Human and AI Arabic Text Detection

Abstract

The development of generative artificial intelligence has increased the difficulty of distinguishing between human-written and machine-generated Arabic text. This study aims to develop an effective detection framework capable of differentiating between both types of writing using a combination of classical machine-learning models as Logistic Regression, Support Vector Machines (SVM), and XGBoost, also Feedforward Neural Network (FFNN)used. The results highlight the potential of combining traditional algorithms with neural approaches to improve detection accuracy, valuable insights, and AI safety applications. The Feedforward Neural Network (FFNN) algorithm achieved the highest accuracy.

1.Introduction

The emergence of large language models (LLMs) and generative AI systems has transformed how Arabic digital content is produced. With the increasing fluency and coherence of AI-generated text, distinguishing machine-created content from human writing has become a pressing challenge for researchers, educators, and digital governance institutions. This issue is important in the Arabic context due to the richness and morphological complexity, as well as the relative scarcity of high-quality detection datasets.

Recent tasks and research initiatives have emphasized the growing need for reliable AI-text detection technologies in Arabic. The AraGenEval Shared Task introduced by (Abudalfa et al,2025) marked a significant step in standardizing evaluation methods for Arabic authorship style transfer and AI-generated text identification. Studies such as (Alghamdi and Alowibdi,2024) further demonstrated that machine-learning techniques can successfully distinguish between AI-generated and human tweets by analyzing lexical and stylistic cues. (Alshammari and El-Sayed,2023) contributed AIRABIC, a dedicated benchmark dataset designed to examine the performance of AI detectors across various Arabic text types.

Despite these advances, there remains a clear need for simple, efficient, and accessible detection models that do not rely exclusively on large transformer-based architectures. This work addresses that need by investigating classical machine-learning modelsLogistic Regression, SVM, XGBoost and Feedforward Neural Network (FFNN), evaluating their ability to detect AI-generated Arabic text using linguistic and statistical features. This approach aims to balance interpretability and practical accuracy.

2.Related Work

Research on detecting AI-generated Arabic text has grown with the increasing of large language models. Prior studies have explored datasets, detection methods and computational models that support the identification of machine-generated content. This section summarizes the most relevant contributions in four interconnected domains: benchmark datasets, detection approaches, ensemble methods, and Arabic deep learning research.

2.1 Datasets for Arabic AI-Generated Text

One of the foundational drivers of progress in this field is the availability of high-quality annotated datasets. The AIRABIC dataset, proposed by (Alshammari,2023), introduced a collection of human- and AI-generated Arabic texts designed specifically for evaluating AI detectors. This dataset filled an important gap by providing diverse text samples representing various Arabic dialects and writing styles.

The AraGenEval Shared Task (Abudalfa et al., 2025) significantly expanded dataset availability by offering a standardized benchmark for Arabic authorship style transfer and AI-generated text detection. Its data covers multiple genres and sources, enabling researchers to evaluate new detection methods under comparable conditions.

These datasets provide the backbone for current detection systems and directly influence the design and evaluation of the models in this study.

2.2 Classical and Machine Learning Approaches

Several studies have examined the performance of classical machine-learning algorithms for distinguishing AI-generated from human-written Arabic text. (Alghamdi,2024) demonstrated that simple models such as Logistic Regression and SVM can effectively classify AI-generated tweets by exploiting lexical patterns and stylometric features. Their results highlight that lightweight models remain valuable for short-form and social media text.

Parallel research in broader NLP contexts shows similar observations. Traditional algorithms maintain competitive performance when combined with strong feature extraction methods such as TF-IDF or embeddings, making them suitable for real-time or large-scale systems where computational cost matters.

2.3 Ensemble Learning

Ensemble methods have become important for improving the robustness and accuracy of AI-generated text detection. (Dong et al,2020) provided a broad survey illustrating how ensemble learning through model averaging and boosting can enhance classification results compared to individual models.

The LMSA system presented in the AraGenEval Shared Task (2025) applied an ensemble of multilingual and Arabic-specific models to detect AI-generated Arabic text with high accuracy. Their experimental findings reinforce the effectiveness of combining complementary models, especially for complex and morphologically rich languages like Arabic.

