# AIRABIC: Arabic Dataset for Performance Evaluation of AI Detectors

Hamed Alshammari
*Department of Computer Science and Engineering*
*University of Bridgeport*
Bridgeport, CT 06604, USA
halsh@my.bridgeport.edu

Ahmed El-Sayed
*Department of Electrical and Computer Engineering*
*University of Bridgeport*
Bridgeport, CT 06604, USA
aelsayed@bridgeport.edu

***Abstract*— In the rapid expansion of Large Language Models (LLMs), such as ChatGPT, AI-generated text detection models have made substantial advancements, marking meaningful progress in several research and industrial applications. However, the performance of these models in the Semitic languages' context, especially regarding diacritic usage, continues to be inadequately explored. One of these Semitic languages that is still commonly used is Arabic language. To study the performance of the recent AI-generated text detection models on the Arabic language, this paper introduces the AIRABIC dataset, a combination of 1000 examples encompassing 500 human-written passages from 41 unique sources and an equal number of AI-generated texts from the OpenAI's Generative Pre-trained Transformer version 3.5 (GPT-3.5 Turbo ChatGPT). This study focuses on the performance evaluation of two prominent AI-generated text detectors, GPTZero and OpenAI's Text Classifier. Our findings reveal that GPTZero achieves an overall accuracy rate of 62.6%, while the OpenAI Text Classifier exhibits a 50% rate of biased categorizations when analyzing human-written text. Further analysis shows the design gaps of these detectors, especially for identifying human-written Arabic text, mainly when diacritics are involved. The detection accuracy for human-written texts with diacritics is as low as 30% for GPTZero and 0% for the OpenAI Text Classifier. Results also show the potential of diacritics to reduce the detectors' accuracy and the need to handle them in the detectors' design process.**

***Keywords—AI Detectors, GPTZero, OpenAI Text Classifier ChatGPT, GPT, Diacritized Arabic texts, Diacritics, Arabic NLP, ANLP, AIRABIC, AI text.***

## I. Introduction

The field of artificial intelligence (AI) has improved dramatically in recent years. Throughout its history, it has expanded significantly in many areas, especially in Natural Language Processing (NLP). Advancements in Large Language Models (LLMs) have adapted to generate high-quality text that has a broad spectrum of applications. For example, the ChatGPT [1] model from OpenAI can generate texts that mimic human-written text. Its capability has increased with its improved variant based on GPT-4 architecture. Nonetheless, these technological improvements carry the unintended consequence of enabling unethical applications, including plagiarism and creating fake textual content. As a result, AI-generated text has been the subject of many academic research investigations. Some of these studies arrived before the ChatGPT release, as shown in [2-5]. Following the advent of ChatGPT, scholarly interest in this domain has notably increased, as illustrated by [6-8].

In response to the potential misuse of the AI generative models, some recently proposed algorithms identify AI-generated text, such as [9, 10]. Additionally, there exist commercial platforms offering ready-to-use solutions for this purpose, such as Turnitin [11], GPTZero [12], and OpenAI Text Classifier [13]. Nonetheless, studies like [6] underscore the need for more reliable techniques to prevent the unethical usage of LLMs and critically assess the efficacy of systems that claim to identify AI-generated text. Nevertheless, these studies focus primarily on the English language. Despite their imperfect results, these AI detectors can distinguish between human-written text and AI-generated text with some confidence. To the best of our knowledge, those detectors have not been tested on Semitic languages to show the performance with non-Latin-based language.

The Arabic language serves as a representative example of Semitic languages and is generally categorized into three distinct forms: Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialects Arabic (DA). CA is a pure form of the original Arabic language used in religious texts and old historical and poetry Arabic books. MSA is a derivative of CA, employed in formal contexts such as news broadcasts and educational settings. Finally, DA is the colloquial Arabic dialect used in everyday interactions with family and friends [14], and it changes by geographic place and origin. Based on that, addressing the performance analysis of the AI detectors in Arabic will focus mainly on CA and MSA.

### A. Problem Identification

The Arabic language presents many orthographic challenges for the NLP systems because of the similarity between letters and sounds, which has been discussed in [14-16]. Moreover, two unique characteristics and the NLP challenges of the Arabic language are the writing from right to left and the use of optional diacritical marks. Utilizing these diacritical marks helps to articulate short vowels and consonantal sounds [17].

