Journal of Information Systems and Telecommunication
Vol.14, No.1, January-March 2026, 59-66
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Distinguishing Human from Bot Texts: A Graph-Based and Few-Shot Learning Approach

Ohood Al-Minshidawi[1], Abdol-Hossein Vahabie[2*]

[1]. Computer Engineering Department, College of Alborz, University of Tehran, Tehran, Iran
[2]. Computer Engineering Department, College of Alborz, & Electrical and Computer Engineering (ECE) Faculty, College of Engineering, University of Tehran, Tehran, Iran

Received: 17 Mar 2025/ Revised: 04 Apr 2026/ Accepted: 06 May 2026

Abstract
This study examines the identification of human-generated versus bot-generated content on Arabic social media. The rise of bot accounts and AI-generated writing has facilitated the spread of false information, and the limited reliable Arabic data, as well as linguistic complexity, hinders annotated-data collection. Researchers using traditional supervised and deep learning for Arabic bot detection often rely on computationally expensive methods and large annotated datasets, complicating the evaluation of few-shot learning as an alternative. This study proposes a framework to compare methodological paradigms for classifying Arabic bots using the AutoTweet-Dataset-v1.0. The framework contrasts "graph-based" deep learning methods (e.g., Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT)) with few-shot learning based on the SetFit model (one of three methods tested). It combines a semantic graph representation with a data-efficient few-shot learning framework for Arabic bot classification. The primary finding is that SetFit achieved the highest accuracy (88.35%), outperforming both GCN and GAT. The results suggest that transformer-based few-shot learning offers scalable, effective solutions for identifying bots in low-resource settings and can improve the moderation and integrity of Arabic social media.

Keywords: Arabic Text Bot Detection; Graph Neural Networks; Graph Attention Networks; Graph Convolutional Networks; SetFit Model.

1- Introduction

Distinguishing human-generated from AI-generated Arabic text on social platforms represents a significant research challenge. The overwhelming number of machines producing tweets and other forms of material across various platforms, including Twitter, has facilitated the propagation of misinformation, spamming, and the spread of manipulated narratives [1]. Although social bots have some genuine applications (such as supplying news updates automatically and passing on information), there are many documented cases of these bots being purposefully misused for disinformation purposes and manipulating public opinion [2,3]. Therefore, establishing a reliable way to identify bot-generated Arabic content is vital in maintaining the credibility of online platforms.
This task becomes even more complicated with respect to Arabic as it is classified as a low-resource language [4]; hence, its morphology is difficult, it has a diverse range of dialects that do not share similarities with each other, and there is limited availability of annotated datasets [5]. The advent of Large Language Models (LLMs) provides tremendous fluency and sophistication in the production of Arabic text, making distinguishing between machine-generated and human-generated enormously difficult [6]. Most existing evaluations of AI-generated content for the purpose of establishing its distinction from human authorship have tended to primarily focus on English and similar Latin script languages, leaving the examination of AI-generated Arabic text comparatively unexplored [7].
Research has focused on applying supervised machine learning, deep learning, and transformer-based approaches to develop bot detection systems. Although there are positive findings presented in many of these publications, there are several limitations shared among the studies, including: (1) reliance on large labelled datasets that are resource-intensive, which are often not available for Arabic; (2) many models have focused on utilizing user metadata or behaviour metrics

✉ **Abdol-Hossein Vahabie**
h.vahabie@ut.ac.ir

to distinguish human-generated text from bot-generated text; and (3) few studies have compared the performance of graph-based relational modelling methods to that of data-efficient few-shot learning approaches for Arabic text classification [8,9]. The aforementioned gaps provide motivations for the current study.

Therefore, this study compared the effectiveness of several approaches to classify text written by humans or bots using two different classifications: Graph Convolutional Networks (GCN), Graph Attention Networks (GAT)), and SetFit model, whereas model GCN or model GAT utilizes graph-based learning and model SetFit utilizes few-shot representation learning.

The design of this paper consists of the following sections: In Section 2, we give an overview of the currently known literature on bot detection; In Section 3, we present our proposed techniques; In Section 4, we describe our evaluation methods for these implementations; and finally, in Section 5, we present some conclusions and provide suggestions for future work.

