AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 57(1) (2025) 113-124
DOI: 10.22060/miscj.2025.24060.5408

Classifying AI-Generated Text in Low-Resource Languages like Arabic

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

[1] Department of Computer Engineering, College of Alborz, University of Tehran, Tehran, Iran.
[2] School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran.

ABSTRACT: AI-Generated Texts (AIGTs) refer to written content produced by artificial intelligence systems using technologies such as natural language processing and machine learning. The rise of AIGT has introduced new challenges in content authenticity, trustworthiness, and information integrity across digital platforms. In low-resource languages, like Arabic, AIGT detection is challenging because of their more complex structural features. Accurate identification of AI-generated versus human-written text is essential to combat misinformation, preserve credibility in communication, and enhance content moderation systems. In this study, we propose a novel framework for AIGT detection on the AutoTweet Dataset, an annotated corpus of Arabic tweets. To the best of our knowledge, this is the first work to leverage Large Language Models (LLMs) for AIGT detection in Arabic, addressing a critical gap in low-resource natural language processing. We introduce a dynamic few-shot prompting technique, powered by a retrieval-based Judge Prompter module, which selects semantically and stylistically relevant support examples to enhance the contextual understanding of LLMs. We conduct a comprehensive evaluation across multiple LLMs, including Mistral-7B, LLaMA-3.1-8B, and ALLaM-7B-Instruct-preview, under zero-shot, few-shot, and fine-tuning scenarios. Our best results were achieved using Mistral-7B with QLoRA fine-tuning and dynamic few-shot prompting, reaching an accuracy of 88.69% and an F1-score of 88.35%. These findings demonstrate the feasibility of adapting LLMs for AIGT detection in Arabic and highlight the effectiveness of context-aware prompting in low-resource settings, paving the way for future progress in text classification.

Review History:

Received: Apr. 23, 2025
Revised: Aug. 26, 2025
Accepted: Sep. 08,2025
Available Online: Oct. 10, 2025

Keywords:

Arabic Text Detection
AI-generated Text
Zero-Shot Learning
Few-Shot Learning
Supervised Fine-Tuning

1- Introduction

Artificial Intelligence-Generated Content (AIGC) refers to the creation of digital media, such as images, music, and text, using advanced AI models [1]. By interpreting user instructions, AIGC generates content that closely aligns with human intent. Unlike traditional content creation, AIGC employs cutting-edge Generative AI techniques to automate the process, allowing for the rapid production of substantial amounts of content [2]. As part of AIGC, content generation bots are automated systems designed to produce and share different kinds of content in online media. If the content is text, known as AI-Generated Texts (AIGTs), these bots use Natural Language Processing (NLP) algorithms to generate diverse content forms such as articles, social media posts, and product descriptions [3]. When integrated with AIGT technology, these bots can deliver highly relevant and responsive content in real time, adapting to user inputs. This integration enhances the scalability and speed of content creation [4].

The advanced capabilities of AIGT bring with them significant ethical challenges [5]. While AIGT's capacity to produce coherent, contextually relevant text has many benefits, it also presents risks, such as the spread of misinformation and fake news. These risks can undermine public trust and skew societal perceptions [6]. Other issues include plagiarism, intellectual property infringement, and the creation of deceptive product reviews, which harm both consumers and businesses [7, 8].

In light of the complex and multifaceted ethical considerations surrounding AIGT, the responsible development and deployment of these technologies are crucial to fully realize the societal benefits they can offer. Recent research is now focused on developing detectors to differentiate between machine-generated and human-authored content [9]. These detection tools are a key safeguard against potential misuse of AIGT, helping to maintain integrity and trust in digital information [7].

Efforts to tackle challenges in AIGT have concentrated heavily on the English language, driven by its global reach and the abundance of extensive, diverse datasets crucial for training powerful language models. Key research areas include enhancing factual accuracy, improving the detection

*Corresponding author's email: h.vahabie@ut.ac.ir

of AI-generated misinformation, and reducing biases in generated content [10, 11]. For languages like Arabic, one of the most spoken languages globally, progress in bot detection is slower and faces unique challenges due to fewer resources, tailored research efforts, and linguistic complexity12]]. The Arabic language's complexity, including its rich morphology, diverse dialects, and unique syntactic structures, complicates the development of accurate AI language models [13, 14]. Furthermore, the lack of extensive high-quality datasets in Arabic, considered a low-resource language, limits the performance of AIGT systems [15]. This data limitation can impede the development of models capable of accurately identifying bot-generated content in the Arabic language [16]. Nevertheless, the potential of models to distinguish between human-written and bot-generated text in Arabic is encouraging, despite these barriers [17].

AIGTs are synthetic texts produced by Large Language Models (LLMs) in response to user inputs [18]. In NLP, LLMs play a key role in distinguishing human-written from AI-generated text [19], often through Transformer-based Frameworks (TF) that detect statistical and linguistic patterns unique to AI output [20]. These detectors assess features like vocabulary, sentence structure, and coherence, with methods such as perplexity scoring frequently used, since AI text typically has lower perplexity due to its probabilistic generation process [1]. For Arabic, LLMs offer a significant advantage by understanding and producing language that closely resembles human communication, because of training on large Arabic corpora [21]. This enables them to identify human-like traits, including cultural references, informal expressions, and Arabic-specific structures [22]. Furthermore, LLMs can analyze semantic and syntactic aspects to detect anomalies indicative of bot-generated content [23], using their deep contextual understanding to spot irregularities, inconsistencies, and unnatural language patterns associated with automated text[24].