The diacritized Arabic texts present formidable challenges for AI-based detection mechanisms. Nevertheless, the presence of diacritical marks should not be misconstrued as an indication of text generated by an AI for several reasons. First, diacritical marks are essential in distinguishing some words from others. For example, the word 'كَتَبَ' means 'he wrote,' but if the diacritics have been modified as in this word 'كُتُبْ', its meaning will change to 'books.' This variation is not only limited to semantics but can also be extended to phonetic differences. It has been highlighted in [17] that the primary source of ambiguity in Arabic NLP systems is the absence of diacritic marks. Moreover, most Arabic religious books have diacritics and have been written in ancient times. Therefore, their existence in Arabic texts should not be used as a sole criterion to classify it as AI-generated content.

Motivated by the need to evaluate the performance of existing AI detectors for Arabic text, this paper introduces the AIRABIC dataset. This dataset comprises 1,000 samples, with

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00127

TABLE I. COMPARATIVE OVERVIEW OF HUMAN-WRITTEN AND AI-GENERATED TEXTS IN THE DATASET

| **Human-written Texts** | | | | | **Total Sample** | **Character Count** | | **AI-generated Texts** | | **Total Sample** | **Character Count** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***Examples count*** | | ***Sources*** | | ***Diacritized texts*** | 250 | ***Max*** | 3372 | ***Number of Topics*** | ***Number of prompts entered ChatGPT*** | 250 | ***Max*** | 3095 |
| MSA | CA | Books | News | ***Non-diacritized texts*** | 250 | ***Min*** | 294 | 50 | ***Diacritic applications without prompt*** | 250 | ***Min*** | 386 |
| 263 | 237 | 40 | 1 | ***Total*** | 500 | ***Avg*** | 1122 | | ***Total*** | 500 | ***Avg*** | 1804 |

500 sourced from human-authored books, manuscripts, and custom Arabic passages, and another 500 from AI-generated text, specifically the ChatGPT 3.5 model. Each human-written sample is included twice: once with diacritical marks and once without. Similarly, of the 500 AI-generated samples, 250 were originally without diacritics and were later processed to include them. These selections were meticulously curated from 41 different human-written sources to ensure a wide range of topics and subjects. Thus, the dataset offers a balanced representation of diacritic and non-diacritic texts from human and AI-generated sources. Two prominent AI text detection models, GPTZero and OpenAI Text Classifier, were employed to assess the dataset according to a predefined protocol.

The contributions of this paper can be summarized as follows:

1) Introducing a novel AIRABIC[1] benchmark dataset that includes human-written samples from CA and MSA sources and texts generated by the ChatGPT model. This unique dataset contains texts with and without diacritics, offering a broad spectrum of linguistic variations.

2) Testing the AIRABIC dataset using two of the most prominent AI-generated text detectors, GPTZero and OpenAI's Text Classifier, and comparing their performance and confusion matrices.

3) Highlighting the drawbacks and disadvantages of the existing AI-generated text classifiers on Arabic language texts.

4) Discussing and analyzing the performance of the commercial detection models and illustrating the causes for the tested classifiers' performance.

The remainder of the paper is organized as follows: The second section offers an in-depth description of the AIRABIC dataset, including its statistical characteristics. The third section details the methodology employed in data collection and testing protocols. The fourth section presents the results of evaluating the dataset using GPTZero and OpenAI Text Classifier. The paper concludes with a comprehensive discussion and conclusions section.

## II. DATASET

### A. Dataset Description

Since the GPTZero and OpenAI Text Classifier models allow a specified range of text input in terms of the minimum character limit, e.g., the minimum number of input characters per passage for the GPTZero and OpenAI Text Classifier starts with 250 and 1000 characters, respectively, the proposed dataset must satisfy these minimum requirements. On the other hand, GPTZero has a maximum character limit of 5000 for free use, while OpenAI Text Classifier does not explicitly stipulate an upper limit. Therefore, given these specific parameters, the constructed dataset samples must meet these requirements and also encompass a diverse group of texts and topics.

### B. Dataset Characteristics

*1) Source Diversity:* To guarantee the objectivity and unbiased behavior of the proposed dataset, the human-written passages are selected from different sources, including books and news articles. These selections include 40 books from various topics and ages, with no mixing or overlapping of the passages to retain the uniqueness of each passage. To further enrich this assortment, the human texts also include content from the Aljazeera news website, specifically articles written between the years 2014 and 2016. This wide spectrum of sources ensures a comprehensive and robust dataset free from any AI-generated text and guarantees the objectivity of the human-written texts.