2- Related Works

AI-generated text detection researches have progressed along three major directions: (i) traditional machine learning techniques that rely on feature engineering [10], (ii) content models built using deep neural networks and transformers, and (iii) relational graph-based methods. Progress has been made with existing methods, but their limitations persist, especially in low-resource Arabic environments.

2-1- Feature Engineering and Traditional Machine Learning

The development of early methods relied on hand-crafted features to produce a final output using classical algorithms for classification. Bhandarkar et al. [11] used Count Vectorizer features to calculate bag-of-word counts with a Multinomial Naïve Bayes classifier to classify text, achieving an accuracy rate of 86.2%. While they provide an efficient and interpretable way to work with the data, bag-of-words representations are unable to capture any deeper contextual meaning or dependencies between words and can be fooled by highly stylized AI-generated text. Alhayan et al. [12] evaluated traditional machine learning, deep learning, transformer, and hybrid models for analyzing Arabic reviews of e-commerce products. Despite achieving comparable results using an ensemble method (Logistic Regression (LR) + Convolutional Neural Network (CNN)) with approximately 89.7% average accuracy, these approaches relied on annotated training datasets, which hampered their generalization ability to new generative models where the data had changed and evolved. It is evident that both feature-engineered and traditional machine-learning models retain competitive performance but are limited in their ability to produce contextualized and domain-independent transfer learning results. This is especially problematic due to the rich morphological nature of the Arabic language.

2-2- Deep Learning and Transformer-Based Content Modeling

Typically, contemporary studies are oriented towards contextual representation learning. For example, Harrag et al. [13] validated AraBERT's fine-tuned performance at detecting GPT-2 generated Arabic tweets. Similarly, Alshammari et al. [14] evaluated AIRABIC, a balanced Arabic benchmark dataset designed with specific diacritics included; fine-tuned XLM-R was shown to outperform commercial systems for detection use. Research demonstrating transformer-based contextual embeddings' capacity to capture subtle stylistic variations supports both Harrag's and Alshammari's results. Despite these advancements in the use of transformer-based contextual embeddings, there are still limitations that hinder content modeling. First, most studies rely on synthetic data created by one model (e.g., GPT-2 or ChatGPT), which may present a limitation concerning the robustness of detectors' ability to detect different generative systems. Second, most studies utilize completely supervised training (with relatively large labeled datasets), an issue often encountered in Arabic language situations where sufficient labeled datasets are not commonly available in practice. Additionally, the majority of previous approaches have focused on content modeling exclusively and neglected the exploration of relational patterns within the corpus.

Wei et al. [15] presented BOTLE, a framework based on content that does not require feature engineering. The system utilizes a multi-embedding Bidirectional Lightweight Gated Recurrent Unit (BiLGRU) architecture and thus performs very well, although the performance is highly dependent on the quality of linguistic pre-processing (Part-of-speech (POS) / Named Entity (NE) tagging), and the evaluations were all performed against a single benchmark dataset; thus, there are substantial concerns about the robustness of the results between domains.

2-3- Modeling with Graphs & Relations

Using graphs enables the representation of the relationships between users and content through relational modeling. Alashwal [16] introduced Bot-MGAT, which is a semi-supervised multi-view graph attention network that leverages users' interaction graphs and profile metadata. Bot-MGAT showed transferable learning across TwiBot-20 through the use of semi-supervised learning. Although using relational modeling can increase the robustness and accuracy of relational modeling approaches, they also

introduce practical limitations. For a graph-based system to function properly, large amounts of data, social networks (friendships), and interaction structure must exist at a large scale. Large-scale metadata and social networks may not exist or may contain too much noise or incomplete information. Difficulties may arise when differences in relational properties exist across platforms, as this can affect the scalability and domain portability of a graph-based system. Additionally, most studies do not evaluate their methodology based on purely content-based Arabic benchmarks.

To better understand the differences between textual and lexical relational signals, this study analyzes the AutoTweet dataset, which does not have a rich amount of metadata for user-nodes (users who are actually active social media users). A controlled comparison is conducted between two different types of relational modeling methods (i.e., GCN and GAT), constructed on text structural similarity, compared with fine-tuning on semantic adaptation, using few-sample training methods (e.g., SetFit). The current research methodology and study determine whether corpus-based relational modeling is more beneficial than sentence-level contrastive semantic learning for low-resource Arabic detection of bots. A summary of the results and analytics methodology/results is presented in the Appendix. As illustrated in Figure 1, prior work can be categorized into feature-based, transformer-based, and graph-based approaches, while our study integrates relational modeling and few-shot semantic learning under content-only Arabic settings.