2.4 Transformer-Based Models and Arabic Deep Learning Research

Transformer-based architecture has revolutionized NLP detection tasks. BERT (Devlin et al., 2019) and its variants remain widely used due to their deep contextualized representations, which excel at capturing subtle semantic and syntactic nuances. Their influence extends widely in Arabic NLP, supported by surveys such as (Wahdan et al.,2020), which see that deep learning models, transformer-based ones, enhance Arabic text classification across various domains.

Recently, (Wu et al.,2025) surveyed LLM-generated text detection methods, emphasizing the importance of combining statistical, neural, and linguistic features. Their insights underscore the growing need for hybrid detection pipelines that integrate multiple modeling perspectives.

3. Dataset Description

The KFUPM-JRCAI Arabic Generated Abstracts repository provided the dataset and other preprocessing notebooks utilized in this study. It consists of two primary parts: AI-generated abstracts, which are created by applying many language models to each human-written sample, and human-written abstracts, which are taken straight from the source dataset. A total of 41,940 samples make up the final unified dataset, which has several features: abstract_text, which represents the raw abstract; source_split, which indicates the original data source; generated_by, which indicates whether the text was written by a human or the name of the generating model; and label, where 1 denotes abstracts written by humans and 0 denotes abstracts created by artificial intelligence.

4. Methodology

This study aims to develop a comprehensive detection framework that distinguishes human-written Arabic texts from AI-generated ones. To achieve this, we evaluate four types of models: Logistic Regression, Support Vector Machines (SVM), XGBoost, and a simple Feedforward Neural Network (FFNN) built on top of BERT embeddings. The overall pipeline includes preprocessing, feature extraction, model training, and validation.

4.1 Data Preprocessing and Feature Extraction

Preparing the dataset for model training involved several stages, including text cleaning, tokenization, normalization, and an extensive set of handcrafted features designed to capture lexical, structural, and semantic differences between human-written and AI-generated Arabic text. Since Arabic presents unique linguistic challenges as rich morphology, orthographic variation, and diacritics preprocessing and feature engineering were essential steps to enhance the discriminative power of the machine-learning models.

Arabic text was normalized to reduce spelling variation and unify character forms. This included:

- Removing diacritics (tashkeel)
- Normalizing hamza forms
- Normalizing ligatures (e.g., "لا" → "لا")
- Removing repeated punctuation and unnecessary whitespace
- Standardizing Arabic and English alphanumeric tokens

Each document was then:

- Tokenized into words
- Split into sentences
- Converted into clean text**,** used for embedding-based features

These steps created consistent input before applying feature extraction.

4.2 Feature Engineering

4.2.1 Feature 2 Ratio of Letters to Total Characters

This feature measures the proportion of alphabetic characters (Arabic and English) relative to all characters:

letters / total_characters

This ratio provides insight into:

- The density of alphabetic content
- The presence of symbols, punctuation, or numeric patterns often found in structured AI outputs
- Writing fluency and noise levels

AI-generated text often has smoother character distributions, while human-written content may include irregularities or mixed-use symbols.

4.2.2 Feature 21 Brunet’s W Measure

Brunet’s W is a lexical diversity metric defined as:

$$W = N^{V^{-\alpha}}$$

where:

- N = number of word tokens
- V = number of unique word types
- α = 0.172

This measure captures the richness of vocabulary in the text. Human authors typically exhibit higher lexical variability, while some AI-generated texts, especially short or templated responses, may show lower diversity.

This feature was computed from the cleaned token list.

4.2.3 Feature 40 Sentence Length Frequency Distribution

It generates a distribution of sentence lengths within each document. This distribution reveals:

- Stylistic preferences (short vs. long sentences)
- Rhythm and pacing
- Homogeneity of structure

AI-generated texts sometimes have repetitive sentence-length patterns, while human texts show greater variation.

4.2.4 Feature 59 Count of the Top-500 Embedding Vocabulary

To measure alignment between text vocabulary and the language model’s internal learned distribution, we computed:

- The top 500 vocabulary tokens from the tokenizer.
- The count of words in each document that appear within this top-500 set

This feature reflects the extent to which the text uses common lexical items favored by LLMs. AI-generated text often overuses high-frequency words in the model's embedding vocabulary, while human authors may use more domain-specific or uncommon terminology.