*2) Sample Size:* The constructed dataset with human-written texts is built to include various numbers of characters within the detectors' minimum and maximum limits, as stated in Table I. The selection of the number of characters was primarily determined by the specifics and the meaningfulness of the chosen passage rather than a rigidly predetermined design choice. Therefore, in order to fulfill the prerequisite conditions stipulated by the OpenAI Text Classifier, any sample that includes less than 1000 characters was padded with spaces. However, the AI-generated texts, on the other hand, are principally determined by the generated outputs of the ChatGPT model. For consistency and upholding a uniform standard, some factors guided the model to compose a meaningful exposition on a specific topic to ensure the output is printed within a reasonable character limit.

3) *Text Variations:* The dataset is designed to include a variety of passage formats. That encompasses one-paragraph, two paragraphs composition, texts with bullet points, passages with an in-text citation, and more for both human-written and AI-generated texts. This diversity of text structure is intentionally selected to scrutinize the adaptability of AI detectors in these various contexts. Although all human-written passages could theoretically be drawn from a single book, this approach might limit the generalization and the robustness of the dataset. Therefore, the dataset incorporates passages from various carefully selected sources to circumvent potential detection bias. This measure ensures a robustness and comprehensive evaluation of the AI detectors' capabilities across varied text types and structures.

[1] https://github.com/Hamed1Hamed/AIRABIC

TABLE II. An Illustrative Example Obtained from Human-written Text with/without Diacritics Against GPTZero Detector

| Diacritized Text Sample | | Non-diacritized Text Sample | | Translation |
|---|---|---|---|---|
| لا يُذْكَرُ اسْمُ الْوَرْدِ إِلَّا اسْتَحْضَرَت الْأَذْهَانُ مَدِينَةَ الطَّائِفِ الَّتِي يُعْتَبَرُ عِطْرُهَا مِنْ أَغْلَى الْعُطُورِ.<br>يَرْتَبِطُ الْوَرْدُ بَيْنَ النَّاسِ بِالصَّبَّاحَاتِ وَالْمَسَاءَاتِ وَالْهَدَايَا وَالزِّيَارَاتِ أَمَّا عِنْدَ أَهْلِ الطَّائِفِ فَهُوَ ارْتِبَاطٌ رُوحِيٌّ وَتَارِيخِيٌّ وَعِشْقٌ مُتَبَادَلٌ، يَسْقُونَهُ وَيَعْرِفُونَهُ كَمَا يَعْرِفُهُمْ مُنْذُ الْقَرْنِ التَّاسِعِ الْهِجْرِيِّ.<br>لا يُذْكَرُ اسْمُ الْوَرْدِ إِلَّا اسْتَحْضَرَت الْأَذْهَانُ مَدِينَةَ الطَّائِفِ (88 كِيلُومِتْرًا شَرْقَ مَكَّةَ الْمُكَرَّمَةَ) وَرَائِحَةَ عِطْرِ وَرْدِها الْفَوَّاحَةَ، الَّذِي يَعْتَبِرُهُ صُنَّاعُ الْعُطُورِ مِنْ أَغْلَى الْعُطُورِ.<br>وَأَطْيَبُ الْوَرْدِ الطَّائِفِيِّ مَا كَانَ فِي جِبَالِ الْهَدَا وَالشِّفَاءِ، وَأَعْبَقُهَا مَا يُرْوَى مِنْ مِيَاهِ الْمَطَرِ وَلَمْ يُؤْذِهِ قَرٌّ أَوْ حَرٌّ، يَطِيبُ الْمُزَارِعُ مَعَهُ فَيَزْدَادُ طِيبًا.<br>وَفِي كُلِّ عَامٍ مِنْ مَوْسِمِ الْوَرْدِ الطَّائِفِيِّ (نِهَايَةَ مَارِسَ/آذَارَ وَأَوَائِلَ أَبْرِيلَ/نِيسَانَ) تُرْوَى قِصَّةُ حُبٍّ بَيْنَهُ وَبَيْنَ أَهْلِهِ عَلَى قِمَمِ وَسُفُوحِ جِبَالِ الطَّائِفِ. | | لا يذكر اسم الورد إلا استحضرت الأذهان مدينة الطائف التي يعتبر عطرها من أغلى العطور.<br>يرتبط الورد بين الناس بالصباحات والمساءات والهدايا والزيارات أما عند أهل الطائف فهو ارتباط روحي وتاريخي وعشق متبادل، يسقونه ويعرفونه كما يعرفهم منذ القرن التاسع الهجري.<br>لا يذكر اسم الورد إلا استحضرت الأذهان مدينة الطائف (88 كيلومترا شرق مكة المكرمة) ورائحة عطر وردها الفواحة، الذي يعتبره صناع العطور من أغلى العطور.<br>وأطيب الورد الطائفي ما كان في جبال الهدا والشفاء، وأعبقها ما يروى من مياه المطر ولم يؤذه قر أو حر، يطيب المزارع معه فيزداد طيبا.<br>وفي كل عام من موسم الورد الطائفي (نهاية مارس/آذار وأوائل أبريل/نيسان) تروى قصة حب بينه وبين أهله على قمم وسفوح جبال الطائف. | | The rose name would not be mentioned without bringing to mind the city of Taif, that its perfume is considered as one of the most expensive.<br>Roses are associated with people for mornings and evenings, gifts, and visits. However, for the people of Taif, roses are a spiritual and historical connection and love. They have known it since the ninth century AH.<br>The name of the rose is inevitably linked to the city of Taif, (located 88 kilometers east of Mecca). Perfume makers consider the fragrant rose of Taif to be one of the most expensive perfumes.<br>The best Taif roses are found in the mountains of Al-Hada and Al-Shifa. The most fragrant of these roses are those irrigated by rainwater, and those are not harmed by either cold or heat.<br>Every year during the Taif rose season (end of March and early April), a love story is narrated between roses and Taif people on the peaks and slopes of the Taif mountains. |
| ***Number of characters*** | 1073 | ***Number of characters*** | 653 | |
| ***Results*** | Detected as AI-generated texts | ***Results*** | Detected as Human-written texts | |

