Evaluation of Large Language Models on Arabic Punctuation Prediction

Asma Al Wazrah, Afrah Altamimi, Hawra Aljasim, Waad Alshammari, Rawan Al-Matham, Omar Elnashar, Mohamed Amin and Abdulrahman AlOsaimy
{aalwazrah, a.altamimi, haljasim, walshammari, ralmatham, kelnashar, mamin, aalosaimy}@ksaa.gov.sa
King Salman Global Academy for Arabic Language (KSAA)

Abstract

The linguistic inclusivity of Large Language Models (LLMs) such as ChatGPT, Gemni, JAIS, and AceGPT has not been sufficiently explored, particularly in their handling of low-resource languages like Arabic compared to English. While these models have shown impressive performance across various tasks, their effectiveness in Arabic remains under-examined. Punctuation, critical for sentence structure and comprehension in tasks like speech analysis, synthesis, and machine translation, requires precise prediction. This paper assesses seven LLMs: GPT4-o, Gemni1.5, JAIS, AceGPT, SILMA, ALLaM, and CommandR+ for Arabic punctuation prediction. Additionally, the performance of fine-tuned AraBERT is compared with these models in zero-shot and few-shot settings using a proposed Arabic punctuation prediction corpus of 10,046 data points. The experiments demonstrate that while AraBERT performs well for specific punctuation marks, LLMs show significant promise in zero-shot learning, with further improvements in few-shot scenarios. These findings highlight the potential of LLMs to enhance the automation and accuracy of Arabic text processing.

1 Introduction

Punctuation prediction remains a fundamental yet challenging aspect of natural language processing (NLP), particularly in enhancing the readability and understanding of text derived from spoken language inputs. In addition, this task is especially critical in the post-processing step of automatic speech recognition systems, where achieving high accuracy remains a significant challenge. This task plays a vital role in the coherent transformation of spoken language into written form, which is essential for effective communication and documentation. While several studies have explored punctuation prediction tasks, no study has examined the effectiveness of Large Language Models (LLMs) in predicting punctuation for Arabic texts.

Our research aims to evaluate the capabilities of LLMs in punctuation prediction task. In this study, we conduct a comprehensive evaluation of a fine-tuned AraBERT model alongside seven LLM-based models: GPT4-o, Gemni1.5, JAIS, AceGPT, SILMA, ALLaM, and CommandR+ across six punctuation marks. These models have been selected for their potential in handling the nuanced demands of Arabic natural language understanding and generation, making them ideal candidates for this investigation. The LLMs selection criteria included their pretraining focus, such as JAIS and AceGPT, which are specifically designed for Arabic and bilingual tasks, and their availability, with both open-source models such as JAIS and SILMA and closed-source models such as GPT4-o and Gemni1.5 included. The models also represent a mix of general-purpose systems, such as CommandR+, and those specialized for Arabic morphology and syntax, such as ALLaM.

We employ these LLMs in both zero-shot and few-shot learning scenarios to assess their performance. This dual approach allows us to explore not only the inherent capabilities of these models when presented with limited prior training on punctuation tasks but also their adaptability in learning from a minimal set of examples. Through our experiments, we aim to provide a detailed analysis of how each model handles the complexity of punctuation prediction and to identify the

Processings of the 1st Workshop on NLP for Languages Using Arabic Script (AbjadNLP), pages 144–154

strengths and limitations of each approach. This study hopes to contribute valuable insights into the potential of LLMs to improve punctuation prediction tasks in NLP, thereby enhancing the accuracy and efficiency of converting spoken language into punctuated written text.

The rest of the papers is presented as follows: section 2 presents the background of the used tools and methods. Section 3 presents the related works. Section 4 shows the methodology including the dataset and preparation, in addition to the model architecture. Section 5 discusses the experimental results, along with the error analysis of the testing data results. Finally, Section 6 concludes the paper and presents the future directions.

2 Background

This section reviews LLMs for Arabic NLP, Section 2.1 covers AraBERT model. Section 2.2 presents closed-source models such as GPT-4o and Gemini 1.5, while Section 2.3 discusses open-source models such as JAIS-13b and AceGPT.

2.1 AraBERT

AraBERT v0.2 (Antoun et al., 2020) is a pre-trained model for processing Arabic text, developed using Google's BERT design. It is designed to accommodate Arabic's distinct features, such as its complex morphology and writing system. AraBERT v0.2 enhances the initial version by utilizing a broader corpus that incorporates both Modern Standard Arabic and dialects, resulting in improved performance on various NLP tasks like text classification, sentiment analysis, and named entity recognition. It also contains improvements for managing Arabic accents and symbols.

2.2 Closed-source Generative Model for Arabic NLP

GPT-4o (OpenAI, 2024): Developed by OpenAI and incorporates multimodal capabilities, allowing it to process various inputs, including images, videos, audio, and text. GPT-4o, in contrast to GPT-4, shows improved efficiency by reducing token usage across multiple languages, including Arabic.