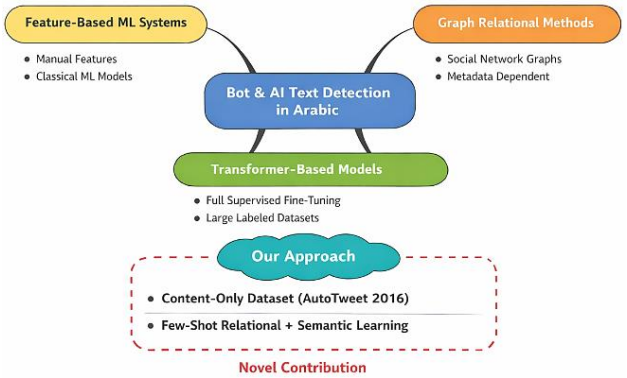

Fig. 1. The existing approaches to AI-generated text detection.

3- Methodology

The central research objective is to evaluate whether data-efficient few-shot learning or graph-based relational modeling is more effective for distinguishing human-generated from bot-generated Arabic text in a low-resource setting. This section describes the methodological framework designed to address this objective. Specifically, this study investigates two complementary modelling paradigms:

1. First, using Graph Convolutional Networks (GCN; GAT) to determine whether modelling the relational structure (i.e., word-document relationship at the corpus level) is sufficient for reliably detecting human versus bot content.
2. Second, few-shot representation learning via sentence-level representations (via SetFit) to determine if a limited set of labelled data for training, along with contrastive-based fine-tuning, provides additional discriminative capability to separate human from bot content.

The modeling paradigm was the primary factor that was examined in the experiments, whereas the overall classification results (as measured by accuracy, precision, recall, and F1-score) served as the evaluation criteria for how well the different paradigms performed. The overall pipeline for the experimentation is shown in Figure 1.

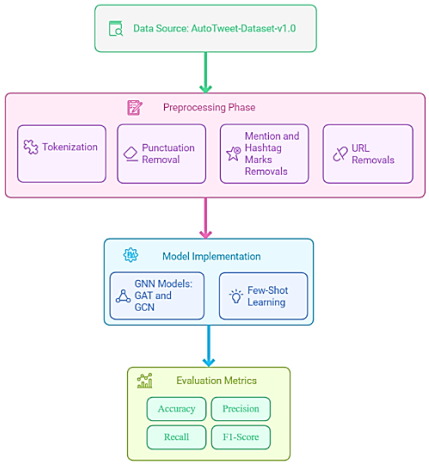

Fig. 2. Our approach for detection of bots and humans in Arabic text.

3-1- Dataset and Preprocessing

The experimental dataset used for this study was the AutoTweet-dataset-v1.0 [17], which contains Arabic tweets collected through Twitter's public API. From this dataset, we extracted a large number of content samples generated by both humans and bots. The experimental dataset had 2,627 total samples in the train dataset and 876 samples in the test dataset. A 75/25 train-test split maintained the class balance in the final dataset. During the pre-processing phase, several steps were taken to ensure the reduction of noise in the training and testing datasets. First, the numerical representation of the text (using UTF-8 encoding) was standardized, followed by transforming all text to a lower-case representation and removing all URLs and all punctuation characters. Following that, all mentions and hash marker identifiers were replaced with their equivalent

text string representation. The text was then tokenized, removing from the graph vocabulary any words that occurred less than 15 times within the training dataset, to further help reduce sparsity. These preprocessing steps led to cleaner, more structured input data, improving the model's ability to detect relevant patterns [18].

3-2- Implementation

3-2-1- Graph-Based Relational Modeling (GCN and GAT)

Bot detection is relational because the way a user uses words typically looks the same, or has nearly identical word usage patterns across the entire collection of data known as "the corpus." To utilize these corpus-level dependencies, we created a heterogeneous graph G = (V, E), which represents all the tweets in our dataset as document nodes and all the words used to create them as word nodes.

We assigned weights to the edges between document and word nodes based on their Term Frequency-Inverse Document Frequency (TF-IDF) [19] score. The edges between word nodes were created based on statistical counts of when words appear in relation to each other, based on a sliding window of 20 words. This composite graph allowed us to display context-related relationships between the words as well as highlight each word's relation to the others across the entire collection of documents.