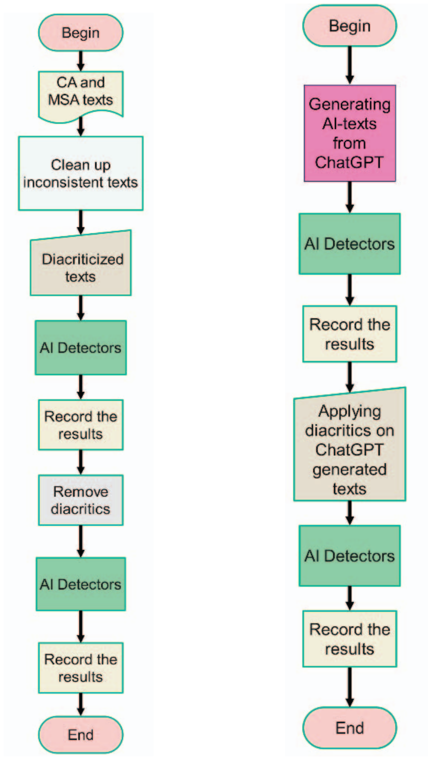
## III. Methodology

### A. Data Collection Protocol

The initial step toward building the human-written text corpus for the proposed dataset involves examining a variety of books to identify well-written and informative passages suitable for the dataset's objectives. The majority of the books used in this paper were sourced from the Shamila Library [18], a readily accessible repository offering a diverse range of Arabic book categories. It is important to note that paragraphs from books are the sole source used in this data collection protocol. Another source with a collection of diacritized textbooks is used in this paper, called the Tashkeela dataset [19], which builds upon the selection from the Shamila Library. However, it should be pointed out that the Tashkeela dataset contains some inconsistent text structures that require further refinement steps to form coherent paragraphs.