In this study, we addressed the critical challenge of detecting bot-generated content in low-resource languages, with a particular emphasis on Arabic. To this end, we made several notable contributions to the field of NLP:

1. Comprehensive evaluation of LLM frameworks: We investigated the capabilities of LLMs under three distinct frameworks, zero-shot prompting, few-shot prompting, and fine-tuning, to evaluate their effectiveness in detecting bot-generated content in low-resource languages.

2. Innovative fine-tuning methodology: In this work, we presented a new dynamic few-shot prompting technique that uses semantic similarity and stylistic compatibility with the test instance to choose the most informative examples from a training set. Unlike static few-shot methods that build static contextual prompts, our approach dynamically selects and builds contextual prompts for every input, improving the interpretability and accuracy of LLM predictions.

This paper is organized as follows: In Section 2, we review related work on the detection of human and bot-generated content using various NLP techniques. Section 3 outlines our methodology, which is based on LLMs. Following this, Section 4 presents our results and conducts a comparative analysis between our methods as well as previous research. Finally, Section 5 concludes the paper and discusses directions for future work.

2- Related works

Initial works of differentiating between human-generated and bot-generated content focused on basic neural language models, such as those pioneered by Bengio et al. [25], which paved the way for applying models to tasks like machine translation. These models, though early stage, laid the groundwork for distinguishing human versus machine language patterns by leveraging neural networks to better understand linguistic structure. Their work laid the foundation for many advancements in language modeling that followed, including the development of more complex architectures like transformers used in today's LLMs. Cutting-edge pre-trained language models encompass masked models (encoders), autoregressive models (decoders), and encoder-decoder models. Each type serves specific functions such as classification or text generation, while encoder-decoder models integrate both roles for improved outcomes [2].

Research on using LLMs to differentiate between human-generated and bot-generated content, especially in Arabic, is still emerging, but several efforts provide a foundation. Alhayan et al. [26] investigated the application of ensemble learning methods to differentiate between human and computer-generated Arabic reviews. The researchers developed a model that combines several machine learning algorithms, such as Naive Bayes (NB), Logistic Regression (LR), Decision Trees (DT), and Random Forest (RF), to improve detection precision. The study aimed to leverage a combination of machine learning and deep learning techniques, including Convolutional Neural Networks (CNN), RNN, LSTM, transformer models, and ensemble methods, to classify Arabic reviews as either human-authored or computer-generated

Alghamdi et al. [27] investigated techniques to distinguish between tweets created by Generative Artificial Intelligence (GenAI) and those authored by humans in Arabic on the X platform (previously known as Twitter). The researchers collected a dataset comprising 375 human-authored tweets and 375 tweets generated by GenAI. To effectively identify the differences between these two categories of content, various machine learning models, including Support Vector Machine (SVM), NB, and DT, were developed and evaluated.

The emergence of transformer models, such as GPT-3 and its successor GPT-4, has markedly enhanced the capacity to effectively represent and analyze natural language models. These models, with billions of parameters, analyze token sequences and context, making it easier to identify inconsistencies typical of bot-generated text compared to human-authored content [28].

Harrag et al. [29] utilized a transfer learning approach with the pre-trained AraBERT model fine-tuned specifically for detecting machine-generated text in Arabic. The model is trained on a dataset that combines human-authored tweets

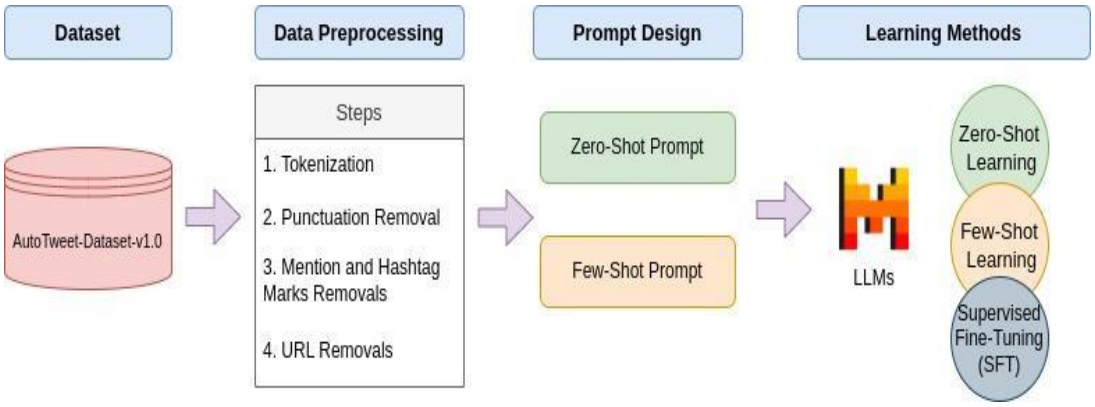

Fig. 1. Methodological framework for end-to-end bot and human detection in Arabic.

with synthetic tweets generated by GPT-2 Small Arabic. The performance of AraBERT is compared against several Recurrent Neural Networks (RNN) baselines, including Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Gated Recurrent Unit (GRU), and Bi-GRU.

Alshammari et al. [30] aimed to create an AI text classifier tailored for the Arabic language, tackling major challenges existing AI detectors encounter with Arabic texts. These challenges include the precise differentiation between Human-Written Texts (HWTs) and AIGTs. The researchers introduced an innovative classifier that leverages two transformer-based models, AraELECTRA and XLM-R. Their study evaluated the performance of these classifiers against current detectors such as GPTZero and OpenAI Text Classifier, achieving an 81% accuracy rate on the AIRABIC benchmark dataset.