Node features were initialized using contextual embeddings created from Sentence-BERT (SBERT) [20], specifically four pre-trained models: 1.) all-MiniLM-L12-v2 [21] (384 dimensions), 2.) paraphrase-mpnet-base-v2 [22] (768 dimensions), 3.) paraphrase-multilingual-mpnet-base-v2 [23] (768 dimensions), and 4.) gtr-t5-large [24] (768 dimensions). For the document nodes, the embeddings were calculated using the mean of all tokens that made up the whole tweet text, while for the word nodes, we used the original tokens for each individual token. We used the SBERT encoders as-is with no further fine-tuning while we were training the GNN, thereby allowing for the GNN to take place on frozen semantic representations.

In tuning our learning algorithms, we implemented two GNN architectures in PyTorch Geometric. The first architecture, the GCN model, used a single GCNConv layer that took the input embeddings and projected them into a 100-dimensional hidden layer. This was followed by a ReLU activation, dropout (p = 0.5), and a fully connected layer mapping to two output classes. The second architecture, the GAT model, utilized a single GATConv layer that projected the input embeddings into a 50-dimensional representation, followed by attention-based aggregation of neighbor nodes, and finally the same classification head as the GCN Model.

Both models were trained using negative log-likelihood loss in combination with the Adam optimizer ($\alpha = 0.001$; $\beta_1 = 0.9$; $\beta_2 = 0.999$, and no weight decay). While the GCN was trained for 50 epochs, GAT required 500 epochs because it converged more slowly. The dataset was split 90:10 into training and validation datasets, and no early stopping or systematic hyperparameter search was done, with the final evaluation being performed on a held-out test set.

The GCN and GAT model configuration and key hyperparameters of the implementation are in Tables 1. Each model had a single graph layer (i.e., aggregators) with ReLU activation and dropout regularization. The primary architectural difference between GCN and GAT was that GAT used an attention mechanism, whereas GCN did not; therefore, maintaining consistency across optimization parameters in both models allowed for an effective comparison between convolutional-based relational aggregation (GCN) and attention-based message passing (GAT).

Table 1. The hyperparameters used in the graph-based implementation step.

Model	Hidden Dim	Dropout	Optimizer	Learning Rate	Epochs
GCN	100	0.5	Adam	0.001	50
GAT	50	0.5	Adam	0.001	500

3-2-2- Few-Shot Learning (SetFit)

We explored the data-efficient transformer by utilizing the SetFit framework. Utilizing Contrastive Siamese [25] fine-tuning of a sentence transformer and fitting it with a lightweight classifier as a head makes this method ideal for low-resource situations where little Arabic bot data exists, and labeling this data is not widely performed.

We used the sentence-transformers/all-MiniLM-L12-v2 model as our source encoder. We trained our encoder for a few-shot system using 200 labeled samples per class (400 total training instances). During contrastive fine-tuning, we trained the encoder for five epochs using a batch size of 16. The optimizer for the encoder was AdamW with default settings as described in the SetFit implementation of this library. After creating the contrastive representations, we fitted a classifier head on top of the frozen embeddings. We evaluated two different types of classifiers at this stage: LR and Multi-Layer Perceptron (MLP) with a maximum of 500 iterations, and we trained the classifier for 15 epochs at this stage. We set the random seed to 444 to allow for reproducibility of the results of each classifier. The hyperparameters for both classifier types are summarized in Table 2. The goal of this model was to determine whether learning semantics at the sentence level with low supervision results in an accurate method for detecting Arabic bots without needing to construct an explicit relational graph structure. The experimentation was done using NVIDIA H100 GPUs with a total GPU memory of 80GB and 4 CPUs and a total of 50GB CPU memory.

Table 2. The hyperparameters used in the few-shot learning implementation step.