4.2.5 Feature 78 – Perplexity Using Arabic GPT-2 (AraGPT2-base)

Perplexity is a powerful signal for detecting AI-generated text. It measures how "expected" the content is to a language model:

$$\text{Perplexity} = e^{\text{loss}}$$

Using aubmindlab/aragpt2-base, we computed perplexity for each document:

- Low perplexity : text is predictable by GPT-2 (often AI-generated)
- High perplexity : content is less predictable and more human-like

This feature is computationally expensive but highly informative because it directly evaluates text coherence under an Arabic language model.

4.3 TF-IDF and Embeddings

4.3.1 TF-IDF Vectorization

TF-IDF captured token-level lexical patterns, which served as input for Logistic Regression, SVM, and XGBoost. This representation is particularly suitable for high-dimensional sparse models.

4.3.2 BERT Embeddings

TF-IDF captured token-level lexical patterns, which served as input for Logistic Regression, SVM, and XGBoost. This representation is particularly suitable for high-dimensional sparse models.

4.4 Classical Machine Learning Models

4.4.1 Logistic Regression (Baseline Model)

Logistic Regression was selected as the baseline due to its strong performance on high-dimensional sparse text representations. Using TF-IDF features, the classifier was trained to separate the two classes (AI-generated vs. human-written) through simple linear decision boundaries.

Its interpretability, low computational cost, and robustness make it an ideal reference point for evaluating more advanced models.

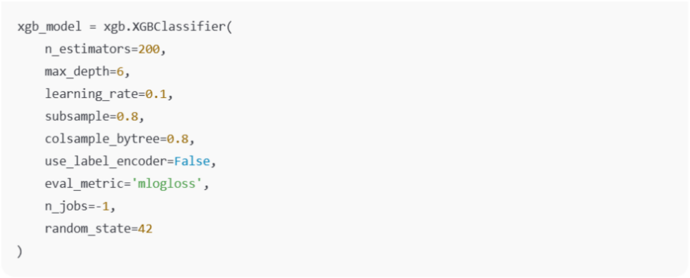
4.4.2 Support Vector Machine

To accelerate training and improve generalization, SVM was coupled with Truncated SVD (Latent Semantic Analysis) to reduce the feature space.

Truncated SVD projects the high-dimensional TF-IDF matrix into a 300-dimensional latent space, capturing semantic structure while improving training speed. This pipeline enables the Linear SVM model to operate more efficiently on large datasets without sacrificing much accuracy.

4.4.3 XGBoost Classifier

XGBoost was used as an advanced ensemble method optimized for handling large and sparse TF-IDF vectors. It builds successive boosted decision trees that correct previous errors.



```
xgb_model = xgb.XGBClassifier(
    n_estimators=200,
    max_depth=6,
    learning_rate=0.1,
    subsample=0.8,
    colsample_bytree=0.8,
    use_label_encoder=False,
    eval_metric='mlogloss',
    n_jobs=-1,
    random_state=42
)
```

Its scalability and ability to model non-linear patterns make XGBoost especially suitable for stylistically diverse AI-generated texts.

4.4.4 Feedforward Neural Network (FFNN)

To incorporate contextual information, we built a simple FFNN classifier using BERT embeddings as input. Two strategies were used:

1. **Fixed BERT embeddings:**
   BERT is used as a feature extractor, and the embeddings are passed to a fully connected network.
2. **Fine-tuned BERT model:**
   The pre-trained BERT model is fine-tuned directly on the classification task to optimize contextual representations.

The FFNN structure includes:

- Dense layer (ReLU activation)
- Dropout regularization
- Output sigmoid layer (binary classification)

This setup enables the model to capture semantic differences between natural and artificial writing styles.