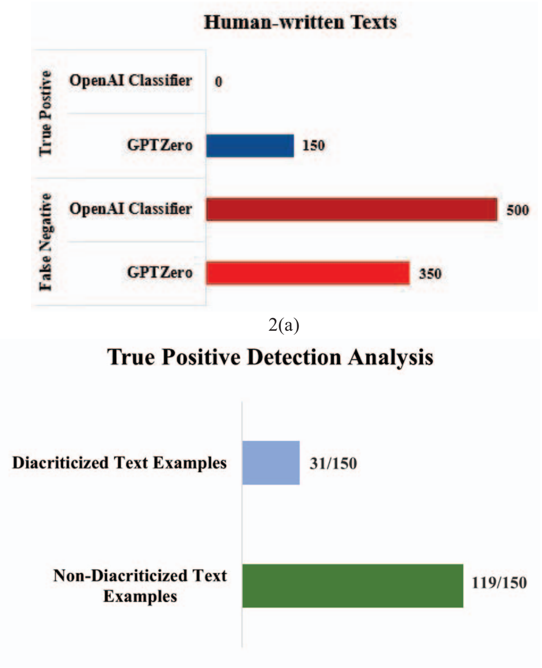
### B. Data Processing

Various NLP processing techniques are tailored for Arabic language text, including the addition and removal of diacritical marks. ARBML [20] is a model equipped with a wide range of functionalities to meet various needs related to the Arabic language. One of its functions is providing diacritic marks for non- diacritized text. It has been built upon Shakkala [21], a model for creating diacritics using a deep neural network. However, a notable limitation of the ARBML model is its restricted input size capacity, which has a maximum limit of 315 characters, making it less suitable for the collected data, since the input, in most cases, exceeds the limit. Another set of tools that does the same function is CAMeL tools [17]. CAMeL tools are comprehensive tools that deal with the Arabic language, including adding and removing diacritics with no input limit. Because of this, the CAMeL tools have been selected in this paper to add or remove diacritics from the collected dataset samples.



(a) (b)

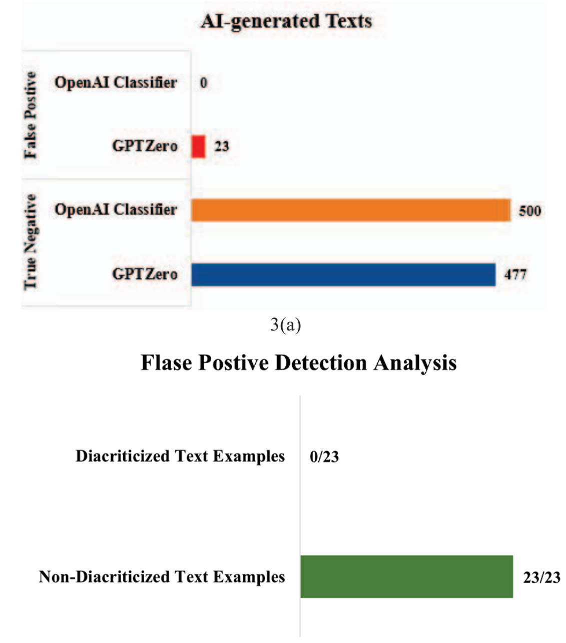
Fig. 1: Testing Process: (a) Testing Process of AI Detectors Against Human-Written Passages. (b) Testing Process of AI Detectors Against AI-Generated Text



2(a)

2(b)

Fig. 2: (a) Performance of GPTZero and OpenAI Classifiers on Human-Written Texts dataset. (b) GPTZero Detector: Analysis of Correct Non-diacritized and Diacritized Text Examples.



3(a)

3(b)

Fig 3: (a) Performance of GPTZero and OpenAI Classifiers on AI-generated Texts dataset. (b) GPTZero Detector: Analysis of false Non-diacritized and Diacritized Text Examples.

### *C. Testing Protocol*

A preliminary examination of AI detectors' interaction with human-written passages shows a recurrent tendency of these detectors to categorize Arabic language texts as AI-generated. Based on this observation, an experiment incorporating two distinct steps has been developed, as shown in Fig. 1. These steps are discussed in the following points.

*1) Testing Human-written Samples:* The AI detectors, namely GPTZero and OpenAI Text Classifier, are initially tested on text from MSA and CA books' passages that were embellished with diacritics. The output of these AI detectors has been considered a binary classification problem. In the case that the AI detector determines the text to be human-written entirety or even partially, the output will be marked and classified as a correct detection (True Positive). On the other hand, if the detector indicates that an AI entirely or partially has produced the text, the outcome would fall under the category of false detection (False Negative). More details about this classification are in the discussion section. Afterward, the same samples without diacritics are applied as input to the detector, maintaining each paragraph's syntactic and semantic integrity, and re-engaging the detectors. This way of testing helps identify problems with the detector under test. It is important to clarify that at this stage, the evaluation is not designed to test AI detectors on AI-generated text that a human has subsequently paraphrased or mixed human texts with AI-generated ones. Instead, this investigation is intended to understand the performance of AI detectors with the problems associated with diacritic Arabic texts. For that reason, this criterion directed the choice toward books written in MSA and CA. Fig. 1(a) depicts the process executed during this examination stage. Table II compares a sample input text showcasing its appearance before and after the removal of diacritics. Observable results of removing diacritics include reducing text character count since each diacritic is treated as a separate character.

*2) Testing AI-generated Samples:* For this study, text samples were collected from the ChatGPT 3.5 model across 50 different topics. These samples were then used as input for the GPTZero and OpenAI Text Classifier. To ensure consistency with the human-written samples, diacritical marks were added to these AI-generated samples. A second round of testing was subsequently conducted. A visual representation of the testing procedure can be viewed in Fig. 1(b).