Parameter	Value
Base Encoder	all-MiniLM-L12-v2
Few-shot samples	200 per class
Batch size	16
Contrastive epochs	5
Classifier	LR / MLP
MLP max_iter	500
Classification epochs	15
Optimizer	AdamW (default)
Random seed	444

4- Results and Discussion

The performance of all models was measured by calculating accuracy, precision, recall, and F1 score with respect to the held-out test set. As shown in Table 4, the top-performing model, SetFit, achieved F1 and accuracy scores of 88.11% and 88.35%, respectively. Both GCN and GAT performed close to each other, with GAT (F1 = 87.28%) marginally outperforming GCN (F1 = 87.16%). GAT's minor but consistent edge relative to GCN is likely due to the attention mechanism in GAT allowing for the use of different weights when aggregating the neighborhood information of a node. In contrast, the weights assigned in aggregation by GCN are uniform across all nodes. For example, when detecting whether text is human-authored or machine-generated (bots), certain lexical nodes (i.e., repetitively used promotional language and automated language) may provide a stronger indication relative to other lexical nodes in determining the difference between the two types of authorship. GAT's attention mechanism thus allows it to weigh the contribution of such important neighbours greater than that of less informative neighbours, resulting in slightly better accuracy. The relatively small performance difference may suggest that the relational nature of the graph (i.e., how the examples are related to one another) has a more significant impact on the overall performance of GNNs than the manner in which the aggregation is performed. SetFit outperformed both GNN-based models by about 0.8%–1% with respect to F1-score, and this is likely due to being able to model sentence-level semantics through contrastive fine-tuning of transformer embeddings. In contrast, the GNN-based models primarily rely on lexical co-occurrence patterns represented within the graph structure. While GNN-based models are capable of identifying the types of words that co-occur together, they are unable to discern patterns that occur within a single sentence since they are unable to create representations at this level. In contrast, SetFit is able to directly optimize the representation of sentences for the detection of bots vs humans and thus, be able to identify patterns at a higher level (both semantically and stylistically) beyond simply an analysis of word frequency statistics. In addition, the few-shot training paradigm allows the encoder to concentrate on user-level distinctions relevant to the task, enabling the capture of fine-grained stylistic cues typical of automated accounts. Conversely, graph-based models rely heavily on the stability of co-occurrence statistics, which may not be stable enough for use in short and noisy social text data. By adding context to the embedding, transformer-based embeddings can also operate more effectively with sparse and varied lexicons. Using limited supervision, semantic representation learning has greater discriminative power than purely relational lexical models for detecting Arabic bots in social media content that is written in short form. Additionally, we compared our results with those of Hassan et al. [26], who used traditional machine learning, ensemble, and deep learning on the same dataset. The best results by their support vector machine (SVM) with unigram features on non-preprocessed text achieved an accuracy of 83.11% and an F1-score of 82.71%. All three models we proposed provided significantly better performance than this baseline, offering approximately a 4-5% F1-score improvement per model. Notably, Hassan et al.'s model achieved a precision rate of 95.16%, while it attained a rather low rate of 73.14% recall, which indicates that their model conservatively labels AI-generated texts and failed to detect a substantial proportion of automated accounts. In contrast, we have provided a more balanced ratio of precision to recall, thereby improving the overall F1-score for our models. Hassan's results indicate better performance using contextual embedding methods (SetFit) and relational graph models (GAT/GCN) as compared to either bag-of-words methods or shallow classifiers. These results demonstrate that both semantic representation and relational structure provide additional discrimination power in identifying Arabic AI-generated texts. The data clearly demonstrate this performance order: SetFit > GAT > GCN > traditional machine learning baselines. It is essential to note that while relationally modeled data provide meaningful contributions to system performance, the largest gains in our system performance were obtained via the application of task-specific semantic fine-tuning in the implementation of few-shot paradigms. Thus, we indicate that sentence-level semantic adaptation is more important than lexical propagation through graph structures alone.

Table 3. Results on test sets for proposed models.

Model	Accuracy	Precision	Recall	F1-Score
GCN	87.44%	86.96%	87.43%	87.16%
GAT	87.55%	87.07%	87.60%	87.28%
SetFit	**88.35%**	**87.88%**	**88.45%**	**88.11%**
Hassan et al. [26]	83.11%	95.16%	73.14%	82.71%

5- Conclusion

This research explored two approaches to identifying AI-generated Arabic text: the relational graph modeling method, which included the GCN and GAT, and data-efficient representation learning through the SetFit model. The results have produced a clear ranking of the three models based on their F1-scores, which reveal that SetFit achieved superior performance (88.11%) compared to GCN and GAT.