Alshammari et al. [18] developed detection models using transformer-based pre-trained architectures, including AraELECTRA, AraBERT, XLM-R, and mBERT. Their goal was to identify AI-generated text in Arabic essays. To achieve this, they constructed innovative datasets that included both discretized and non-discretized texts, acknowledging the additional challenge that diacritics can pose in the detection process. Their experiments demonstrated that models trained on discretized examples could achieve an impressive accuracy of up to 98.4%, significantly surpassing existing tools such as GPTZero, which only achieved 62.7% accuracy on the AIRABIC benchmark dataset.

3- Methodology

Our proposed method for human-bot detection is shown in Figure 1. The AutoTweet-Dataset-v1.0 [31] is used in this method. During the preprocessing phase, we perform several tasks, including tokenization and the removal of punctuation, mentions, hashtags, and URLs. After preprocessing, in the implementation step, we explore three learning approaches using LLMs, which include:

1. Zero-Shot Learning: In this approach, the model classified tweets based solely on a prompt, without any task-specific training.

2. Few-Shot Learning: The few-shot prompt was dynamically constructed with the test tweet and 4 example tweets (2 from each class) that had been purposefully selected with the Judge Prompter module.

3. SFT: In the final stage, we applied SFT using QLoRA [32] to maximize accuracy. QLoRA is a memory-efficient technique that enables the model to adapt its parameters using a larger labeled dataset.

These methods allow the LLM to capture complex stylistic and linguistic cues, significantly improving classification accuracy for Arabic human-bot detection.

3- 1- Dataset

The AutoTweet-Dataset-v1.0 consists of 1,202,815 Arabic tweets collected from 11,764 users over four days using the Twitter streaming API. After filtering for user bias and removing duplicates, 3,503 tweets were labeled as automated (55%) or manual (45%) using the crowdsourcing platform CrowdFlower, with high annotator agreement [31]. Some examples of the data in the dataset are shown in Table 1.

3- 2- Zero-Shot Learning Using LLMs

Zero-shot learning capitalizes on a model's language comprehension to make predictions using the provided prompt information, proving especially useful for tasks with scarce or non-existent labeled data. This strategy has gained more interest in NLP due to its efficiency in enabling LLMs to perform classification with minimal setup [33]. Zero-shot learning will help evaluate LLMs' proficiency in identifying subtle patterns, linguistic nuances, and stylistic features that distinguish human-authored text from bot-generated text, even without prior task-specific knowledge [34].

In our study, we applied zero-shot learning to classify Arabic tweets as either generated by bots or authored by humans, without the need for any task-specific training data. We crafted a simple prompt to guide the model based on its general language understanding, instructing it to categorize the tweet content represented by the placeholder $\{TEXT\}$ in Figure 2. The model is directed to respond solely with "Bot" or "Human," excluding additional text. Zero-shot prompting

Table 1. Some examples of the data in the dataset.

Tweet	Label
RT @real__2012: راموس : الليلة قلنا الى اللقاء لكل النهائيات	Bot
@727Retajالحمره برج الحمره هادى وكل شى فيه نااار والله ..🤦‍♀️..واذا زحمه بس ٢٥ عايله افنيو احط مطار ب قراند يصير دوله	Human
كيف كانت اول ليلة بالسجن وهل القران كان أنيسك ؟ — لا شيء غير القرأن ، والشعور أن الله هو المُدبر وليس أنا.	Bot
قل اللهم مالک الملک تؤتی الملک من تشاء وتنزع الملک ممن تشاء وتعز من تشاء وتذل من تشاء بیدک الخیر إنک علی کل شیء قدیر	Human

```
Arabic Human Bot Detection Task
Given a tweet in the Arabic language, predict whether the tweet is
'Bot', or 'Human'.
Your output should be the 'Bot', or 'Human' without additional text.

### TWEET: {TEXT}
### PREDICTION:
```

Fig. 2. Zero-shot prompt.

leverages the model's pre-trained capabilities without further adjustment, allowing it to generalize across diverse tasks purely through prompts [35].

While zero-shot learning provides a time-saving and cost-effective baseline for LLM-based classification, it may not achieve the specificity that task-tailored fine-tuning can offer. Comparing it with few-shot and fine-tuning approaches can illuminate the advantages and limitations of zero-shot learning [36]. Our objective is to assess LLMs' ability to differentiate between human and bot-generated content, testing the models' language understanding skills specifically in Arabic.

3- 3- Few-Shot Learning Using LLMs

Few-shot learning allows the model to observe a small set of labeled examples, helping LLM identify relevant patterns from limited data and improving classification accuracy without the need for full fine-tuning. This technique enables LLMs to generalize from minimal labeled data, leveraging examples in the prompt to produce accurate task-specific predictions [35].

To enhance the model's ability to perform Arabic human-bot detection based on a small set of knowledge through examples, we implemented a few-shot learning method. Figure 3 shows our designed prompt, allowing LLMs to recognize linguistic patterns specific to each class. In the designed prompt, the exemplar placeholders for few-shot examples are $\{bot-example1\}$, $\{bot-example2\}$, $\{human-example1\}$, and $\{human-example2\}$. We select two examples per class as a representative of bot and human examples.

3- 4- Supervised Fine-Tuning

Despite the advantages of few-shot learning, complex tasks can require domain-specific fine-tuning to allow the model to generalize effectively across diverse tweet structures and patterns specific to Arabic language text [37]. To address these limitations, we employ full fine-tuning of the model on a larger labeled dataset [38]. To fully leverage the model's capacity for Arabic human-bot detection, we implemented SFT of the LLMs.

SFT enables the model to learn from a set of labeled data, allowing it to adapt its internal parameters and capture specific patterns of bot and human-generated Arabic text. This process is particularly beneficial for our task, as it enhances the model's ability to detect complex linguistic complexities that may not be sufficiently learned through few-shot or zero-shot prompting alone [39].