5. Results and Analysis

5.1 Logistic Regression Model

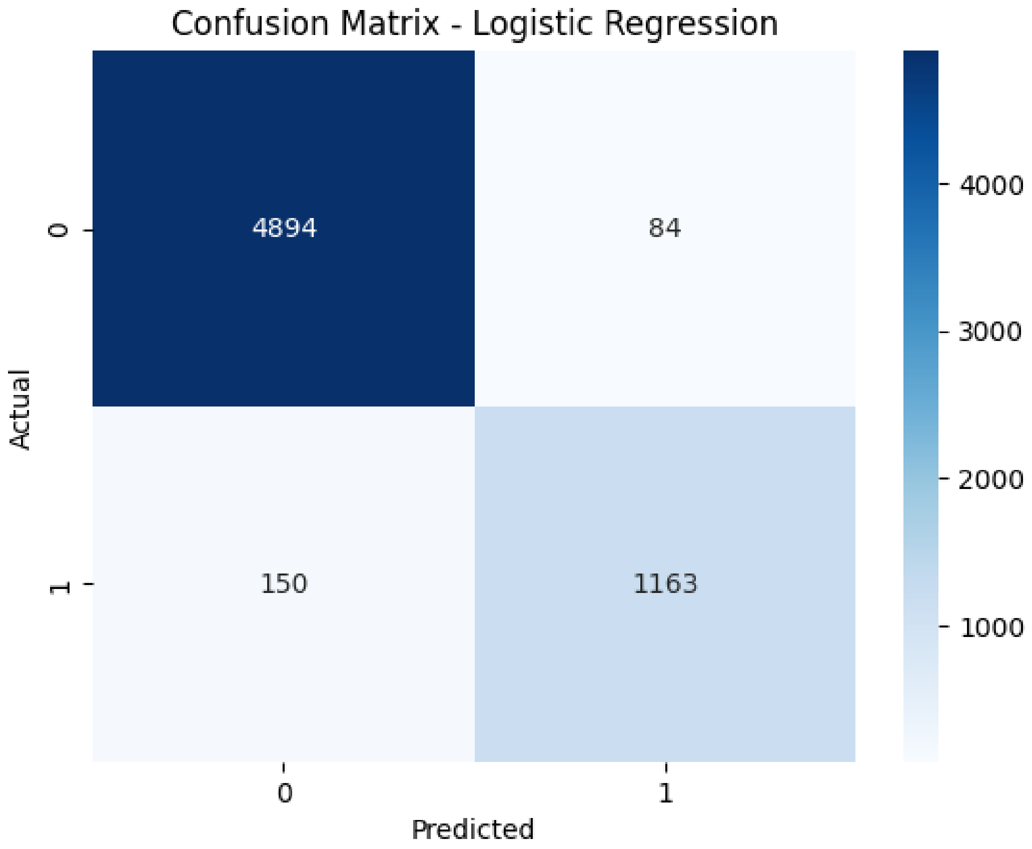
Validation Accuracy:0.9628

Logistic Regression performed remarkably well, offering a balanced trade-off between simplicity and accuracy.

Class	Precision	Recall	F1-score	Support
0	0.97	0.99	0.98	4977
1	0.94	0.88	0.91	1314

The model shows excellent performance in predicting human-written text (class 0), with lower recall for AI-generated text (class 1), suggesting some overlap in linguistic patterns.