From these results, there are meaningful scientific contributions. SetFit's successful performance illustrates that semantic adaptation at the sentence level through contrastive fine-tuning can produce greater discrimination than simply using word co-occurrence frequency. GCN and GAT each capture relational structure within the larger corpus of text; however, their reliance on statistical associations among the constituent terms limits their ability to capture other deeper structural and contextual characteristics. On the other hand, the few-shot learning process associated with the transformer model directly adjusts contextual representations of words/phrases to fit the characteristics of the target task, affording greater flexibility to process shorter, noisier social media posts. Additionally, the minimal increase in performance obtained using GAT as compared to GCN indicates that while relational structure is advantageous, improvements are obtained via adaptive attention.

These results imply that, for low-resource Arabic bot detection, task-specific semantic fine-tuning plays a more critical role than structural graph propagation alone. Accordingly, our research provides empirical data supporting the efficacy of employing contrastive few-shot learning for the classification tasks of low-resource NLP.

5-1- Limitations

Although there were positive outcomes, this study has several limitations. First, the AutoTweet dataset was produced in 2016; the language used in tweets created by bots has changed over time due to newer methods of creating AI-generated text (such as ChatGPT).

Second, the focus of the research was on short, structured tweets; however, this differs from long-format literature, such as Arabic newspapers or magazines, or conversational dialogue (casual back- and- forths) or content from different types of AI-generated systems (GPT-4), which can further limit the potential of cross-validation due to differences in writing formats. Third, correlation testing and cross-domain validations were insufficient, thus making it difficult to determine if there were any limitations to cross-domain validity within any sample of data collected in this dataset (due to insufficient data).

5-2- Future Works

Future work should focus on three areas. (1) Cross-domain validity-testing can be accomplished using only a few-shot semantic learning across various types of content, including many modern Arabic-language literature samples and various forms of free-written (non-structured) dialogue. (2) Further studies to evaluate generative AI models, such as large-scale language models, through the use of three different evaluation conditions (zero-shot, few-shot, and fine-tuning) should be conducted to determine whether recent generative AI systems alter the detection landscape. (3) Combining relational graph methods with contrastively fine-tuned semantic (word-based) encoding would also have the potential for greater robustness via leveraging both corpus-level structure and contextual representation learning. Addressing these directions will enable progress toward more adaptive and future-proof systems.

References

[1] A. Nambiar, "Impact of fake news, message and spam spread through social media on people decision making ability," 2022.

[2] D. Assenmacher, L. Clever, L. Frischlich, T. Quandt, H. Trautmann, and C. Grimme, "Demystifying social bots: On the intelligence of automated social media actors," Social Media+ Society, vol. 6, no. 3, p. 2056305120939264, 2020.

[3] D. Ajiga, P. A. Okeleke, S. O. Folorunsho, and C. Ezeigweneme, "The role of software automation in improving industrial operations and efficiency," International Journal of Engineering Research Updates, vol. 7, no. 1, pp. 22-35, 2024.

[4] S. S. Sabr et al., "A Comprehensive Part-of-Speech Tagging to Standardize Central-Kurdish Language: A Research Guide for Kurdish Natural Language Processing Tasks," Journal of Studies in Science and Engineering, vol. 5, no. 2, pp. 15-38, 2025.

[5] N. S. Alghamdi and J. S. Alowibdi, "Distinguishing Arabic GenAI-generated tweets and human tweets utilizing machine learning," Engineering, Technology & Applied Science Research, vol. 14, no. 5, pp. 16720-16726, 2024.

[6] H. Alshammari and K. Elleithy, "Toward Robust Arabic AI-Generated Text Detection: Tackling Diacritics Challenges," Information, vol. 15, no. 7, p. 419, 2024. [Online]. Available: https://www.mdpi.com/2078-2489/15/7/419.

[7] H. Alshammari, A. El-Sayed, and K. Elleithy, "Ai-generated text detector for arabic language using encoder-based transformer architecture," Big Data and Cognitive Computing, vol. 8, no. 3, p. 32, 2024.

[8] H. Alshammari, AI-Generated Text Detector for Arabic Language. University of Bridgeport, 2024.

[9] B. Sani et al., "Who wrote this? Identifying machine vs human-generated text in Hausa," in Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025), 2025, pp. 82-88.

[10] A. A. Abdullah, N. S. Mohammed, M. Khanzadi, S. M. Asaad, Z. K. Abdul, and H. S. Maghdid, "In-depth analysis on machine learning approaches: Techniques, Applications, and trends," Aro-The Scientific Journal of Koya University, vol. 13, no. 1, pp. 190-202, 2025.