Let $D=\{x'_i, y_i\}_{i=[1,N]}$ denote the labeled dataset, where x'_i represents an input example (an Arabic tweet) and y_i denotes its corresponding label ("Bot" or "Human"). The goal is to adjust the model parameters θ to minimize the loss function L on this dataset as described in equation (1).

$$\hat{\theta} = Argmin \frac{1}{N}\sum_{i=1}^{N} L(f(x'_i;\theta),y_i) \quad (1)$$

Where $f(x'_i;\theta)$ is the model's prediction for input x'_i, given parameters θ, and L is the loss function measuring the difference between the predicted and actual labels.

For fine-tuning, we utilized the QLoRA method, a cutting-edge technique in machine learning, specifically tailored for optimizing LLMs and was introduced in May 2023. This approach significantly enhances efficiency by minimizing memory usage and computational demands, making it an ideal choice for fine-tuning in environments with limited resources. QLoRA's high memory efficiency and reduced computational overhead ensure that even the most extensive language models can be fine-tuned effectively without compromising performance [32]. QLoRA combines two key techniques: Quantization and Low-Rank Adaptation [40].

Quantization reduces the precision of the model's

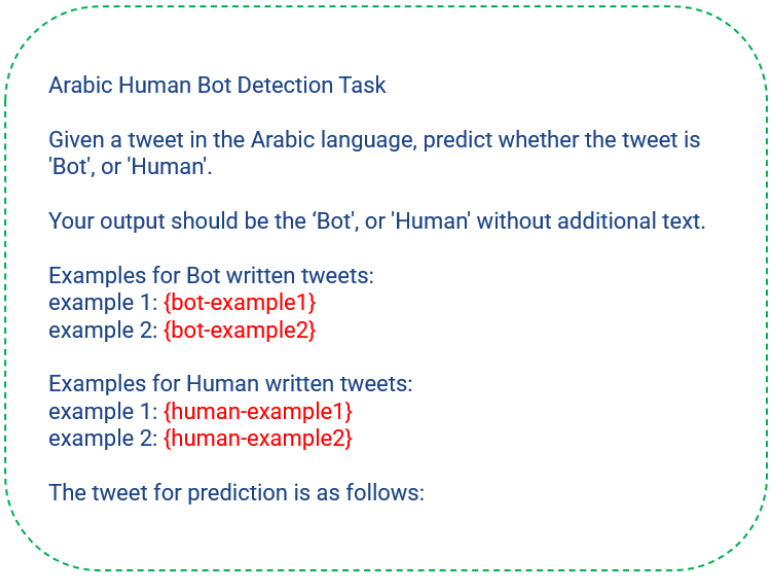

Fig. 3. Few-shot prompt.

Table 2. Distribution of labeled data for training and testing sets.

Split	Bot-Generated Contents No.	Human Written Contents No.	Total
Train set	1432	1195	2627
Test set	512	364	876

weights from floating-point to lower-bit representations, specifically 4-bit in our case. Let W denote the weight matrix of the model; through quantization, W is mapped to a quantized matrix $\hat{W}$, as presented in equation (2):

$$\hat{W} = Quantize\,(W) \tag{2}$$

Here, $Quantize(.)$ represents the function that lowers the bit-width of W. This reduction significantly decreases both memory usage and computational cost, making the model more efficient.

Low-Rank Adaptation approximates weight updates by representing the updated matrix ΔW as a product of two smaller matrices, A and B, where $\Delta W \simeq AB^T$. This approach retains essential information while reducing computational complexity.

Our fine-tuning setup utilized the same prompt structure as in the zero-shot approach (Figure 2) to maintain consistency across different model setups. By iteratively adjusting the model weights during fine-tuning, we enable our LLM to gain a deeper understanding of the distinctive features of Arabic bot and human tweets, achieving a more refined and accurate classification performance.

4- Results and Discussion

4- 1- Implementation

We carried out experiments on NVIDIA H100 GPUs, which offer a total of 80GB of GPU memory, alongside four CPUs that provide 50GB of CPU memory. To facilitate model training and validation, we split the labeled data, using a 75%-25% division for the training and testing sets, as detailed in Table 2. This approach ensured that the class distribution of both automated and manual tweets was preserved in each subset. The test sets were employed for zero-shot and few-shot evaluations, while the training sets were used for fine-tuning.

To test the generalization capabilities of various LLMs, we experimented with three open-source instruction-tuned models. Mistral-7B [41], is a dense transformer language model with 7 billion parameters that achieves competitive performance while being efficient with inference. LLaMA-3.1-8B [42] is a larger-scale model from Meta's LLaMA family

Table 3. Hyperparameter Settings for Fine-Tuning.

Hyperparameter	Value / Setting
Quantization	4-bit
QLoRA Rank (r)	8
Optimizer	AdamW
Learning Rate	2×10^{-4}
Epochs	10
Batch Size	5
Sequence Length	512 tokens

with 8 billion parameters and has improved multilingual understanding, and ALLaMA-7B-Instruct-preview [43] is a family of LLMs trained for Arabic and further fine-tuned for a variety of downstream tasks, including classification.

Table 4 presents the setup for fine-tuning LLMs using the SFT approach. We use QLoRA parameters with a rank $r = 8$ with a 4-bit quantization mode. We incorporated an AdamW optimizer with a learning rate of $2*10^{-4}$. The training process continued for 10 epochs with a batch size of 5, using fixed sequence lengths 512 in the tokenization step of LLM.

