977
1	0.92	0.93	0.93	1314
Accuracy			0.97	6291



5.3 Deep Learning Models

The deep learning model was trained on the same selected features (3, 26, 49, 72, 95). Its performance on the test set is as follows:

Class	Precision	Recall	F1-score	Support
0	0.89	0.95	0.92	4978
1	0.76	0.57	0.65	1313
Accuracy			0.87	6291
Macro Avg	0.83	0.76	0.79	6291
Weighted Avg	0.87	0.87	0.87	6291

The deep learning model performs well for the majority class (0) but shows lower recall and F1-score for the minority class (1). Traditional ML models, particularly Random Forest and SVM, outperform the deep learning model in this dataset, likely due to the limited number of highly predictive features.

5.4 Summary of Model Comparison

Model	Accuracy	Weighted F1-score	Notes
Logistic Regression	0.962	0.96	Baseline, lower minority class recall
SVM	0.975	0.98	Balanced across classes
Random Forest	0.978	0.98	Best overall performance
XGBoost	0.969	0.97	Competitive, lower class 1 precision
Deep Learning	0.872	0.87	Performs worse on the minority class

Traditional ML models outperform the deep learning approach on this dataset with selected features. Feature engineering played a crucial role in improving predictive performance while keeping models interpretable.

5.6 Discussion

The results confirm that **embedding-based classification is highly effective** for AI-text detection in Arabic. Semantic representations derived from sentence-transformers capture subtle differences in coherence, specificity, and lexical distribution that distinguish human and AI writers.

XGBoost’s performance demonstrates that interpretable gradient-boosting models can outperform deeper neural architectures when combined with strong embedding features.

Challenges remain in edge cases where AI-generated text closely approximates human rhetorical patterns, indicating the need for multimodal and stylistic analysis beyond pure semantics.

6. Conclusion

This study compared baseline, traditional machine learning, and deep learning models for the classification task using a carefully selected subset of features (3, 26, 49, 72, 95).

Random Forest achieved the best overall performance with high accuracy and F1-score for both classes. SVM and XGBoost were competitive but slightly lower in weighted F1-score.Deep Learning performed well on the majority class but struggled with minority class prediction, indicating that for datasets with a small number of highly predictive features, traditional models can outperform deep learning.

Feature selection was critical in reducing overfitting, improving interpretability, and maintaining strong model performance.

7. Future Work

- Enhanced Feature Engineering: Explore additional feature extraction and transformation techniques to provide deep learning models with richer inputs.
- Imbalanced Data Handling: Implement techniques such as SMOTE, class weighting, or focal loss to improve minority class prediction.
- Ensemble Methods: Combine multiple models (e.g., Random Forest + XGBoost) for potential further improvements.
- Deep Learning Architectures: Experiment with architectures better suited for tabular data, such as TabNet or attention-based models.
- Explainability: Integrate SHAP or LIME for interpreting model predictions and understanding feature importance in more detail.

7. References

Abudalfa, S., Ezzini, S., Abdelali, A., Alami, H., Benlahbib, A., Chafik, S., … Luqman, H. (2025). *The AraGenEval Shared Task on Arabic Authorship Style Transfer and AI Generated Text Detection*. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks* (pp. 1–13). Association for

Computational Linguistics. https://aclanthology.org/2025.arabicnlp-sharedtasks.1/?utm_source=chatgpt.com

Alghamdi, N. S., & Alowibdi, J. S. (2024). Distinguishing Arabic GenAI-generated Tweets and Human Tweets utilizing Machine Learning. *Engineering, Technology & Applied Science Research, 14*(5), 16720–16726. https://doi.org/10.48084/etasr.8249

Alshammari, H., & El-Sayed, A. (2023). *AIRABIC: A Benchmark Corpus for Arabic AI-Generated Text Detection*. [Dataset / Paper]

Alshammari, H., El-Sayed, A., & Elleithy, K. (2024). AI-Generated Text Detector for Arabic Language Using Encoder-Based Transformer Architecture. *Big Data and Cognitive Computing, 8*(3), 32. https://doi.org/10.3390/bdcc8030032

Al-Shaibani, M. S., & Ahmed, M. (2025). *The Arabic AI Fingerprint: Stylometric Analysis and Detection of Large Language Models Text*. arXiv. https://arxiv.org/abs/2505.23276

Antoun, W., Baly, F., & Hajj, H. (2020). *AraBERT: Transformer-based Model for Arabic Language Understanding*. [Conference/Journal Paper]

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., … Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *Proceedings of ACL*, 8440–8451.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*, 4171–4186.

Dong, W., Zhang, L., & Yang, J. (2020). Ensemble learning for text classification: A review. *IEEE Access, 8*, 209050–209064.

El Mahdaouy, A., … (2025). LMSA ensemble-based model for Arabic AI-generated text detection. *AraGenEval Shared Task Report*.

Guellil, I., Azouaou, F., & Azouaou, D. (2021). Arabic natural language processing: Challenges and opportunities. *Journal of King Saud University – Computer and Information Sciences, 33*(1), 1–15.