## IV. Results

To achieve the main goal of this paper, which is to examine the performance of the two most popular AI-text detectors, all collected samples, both human-written texts and AI-generated, were subjected to tests using both GPTZero and the OpenAI Text Classifier, in accordance with the described protocol. As indicated in Table III, the GPTZero classifier achieved a total accuracy rate of 62.7%, outperforming the OpenAI Text Classifier, which registered a total accuracy rate of 50%. The Receiver Operating Characteristics (ROC) Curves for both classifiers are depicted in Fig. 4. The results yield several key insights, which are analyzed in the subsequent sections.

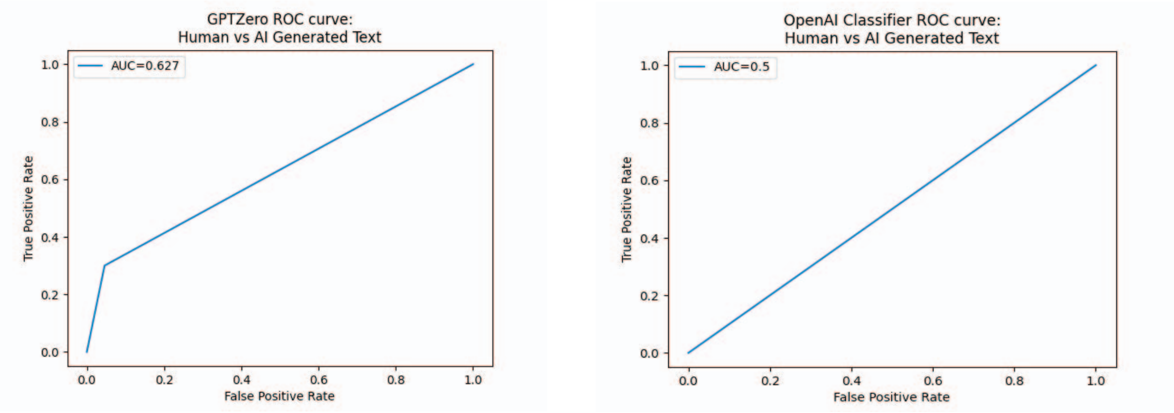
TABLE III. EVALUATION METRICS FOR GPTZERO AND OPENAI CLASSIFIER ON HUMAN-WRITTEN AND AI-GENERATED TEXTS

(A)

| GPTZero | Predicted: Human-written | Predicted: AI-generated | Performance Metrics | Value |
|---|---|---|---|---|
| **Actual: Human-written** | 150 (TP) | 350 (FN) | Sensitivity | 30% |
| | | | Specificity | 95% |
| **Actual: AI-generated** | 23 (FP) | 477 (TN) | Precision | 86.7% |
| | | | Accuracy | **62.7%** |
| | | | F1-Score | 44.5% |

(B)

| OpenAI Classifier | Predicted: Human-written | Predicted: AI-generated | Performance Metrics | Value |
|---|---|---|---|---|
| **Actual: Human-written** | 0 (TP) | 500 (FN) | Sensitivity | 0% |
| | | | Specificity | 100% |
| **Actual: AI-generated** | 0 (FP) | 500 (TN) | Precision | 0% |
| | | | Accuracy | **50%** |
| | | | F1-Score | 0% |



(a)

(b)

Fig. 4. ROC Curves of Human vs AI-Generated Text Detection; (a) Performance of GPTZero and (b) Performance of OpenAI Classifier.

### A. *Human-written Texts:*

For the 500 distinct instances of human-written texts, the OpenAI Text Classifier has consistently misclassified human-written texts as AI-generated text. In the ideal scenario, the classifier's expected behavior should be 'unlikely' or 'very unlikely' AI-generated to be considered as human-written texts correctly. However, the classifier processed 499 out of the 500 examples with the 'possibly' or 'likely' AI-generated labels, both of which officially mean AI-generated texts according to OpenAI's guidelines [13]. Only one instance was classified as 'unclear', which in the context of our experiment has been considered an incorrect binary classification. Moreover, regardless of the number of diacritical marks present in Arabic text, human-written input cannot be accurately classified. This was uniformly observed across all 500 examples, as illustrated in Fig. 2(a). As a result, this classifier is not currently capable of processing the Arabic language effectively at its current stage of development due to its bias towards AI text.