4- 2- Judge Prompter

The Judge Prompter module is a fundamental part of our few-shot prompting framework, with the focus on contextualizing test instances with support examples in low-resourced language classification. In this context, the judge module allows it to be a semantic and stylistic filter that allows for the selection of example filters from candidate pools of either class (bot or human) to create effective few-shot examples for LLMs. Classic few-shot learning usually involves the task of either a static set of example prompts or a randomly assigned sample of examples, which can introduce noise and reduce classification performance.

The Judge Prompter would dynamically find support samples that lexically, semantically, topically, and stylistically align to the provided query tweet, and the result would lead to better-informed and interpretable output predictions with the LLMs for the classification decisions. The Judge Prompter works in the following multi-stage process:

1. Preparing Input: First, for the input tweet (the query), we first apply Arabic BERT to compute an embedding. Then, using cosine similarity in embedding space, we find the top-5 similar tweets from the bot class and the top-5 from the human class. This makes for a candidate pool of 10 tweets.

2. Prompt Construction for Judging: The candidate examples, along with the query tweet, were taken into the judging prompt with a structured template, and then the examples were passed into the LLM (e.g., GPT-4o) as the judge, which selects the best-placed support examples. The structured template is defined in Figure 4:

3. Candidate Selection: The LLM interprets the structured prompt and rates all candidate tweets according to the prescribed judgment criteria. To complete the task, the LLM selects 2 tweets from each class that best satisfy the semantic and stylistic alignment of the query. The four examples are intended as the support set for the final few-shot classification.

4. Use of LLM Inference: After the Judge Prompter returns the selected support examples, these are brought together with the query tweet into a few-shot classification prompt and then submitted to the target LLM (Mistral-7B, LLaMA-3.1-8B, or ALLaM-7B-Instruct) for it to predict whether the query tweet was authored by a bot or human in the few-shot context.

The Judge Prompter module of this study is an original contribution that facilitates effective dynamic few-shot learning separately through semantic and stylistic affinity with candidate examples. In doing so, it utilizes prompt-based retrieval and reasoning, bridging the need for supervised learning rigidity against flexible prompt-based experience and interaction abilities in large-scale language models. The steps of the Judge Prompter module are visualized in Figure 5. An example of judging a few shots is presented in Table 4.

4- 3- Prior Works

Almerekhi and Elsayed [31] explored the application of machine learning, ensemble, and deep learning algorithms to classify Arabic tweets as either human- or bot-generated. The authors evaluate models on both preprocessed and non-preprocessed tweets to measure the impact of preprocessing on classification accuracy. In addition to this, they test several tokenization techniques, including unigram, trigram, and Term Frequency-Inverse Document Frequency (TF-IDF), to determine the most effective representation for Arabic text. Their experiments using the temporal feature model (J48) obtained the best results.

Hassan et al [44] addressed the issue by proposing a feature engineering approach that classifies Arabic tweets based on four main categories of features: formality, structural, tweet-specific, and temporal. Various algorithms are tested,

You are given a query text and two sets of related samples, one from the "bot" class and one from the "human" class. Your task is to identify which samples (if any) from each class are most relevant in helping a large language model predict the correct label for the query.

Judgment criteria:
- Lexical and semantic relatedness to the query
- Stylistic correlation (e.g., formal/informal tone)
- Topic similarity
- Usefulness as context to resolve ambiguity in the query

Query:
{tweet}

Bot candidates:
1. {candidate-bot-1}
2. {candidate-bot-2}
3. {candidate-bot-3}
4. {candidate-bot-4}
5. {candidate-bot-5}

Human candidates:
1. {candidate-human-1}
2. {candidate-human-2}
3. {candidate-human-3}
4. {candidate-human-4}
5. {candidate-human-5}

Which samples (from either class) are most helpful? Return your answer as a list of useful samples.

Use the following format:

{{"bot": ["sample-1", ...], "human": ["sample-1", ...]}}
Output two samples per class.

Fig. 4. The structured template.

including NB, SVM, DT, and deep learning models like LSTM and CNN-LSTM. The findings reveal that SVM with unigram tokenization on non-preprocessed data achieves the highest accuracy (83.11%), while the CNN-LSTM model performs well among deep learning approaches (82.65% accuracy).

4- 4- Evaluation

The experimental results summarized in Table 6 demonstrate the clear superiority of our proposed approach compared to both traditional machine learning baselines and prompting strategies. According to the findings, the J48 model achieves a precision of 74.35%, a recall of 74.55%, and an F1-score of 74.00%. The CNN-LSTM model performs significantly better, with an accuracy of 82.65% and an F1-score of 83.89%, establishing a strong classical benchmark. However, our use of LLMs under various prompting and fine-tuning configurations significantly outperforms these earlier approaches. In the zero-shot and few-shot prompting scenarios without instruction-specific fine-tuning, all three evaluated models, namely, Mistral-7B, LLaMA-3.1-8B, and ALLaM-ALLaM-7B-Instruct-preview, exhibited noticeable performance improvements when transitioning from zero-shot to few-shot prompting. For instance, Mistral-7B