Confusion Matrix



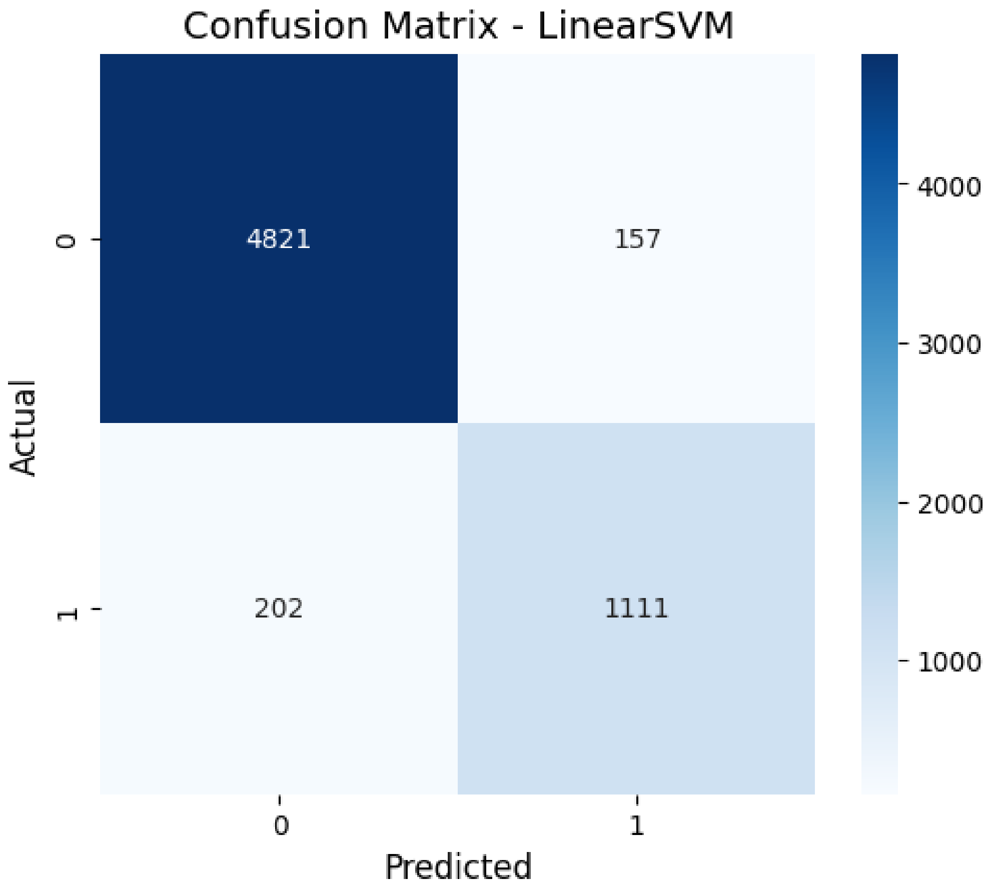
5.2 Linear SVM with SVD

Validation Accuracy: 0.9433

Class	Precision	Recall	F1-score	Support
0	0.96	0.97	0.96	4977
1	0.88	0.85	0.86	1314

SVM performed well but slightly below Logistic Regression. The dimensionality reduction improved computational efficiency but may have removed some subtle linguistic signals important for minority class detection.