[11] A. Bhandarkar, M. A. DM, D. Vishwachetan, A. Mushtaq, D. Kadam, and S. Saxena, "Unmasking the AI Hand: A Machine Learning Approach to Deciphering Authorship," in 2024 3rd International Conference for Innovation in Technology (INOCON), 2024: IEEE, pp. 1-6.

[12] F. Alhayan and H. Himdi, "Ensemble learning approach for distinguishing human and computer-generated Arabic reviews," PeerJ Computer Science, vol. 10, p. e2345, 2024.

[13] F. Harrag, M. Debbah, K. Darwish, and A. Abdelali, "Bert transformer model for detecting Arabic GPT2 auto-generated tweets," arXiv preprint arXiv:2101.09345, 2021.

[14] H. Alshammari, "AI-Generated Text Detector for Arabic Language," University of Bridgeport, 2024.

[15] F. Wei and U. T. Nguyen, "Twitter bot detection using neural networks and linguistic embeddings," IEEE Open Journal of the Computer Society, vol. 4, pp. 218-230, 2023.

[16] E. Alothali, "STREAM-EVOLVING BOT DETECTION FRAMEWORK USING GRAPH-BASED AND FEATURE-BASED APPROACHES FOR IDENTIFYING SOCIAL BOTS ON TWITTER," 2023.

[17] H. Almerekhi and T. Elsayed, "Detecting automatically-generated arabic tweets," in Information Retrieval Technology: 11th Asia Information Retrieval Societies Conference, AIRS 2015, Brisbane, QLD, Australia, December 2-4, 2015. Proceedings 11, 2015: Springer, pp. 123-134.

[18] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," Information processing & management, vol. 50, no. 1, pp. 104-112, 2014.

[19] W. I. D. Mining, "Data mining: Concepts and techniques," Morgan Kaufinann, vol. 10, no. 559-569, p. 4, 2006.

[20] N. Reimers, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," arXiv preprint arXiv:1908.10084, 2019.

[21] "all-MiniLM-L12-v2." https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2 (accessed.

[22] N. a. G. Reimers, Iryna, "paraphrase-mpnet-base-v2," 2019. [Online]. Available: https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2.

[23] N. a. G. Reimers, Iryna, "paraphrase-multilingual-mpnet-base-v2," 2019. [Online]. Available: https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2.

[24] "gtr-t5-large." https://huggingface.co/sentence-transformers/gtr-t5-large (accessed.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in International conference on machine learning, 2020: PMLR, pp. 1597-1607.

[26] S. I. Hassan, L. Elrefaei, and M. S. Andraws, "Arabic Tweets Spam Detection Based on Various Supervised Machine Learning and Deep Learning Classifiers," MSA Engineering Journal, vol. 2, no. 2, pp. 1099-1119, 2023.

Appendix

Comparative Summary of Representative AI-Generated Text Detection Studies

Study	Methodology	Advantages	Limitations
Alshammari [14]	Fine-tuned XLM-R, AraBERT, mBERT + Dediacritization layer	• First balanced Arabic benchmark (AIRABIC) • Systematic handling of diacritics • Strong F1	• Reliance on ChatGPT-generated data • Limited dialectal coverage • Domain shift sensitivity
Alashwal [16]	Semi-supervised Multi-view Graph Attention Network (Bot-MGAT) with transfer learning	• Effective use of labeled + unlabeled data • Strong generalization • High F1	• Requires rich metadata • Scalability concerns • Limited Arabic evaluation
Wei et al. [15]	Multi-embedding (word, char, POS, NE) + BiLGRU	• No handcrafted profile features • Competitive performance	• Evaluated on single dataset • Limited domain generalization • Content-only constraints
Bhandarkar et al. [11]	Count Vectorizer + Multinomial Naïve Bayes	• Computational efficiency • Interpretability	• Bag-of-words limitation • Vulnerable to stylistic evolution • Multilingual weakness
Harrag et al. [13]	Fine-tuned AraBERT vs RNN baselines	• Strong contextual modeling • High accuracy	• Potential overfitting to GPT2 • Limited robustness to unseen generators
Alhayan et al. [12]	ML, DL, Transformer, Ensemble (LR+CNN)	• Broad comparative evaluation • Hybrid effectiveness	• Domain-specific dataset • Limited generalization to evolving generators