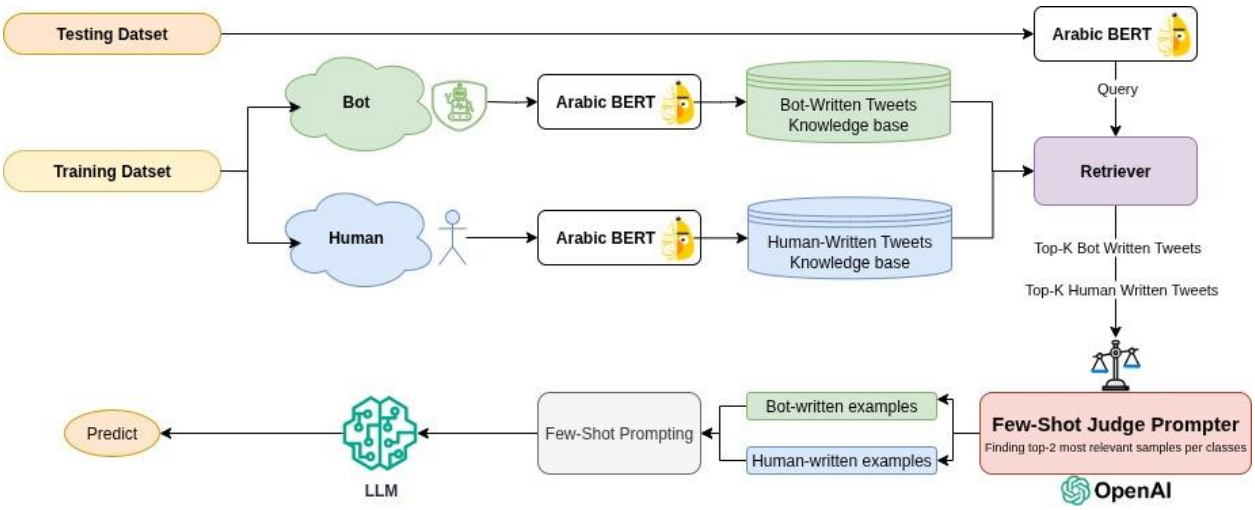

Fig. 5. The steps of the Judge Prompter module.

Table 4. An example of judging a few shots.

Tweet	Label
Arabic Human Bot Detection Task Given a tweet in the Arabic language, predict whether the tweet is 'Bot' or 'Human'. Your output should be the 'Bot', or 'Human' without additional text. Examples for Bot written tweets: example 1: كان رسول الله صلى الله عليه وسلم يتكلم بجوامع الكلم، كلامه فصل لا فضول فيه قلة الكلام مع كثرة المعانى نشرسيرته example 2: قال رسول الله صلى الله عليه وسلم يا عائشة عليك بالرفق وإياك والعنف أو الفحش صحيح البخاري hadith Examples for Human written tweets: example 1: ما ضرب رسول الله صلى الله عليه وسلم شيئًا قط بيده ولا امرأةً ولا خادماً إلا أن يجاهد في سبيل الله نشرسيرته example 2: كان النبي صلى الله عليه وسلم أشد حياء من العذراء في خدرها فإذا رأى شيئا يكرهه عرفناه في وجهه صحيح البخاري Hadith Given tweet for prediction is as follow: ### TWEET: كان صلى الله عليه وسلم يحب الطيب والرائحة الحسنة وكان لا يسلك طريقاً فيتبعه أحد إلا عرف أنه قد سلكه من طيب عرفه نشرسيرته ### PREDICTION:	**Human**
Arabic Human Bot Detection Task Given a tweet in the Arabic language, predict whether the tweet is 'Bot' or 'Human'. Your output should be the 'Bot', or 'Human' without additional text. Examples for Bot written tweets: example 1: RT trll احترم البياض اللي على شنبك💔😂😂😂😂😂 لا يا جدي لا تتحمس شوي example 2: تصير الدقيقة سنة لما ؟ — لما انتظر الاكل يستوي واني ميته جوع Examples for Human written tweets: example 1: RT aa جميله للأحتضان احتضنها ترى ما تعض لا تخاف ريتويت يللي ما تخاف example 2: RT reem 😍 ياخخ توني ايش ذا 😔مزيون ولعبه زين ورايق Given tweet for prediction is as follow: ### TWEET: RT shxii بنات تحبينهم من تويتر ؟ — منار المطيري هيونز تهوميكا شما ام لكع العنوط ابرار النجار مخفوعه انجل tc… ### PREDICTION:	**Bot**

Table 5. Results on test sets for baseline and proposed models.

Model			Accuracy	Precision	Recall	F1-Score
Prior Works		Temporal feature model (J48) [31]	-	74.35	74.55	74.00
		CNN-LSTM [44]	82.65	86.09	81.82	83.89
Our proposed	Mistral-7B	Learning Without Instruction-Specific Finetuning of LLMs				
		Zero-Shot Prompting	0.5011	0.5326	0.5299	0.4978
		Few-Shot Prompting	0.6632	0.6576	0.6281	0.6264
	LLaMA-3.1-8B	Zero-Shot Prompting	0.5696	0.5629	0.5643	0.5626
		Few-Shot Prompting	0.6324	0.6224	0.5894	0.5797
	ALLaM-7B-Instruct-preview	Zero-Shot Prompting	0.5388	0.5435	0.5447	0.5373
		Few-Shot Prompting	0.6529	0.6437	0.6189	0.6171
Our proposed		Fine-tuning of LLMs				
		Mistral-7B	0.8869	0.8840	0.8830	0.8835
		LLaMA-3.1-8B	0.8390	0.8534	0.8182	0.8272
		ALLaM-7B-Instruct-preview	0.8732	0.8685	0.8757	0.8709

improved its F1-score from 0.4978 in the zero-shot setting to 0.6264 in the few-shot setup. Similar gains were observed for LLaMA-3.1-8B and ALLaM-7B, indicating that few-shot prompting, when executed effectively, can enhance model performance even in low-resource language settings such as Arabic.