The performance of GPTZero in detecting human-written texts requires in-depth analysis. The results indicate that the GPTZero classifier can easily misclassify when dealing with diacritized texts. While it performs modestly in identifying human-written Arabic texts, it significantly outperforms the OpenAI Text Classifier. It has accurately detected 31 out of 250 instances of diacritized Arabic text from collected samples, as shown in Fig. 2(b). It is important to note that, once the model successfully identifies diacritized texts as human-written, it retained this correct classification even after diacritic removal in all tested samples, whereas the reverse was not true. The model's recognition capability has been improved upon removing diacritics from input samples, with correctly identifying 119 out of 250 non-diacritics examples, as shown in Fig. 2(b). As a result, the total successful detection of the human-written samples is 150 out of 500, as shown in Table III.

### B. *AI-generated Texts:*

Since the OpenAI Text Classifier exhibits bias in the Arabic text, it classifies nearly all samples of AI-generated texts as such, as demonstrated in Fig. 3(a). In detail, out of the 500 AI-generated text samples, the classifier identified 492 samples as 'likely AI-generated' and eight samples as 'possibly AI-generated.' GPTZero, on the other hand, has shown a specificity of 95% due to misclassifying 23 non-diacritized text examples as human-written texts. Notably, adding diacritics to these 23 samples corrected the initial misclassification, and they were subsequently accurately identified as AI-generated texts, as depicted in Fig. 3(b). An in-depth evaluation of the two AI detectors' performance, including key metrics such as sensitivity, specificity, precision, accuracy, and F1-score, is presented in Table III.

## V. DISCUSSION

The results of this study indicate that both tested AI detectors are still at an early stage of their ability to recognize Arabic human-written vs. AI-generated texts[2]. Both detectors

[2] The first stage of testing the two detectors started on April 30th, 2023. The testing was revised three times to verify the results. Revising the dataset ends on June 1st, 2023.

vary in terms of detecting human-written texts, as shown in Table III and Fig. 4. On the one hand, the OpenAI Text Classifier exhibits a discernible bias towards Arabic-language text regardless of the size or type of the input characters, with or without diacritics. Consequently, this results in an inability of this classifier to distinguish between human-written vs. AI-generated text.

On the other hand, the GPTZero performs better for shorter Arabic text passages or smaller size input samples. However, its detection patterns remain closer to a random chance selection. For example, a paragraph with 641 characters length is correctly identified as human-written, yet merely adding a single character to become with 642 characters input results in a misclassification as an AI-generated text. However, it occasionally exhibits the capacity to accurately label text passages exceeding 1900 characters as human-written text. Intriguingly, the presence of a period ('.') or a new line can sway GPTZero's decision, leading to the misclassification of human text as AI-generated.

Further analysis of the proposed dataset can show that the readability and comprehension of the Arabic language are substantially reliant on diacritic marks, implying that their inclusion or exclusion should not be considered as a main factor in distinguishing between human and AI-generated text. Apart from the reasons mentioned in the introduction, the word embedding for a word carrying diacritics should be completely different (in some cases) from the one without because of the meaning variant. That has been explained in a recently released embedding model called icrOSim, which was introduced in [22].

Additionally, the AI-generated text by one of the LLMs, such as ChatGPT, does not generate diacritized texts, but diacritics can be added using a later automated processing technique. Therefore, diacritized input should not be considered a dominant factor in identifying the AI-generated text. Instead, the focus should shift to the structure and style of the written text itself. For instance, from the samples obtained from ChatGPT, the quality of generated texts can be detected by a glance that the writing style being used reveals specific characteristic patterns in the output, such as a tendency to compose short sentences per paragraph (around three sentences before initiating a new paragraph). Also, the quality of the Arabic-generated text by ChatGPT on abroad topics does not quite match its proficiency in generating English text. However, the choice of using ChatGPT as the source for producing Arabic AI-generated texts has been primarily driven by its unique capability among other LLMs to produce Arabic text, a function which, for instance, BARD [23] cannot do.

One plausible solution to improve the misclassification of the existing detection models on human-written texts, especially the diacritized text, is constructing an additional input layer that can work on filtering the input from any diacritics. Fig. 2(b) shows, for example, that once the diacritics are removed, the number of samples identified as human-written texts exceeds those of diacritic ones. While this method may not offer a comprehensive solution to enhance the proficiency of AI detectors in Arabic language recognition, it could serve as a preliminary step toward more accurate AI-generated text detection. An additional approach involves the development of a dedicated classifier specifically tailored to recognize Arabic text, with a refined ability to distinguish between human-written and AI-generated content. Further future research will focus primarily on this proposed approach.