Confusion Matrix



5.3 XGBoost

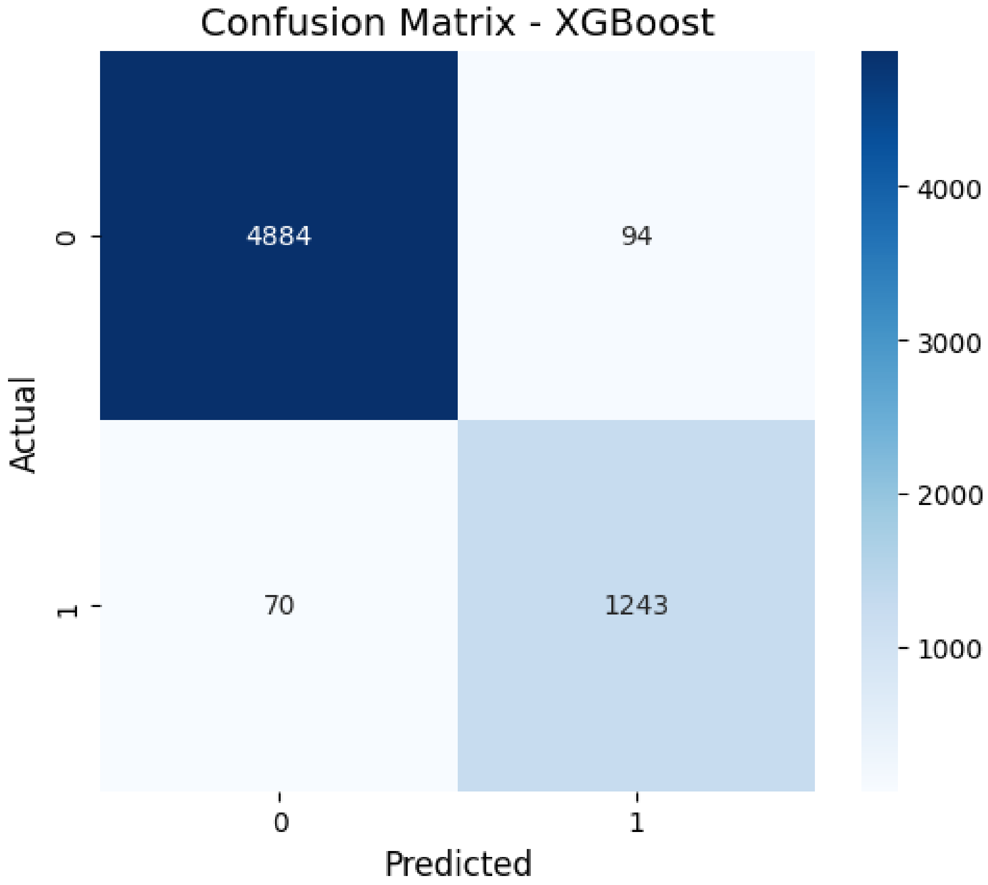
Test Accuracy: 0.9739

Class	Precision	Recall	F1-score	Support
0	0.99	0.98	0.98	4978

Class	Precision	Recall	F1-score	Support
1	0.93	0.95	0.94	1313

XGBoost achieved the highest accuracy, showing excellent performance on both classes. Its ensemble nature likely captured non-linear stylistic patterns missed by linear models.

Confusion Matrix



5.4 Deep Learning Models

The deep learning model results as follows:

Class	Precision	Recall	F1-score	Support
0	1.00	0.99	0.99	4978
1	0.95	1.00	0.98	1313
Accuracy			0.99	6291

The deep learning model performs well for the majority class (0) but shows lower recall and F1-score for the minority class (1).

5.4 Summary of Model Comparison

Model	**Accuracy**
Logistic Regression	0.9628
SVM	0.9433
XGBoost	0.9739
Deep Learning	0.98

5.6 Discussion

The results confirm that embedding-based classification is highly present highest accuracy for AI-text detection in Arabic as the XGBoost model, which presents the highest performance.

6. Conclusion

This study explored the effectiveness of multiple machine-learning and neural models for detecting AI-generated Arabic text. Logistic Regression provided a strong and efficient baseline, SVM offered competitive performance with reduced computational requirements, and XGBoost delivered the best accuracy of the classical machine learning algorithms. The Feedforward Neural Network built on BERT embeddings presents the value of contextual representations in capturing subtle differences between human and AI writing styles, the model achieved the highest accuracy, scoring 0.98 (98%). This performance suggests it is the most effective model for this specific classification task.

The findings confirm that both classical and modern approaches can successfully address the challenge of Arabic AI-text detection, especially when paired with strong feature extraction methods. As generative models continue to evolve, future research should explore hybrid ensemble systems, task-specific transformer architectures, and larger multilingual datasets to continue improving detection accuracy and robustness.

7. Future Work

- Cross-Domain and Cross-Language Generalization:

While the current work focuses on a specific dataset and language setting, future research may evaluate the models on multilingual corpora or domain-shifted data. Techniques such as multilingual embeddings, cross-lingual transfer learning, and domain adaptation could enable wider applicability of the system.

- Real-Time Inference Optimization:

Building lightweight versions of the best-performing models would allow deployment in real-time applications. Approaches such as model pruning, quantization, and knowledge distillation can significantly reduce inference latency while maintaining acceptable accuracy.

- Automated Feature Discovery and Neural Feature Selection:

Future experiments could explore neural architecture search (NAS) or automated machine learning (AutoML) frameworks to automatically discover high-value feature combinations. These methods may uncover interactions between textual and linguistic features that are not easily identified manually.

- Robustness and Adversarial Evaluation:

Additional work is needed to assess the system’s resilience to noisy, intentionally manipulated, or adversarial text inputs. Conducting adversarial robustness evaluations and implementing defensive training strategies may increase reliability in real-world environments.

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]

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.