The most significant results were observed in the fine-tuning scenario using QLoRA, a memory-efficient and effective method for adapting LLMs. Among all models, Mistral-7B achieved the highest performance across all metrics, with an accuracy of 88.69% and an F1-score of 88.35%. Notably, it outperformed ALLaM-ALLaM-7B-Instruct-preview, a model trained for instruction-following tasks. Despite ALLaM-7B-Instruct-preview being part of a family of models tailored for Arabic, Mistral-7B consistently outperformed it across all evaluation metrics. This outcome suggests that Mistral-7B benefits from a more optimized and efficient architecture, which enables better generalization and adaptability during fine-tuning, especially when using parameter-efficient methods like QLoRA. A key innovation that contributed to these improvements is our dynamic few-shot prompting mechanism, enabled by the Judge Prompter module. This dynamic selection process ensures that each test instance is accompanied by a tailored contextual prompt, enabling the LLM to reason more effectively about the classification task.

5- Conclusion

This study addresses the lack of research on applying LLMs to low-resource languages, focusing on Arabic. It presents the first evaluation of LLMs for Arabic AIGT detection, demonstrating that models like Mistral-7B, when fine-tuned with QLoRA, can outperform even Arabic-specific instruction-tuned models. A key innovation is the dynamic few-shot prompting mechanism, powered by the Judge Prompter, which selects semantically and stylistically relevant examples for each query. This approach surpasses static prompting and significantly enhances classification performance in low-resource settings. The findings highlight that effective results come not just from model

size or instruction tuning but from combining efficient fine-tuning, context-aware example selection, and prompt-based reasoning.

Future research holds significant potential for advancing Arabic human-bot detection, improving the adaptation of LLMs to Arabic, and expanding the applicability of these findings to other low-resource languages. To start, a critical step is the expansion of existing datasets coupled with innovative data augmentation strategies [45, 46]. One of the main challenges in Arabic NLP is the scarcity of diverse, high-quality labeled data. Research should explore techniques such as synonym replacement and LLM-based data synthesis to enhance Arabic text datasets. Furthermore, using advanced LLMs to generate synthetic Arabic datasets can effectively mimic both human-like and bot-like text patterns, thereby refining the training process for practical applications. Additionally, since human communication often involves subtle differences in language and culture that robots strive to mimic, it is crucial to develop models capable of recognizing cultural distinctions in expression, accent, and various dialects. Recent studies indicate that advanced models like GPT-4 and LLaMA variants are starting to tackle these complexities [21, 47, 48]. However, there remains a need for more comprehensive benchmarks to evaluate their performance in Arabic language applications. By incorporating these strategies, we can enhance the accuracy of models in recognizing human-bot interactions in Arabic, thus laying the groundwork for similar methodologies in other low-resource languages and significantly broadening the impact of this research.

References

[1] J. Wu, W. Gan, Z. Chen, S. Wan, H. Lin, Ai-generated content (aigc): A survey, arXiv preprint arXiv:2304.06632, (2023).

[2] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P.S. Yu, L. Sun, A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt, arXiv preprint arXiv:2303.04226, (2023).

[3] S. Kumar, S. Garg, Y. Vats, A.S. Parihar, Content Based Bot Detection using Bot Language Model and BERT Embeddings, in: 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2021, pp. 285-289.

[4] A. Gao, From PGC to UGC to AIGC: Change of content paradigm, in: SHS Web of Conferences, EDP Sciences, 2024, pp. 03017.

[5] R. Bommasani, D.A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M.S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258, (2021).

[6] N. Prova, Detecting AI Generated Text Based on NLP and Machine Learning Approaches, arXiv preprint arXiv:2404.10032, (2024).

[7] S. Chakraborty, A.S. Bedi, S. Zhu, B. An, D. Manocha, F. Huang, On the possibilities of ai-generated text detection, arXiv preprint arXiv:2304.04736, (2023).

[8] K. Hayawi, S. Shahriar, S.S. Mathew, The imitation game: Detecting human and AI-generated texts in the era of ChatGPT and BARD, Journal of Information Science, (2024) 01655515241227531.

[9] Y. Xie, A. Rawal, Y. Cen, D. Zhao, S.K. Narang, S. Sushmita, MUGC: Machine Generated versus User Generated Content Detection, arXiv preprint arXiv:2403.19725, (2024).

[10] S. Yang, S. Yang, C. Tong, In-Depth Application of Artificial Intelligence-Generated Content AIGC Large Model in Higher Education, Adult and Higher Education, 5(19) (2023) 9-16.

[11] Y. Wang, Y. Pan, M. Yan, Z. Su, T.H. Luan, A survey on ChatGPT: AI-generated contents, challenges, and solutions, IEEE Open Journal of the Computer Society, (2023).

[12] E.G. Said, Arabic Chatbots Challenges and Solutions: A Systematic Literature Review, Iraqi Journal For Computer Science and Mathematics, 5(3) (2024) 128-169.

[13] K. Darwish, N. Habash, M. Abbas, H. Al-Khalifa, H.T. Al-Natsheh, H. Bouamor, K. Bouzoubaa, V. Cavalli-Sforza, S.R. El-Beltagy, W. El-Hajj, M. Jarrar, H. Mubarak, A panoramic survey of natural language processing in the Arab world, Commun. ACM, 64(4) (2021) 72–81.

[14] A.A. ElSabagh, S.S. Azab, H.A. Hefny, A comprehensive survey on Arabic text augmentation: approaches, challenges, and applications, Neural Computing and Applications, 37(10) (2025) 7015-7048.

[15] E.H. Almansor, A. Al-Ani, F.K. Hussain, Transferring informal text in arabic as low resource languages: State-of-the-art and future research directions, in: Complex, Intelligent, and Software Intensive Systems: Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019), Springer, 2020, pp. 176-187.

[16] Y. Saoudi, M.M. Gammoudi, Trends and challenges of Arabic Chatbots: Literature review, Jordanian Journal of Computers and Information Technology (JJCIT), 9(03) (2023).