## VI. Conclusion

This paper has introduced a new benchmark Arabic dataset that contains texts obtained from human-written and AI-generated texts. The efficacy and performance of two AI text detectors have been explored and evaluated, specifically GPTZero, and OpenAI's Text Classifier, on the introduced AIRABIC dataset to identify human-written Arabic passages as well as detect AI-generated texts. Results have revealed that these AI detectors exhibit significant room for improvement, particularly concerning the detection of diacriticized Arabic texts. The OpenAI Classifier's results have shown similar performance to a random chance with a complete bias towards the AI-generated text. Moreover, it has been shown that the GPTZero demonstrated better performance with non-diacritized human written texts, implying that the presence of diacritics may serve as a source of distraction or confusion for the classifier. This observation forms a key consideration for the development of future detection models. In light of these findings, we proposed potential solutions to augment the performance of these detectors. These include the construction of an input layer for diacritics processing and the development of a dedicated classifier specifically tailored to Arabic text that should be trained on diacritics and non-diacritics texts. These proposed interventions could serve as initial steps toward enhancing the capability of AI detectors in the Arabic domain. As future work, more research will take place and focus on implementing and evaluating these proposed solutions to enrich the field of AI text detection in Arabic and Sematic languages.

## VII. References

[1] OpenAI, "Optimizing language models for dialogue," *OpenAI,* February 1 2022. [Online]. Available: https://online-chatgpt.com/.

[2] G. Jawahar, M. Abdul-Mageed, and L. V. Lakshmanan, "Automatic detection of machine generated text: A critical survey," *arXiv preprint arXiv:2011.01314,* 2020.

[3] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. A. Ranzato, and A. Szlam, "Real or fake? learning to discriminate machine from human generated text," *arXiv preprint arXiv:1906.03351,* 2019.

[4] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi, "TweepFake: About detecting deepfake tweets," *Plos one,* vol. 16, no. 5, p. e0251415, 2021.

[5] W. Antoun, F. Baly, and H. Hajj, "AraGPT2: Pre-trained transformer for Arabic language generation," *arXiv preprint arXiv:2012.15520,* 2020.

[6] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, "Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense," *arXiv preprint arXiv:2303.13408,* 2023.

[7] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, "Can AI-Generated Text be Reliably Detected?," *arXiv preprint arXiv:2303.11156,* 2023.

[8] H. Ibrahim *et al.*, "Perception, performance, and detectability of conversational artificial intelligence across 32 university courses," *Scientific Reports,* vol. 13, no. 1, p. 12187, 2023.

[9] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A watermark for large language models," *arXiv preprint arXiv:2301.10226,* 2023.

[10] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "Detectgpt: Zero-shot machine-generated text detection using probability curvature," *arXiv preprint arXiv:2301.11305,* 2023.

[11] Turnitin. https://help.turnitin.com/ (accessed June 1, 2023).

[12] GPTZero. https://gptzero.me/ (accessed June 1, 2023).

[13] OpenAI. https://beta.openai.com/ai-text-classifier (accessed June 1, 2023).

[14] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," *ACM Transactions on Asian Language Information Processing (TALIP),* vol. 8, no. 4, pp. 1-22, 2009.

[15] K. Darwish *et al.*, "A panoramic survey of natural language processing in the Arab world," *Communications of the ACM,* vol. 64, no. 4, pp. 72-81, 2021.

[16] N. Y. Habash, "Introduction to Arabic natural language processing," *Synthesis lectures on human language technologies,* vol. 3, no. 1, pp. 1-187, 2010.

[17] O. Obeid *et al.*, "CAMeL tools: An open source python toolkit for Arabic natural language processing," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 7022-7032.

[18] Shamela. https://shamela.ws/ (accessed May 3, 2023).

[19] T. Zerrouki and A. Balla, "Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems," *Data in brief,* vol. 11, pp. 147-151, 2017.

[20] Z. Alyafeai and M. Al-Shaibani, "ARBML: democritizing arabic natural language processing tools," in *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, 2020, pp. 8-13.

[21] A. Fadel, I. Tuffaha, and M. Al-Ayyoub, "Arabic text diacritization using deep neural networks," in *2019 2nd international conference on computer applications & information security (ICCAIS)*, 2019: IEEE, pp. 1-7.

[22] M. Abbache, A. Abbache, J. Xu, F. Meziane, and X. Wen, "The Impact of Arabic Diacritization on Word Embeddings," *ACM Transactions on Asian and Low-Resource Language Information Processing,* 2023.

[23] G. A. Bard. "BARD." Bard, Google AI. https://bard.google.com/ (accessed June 1, 2023).