[17] S. Feng, H. Wan, N. Wang, Z. Tan, M. Luo, Y. Tsvetkov, What Does the Bot Say? Opportunities and Risks of Large Language Models in Social Media Bot Detection, arXiv preprint arXiv:2402.00371, (2024).

[18] H. Alshammari, K. Elleithy, Toward Robust Arabic AI-Generated Text Detection: Tackling Diacritics Challenges, Information, 15(7) (2024) 419.

[19] B.S. Leite, Generative Artificial Intelligence in chemistry teaching: ChatGPT, Gemini, and Copilot's content responses, Journal of Applied Learning and Teaching, 7(2) (2024).

[20] Z. Lai, X. Zhang, S. Chen, Adaptive ensembles of fine-

tuned transformers for llm-generated text detection, arXiv preprint arXiv:2403.13335, (2024).

[21] A. Abdelali, H. Mubarak, S. Chowdhury, M. Hasanain, B. Mousi, S. Boughorbel, S. Abdaljalil, Y. El Kheir, D. Izham, F. Dalvi, Larabench: Benchmarking arabic ai with large language models, in: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 487-520.

[22] S. Alshahrani, N. Alshahrani, S. Dey, J. Matthews, Performance Implications of Using Unrepresentative Corpora in Arabic Natural Language Processing, in: Proceedings of ArabicNLP 2023, 2023, pp. 218-231.

[23] H. Chouikhi, M. Aloui, C.B. Hammou, G. Chaabane, H. Kchaou, C. Dhaouadi, GemmAr: Enhancing LLMs Through Arabic Instruction-Tuning, arXiv preprint arXiv:2407.02147, (2024).

[24] E. Haque, A Beginner's Guide to Large Language Models, Enamul Haque, 2024.

[25] Y. Bengio, R. Ducharme, P. Vincent, A neural probabilistic language model, Advances in neural information processing systems, 13 (2000).

[26] F. Alhayan, H. Himdi, Ensemble learning approach for distinguishing human and computer-generated Arabic reviews, PeerJ Computer Science, 10 (2024) e2345.

[27] N.S. Alghamdi, J.S. Alowibdi, Distinguishing Arabic GenAI-generated Tweets and Human Tweets utilizing Machine Learning, Engineering, Technology & Applied Science Research, 14(5) (2024) 16720-16726.

[28] K.S. Kalyan, A survey of GPT-3 family large language models including ChatGPT and GPT-4, Natural Language Processing Journal, (2023) 100048.

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

[30] H. Alshammari, A. El-Sayed, K. Elleithy, AI-Generated Text Detector for Arabic Language Using Encoder-Based Transformer Architecture, Big Data and Cognitive Computing, 8(3) (2024) 32.

[31] H. Almerekhi, 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, Springer, 2015, pp. 123-134.

[32] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, Advances in Neural Information Processing Systems, 36 (2024).

[33] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI blog, 1(8) (2019) 9.

[34] H. Wang, X. Luo, W. Wang, X. Yan, Bot or human? detecting chatgpt imposters with a single question, arXiv preprint arXiv:2305.06424, (2023).

[35] B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, Language models are few-shot learners, arXiv preprint arXiv:2005.14165, 1 (2020).

[36] S. Kadam, V. Vaidya, Review and analysis of zero, one and few shot learning approaches, in: Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 1, Springer, 2020, pp. 100-112.

[37] Y. Song, T. Wang, P. Cai, S.K. Mondal, J.P. Sahoo, A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities, ACM Computing Surveys, 55(13s) (2023) 1-40.

[38] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, arXiv preprint arXiv:1801.06146, (2018).

[39] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, N. Smith, Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, arXiv preprint arXiv:2002.06305, (2020).

[40] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685, (2021).

[41] A.Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D.S. Chaplot, D.d.l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, Mistral 7B, arXiv preprint arXiv:2310.06825, (2023).

[42] amitsangani, meta-llama/Llama-3.1-8B-Instruct, (2024).

[43] M.S.B.a.Y.A.a.N.A.A.a.N.M.A.a.H.A.A.a.S.A.a.F.A .M.a.S.Z.A. and, ALLaM: Large Language Models for Arabic and English, (2025).

[44] S.I. Hassan, L. Elrefaei, M.S. Andraws, Arabic Tweets Spam Detection Based on Various Supervised Machine Learning and Deep Learning Classifiers, MSA Engineering Journal, 2(2) (2023) 1099-1119.

[45] K. Gaanoun, I. Benelallam, Arabic dialect identification: An Arabic-BERT model with data augmentation and ensembling strategy, in: Proceedings of the Fifth Arabic Natural Language Processing Workshop, 2020, pp. 275-281.

[46] D. Refai, S. Abu-Soud, M.J. Abdel-Rahman, Data augmentation using transformers and similarity measures for improving arabic text classification, IEEE Access, 11 (2023) 132516-132531.

[47] B. Mousi, N. Durrani, F. Ahmad, M.A. Hasan, M. Hasanain, T. Kabbani, F. Dalvi, S.A. Chowdhury, F. Alam, AraDiCE: Benchmarks for Dialectal and Cultural Capabilities in LLMs, arXiv preprint arXiv:2409.11404, (2024).

[48] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, arXiv preprint arXiv:2402.06196, (2024).

HOW TO CITE THIS ARTICLE
O. Al Minshidawi, A. Vahabie, Classifying AI-Generated Text in Low-Resource Languages like Arabic, AUT J. Model. Simul., 57(1) (2025) 113-124.

DOI: 10.22060/miscj.2025.24060.5408