Gemini 1.5 (Pichai and Hassabis, 2024): Developed by Google and incorporates of advanced processing systems enhances its contextual understanding across languages, including Arabic, thus improving the accuracy of AI applications in natural language understanding, machine translation, and language generation tasks relevant to Arabic.

Command R+ (Gomez, 2024): Command R+ is 104 billion parameter multilingual LLM designed by Cohere for conversational interaction and tasks requiring long context. It focuses on excelling in tasks that require understanding and executing accurately.

ALLaM-1 (Bari et al., 2024): ALLaM, is a 13 billion parameter LLM for Arabic and English, developed by SDAIA, and is designed for a wide range of NLP applications. It is particularly suited for tasks such as text completion, question-answering, document summarization, classification, generation, and translation in Arabic.

2.3 Open-source Generative Model for Arabic NLP

JAIS-13b (Sengupta et al., 2023): JAIS is based on the GPT-3 decoder-only architecture with its focus on bilingual (Arabic and English) capabilities. JAIS aims to address a critical gap in the development of AI solutions for Arabic language speakers.

AceGPT-13b (Huang et al., 2023b): AceGPT is an open-source LLM developed specifically for Arabic, attuned to local culture and values, offering versatile functionality across multiple Arabic-specific applications.

SILMA v1.0 (SILMA, 2024): SILMA is an open-source 9 billion parameter LLM built over the foundational models of Google Gemma, and it is designed for tasks regarding text generation and summarization. The model is currently topping the list of open-source Arabic LLMs according to the OALL classification on Hugging Face (Almazrouei et al., 2023).

In this study, we aim to evaluate the performance of these models for Arabic punctuation prediction. We examine their capabilities under both zero-shot and few-shot learning paradigms.

3 Related Works

This section reviews recent developments in LLMs for Arabic NLP, focusing on their applications, performance, and limitations, particularly in the underexplored area of punctuation prediction.

3.1 Evaluating LLMs

The rise of generative LLMs like ChatGPT and Gemini signifies a breakthrough in generative modeling, showcasing human-like text generation

proficiency across diverse languages, including Arabic.

Several studies demonstrate their superior performance in translation tasks compared to commercial systems (Wang et al., 2023; Peng et al., 2023; Karpinska and Iyyer, 2023). (Bubeck et al., 2023) investigates GPT-4, showcasing its excellent performance across various tasks. (Espejel et al., 2023) experiment the reasoning ability of GPT-3.5, GPT-4, and BARD, highlighting the GPT-4's surpassed performance in zero-shot scenarios. (Laskar et al., 2023) thoroughly evaluates ChatGPT across 140 tasks, facilitating its effectiveness.

Numerous papers (Ogundare and Araya, 2023; Jiao et al., 2023; Bang et al., 2023) observe that ChatGPT is competitive with commercial products for high-resource languages but encounters difficulties with low-resource languages. Low-resource languages have also been investigated by (Ahuja et al., 2023; Lai et al., 2023).

Both (Ziems et al., 2024) and (Sottana et al., 2023) observe that while LLMs fall short of the best fine-tuned state-of-the-art (SoTA) models, they still achieve fair agreement levels with humans. Meanwhile, (Qin et al., 2023) highlights ChatGPT excels in reasoning tasks but faces challenges like sequence tagging. (Sottana et al., 2023) highlights the need for enhanced evaluation metrics for LLMs, identifying GPT-4 as a promising candidate for fulfilling this role. This emphasizes the importance of addressing the limitations in evaluation methodologies, which could contribute to the discrepancies observed in model assessments.

Recent studies reveal innovative methods to improve LLMs. Specifically, (Peng et al., 2023; Gao et al., 2023) conclude task-specific prompts enhance translation systems, while (Huang et al., 2023a) introduce cross-lingual-thought prompting (XLT) to improve cross-lingual performance. Furthermore, (Lu et al., 2023) suggests self-correction techniques for ChatGPT.

The findings of these studies suggest that while GPT-based LLMs are competent language models, their performance is comparable to the current SoTA model in most NLP tasks. However, none of these examinations specifically evaluate the punctuation prediction performance of LLMs.

3.2 Evaluating LLMs for Arabic NLP

The performance of LLMs has been evaluated in various Arabic NLP tasks. (Khondaker et al., 2023) evaluated ChatGPT's performance across 32 Arabic NLP tasks, revealing the necessity for enhancements in instruction-tuned LLMs. (Alyafeai et al., 2023) determined that GPT-4 surpasses GPT-3.5 in five out of the seven Arabic NLP tasks. (Huang et al., 2023b) introduces AceGPT, a culturally sensitive Arabic LLM, which outperformed within various Arabic benchmarks. (Kadaoui et al., 2023) evaluates the machine translation proficiency of ChatGPT (GPT-3.5 and GPT-4) and Bard across ten Arabic varieties, uncovering challenges with dialects lacking datasets. (Al-Thubaity et al., 2023) assesses ChatGPT (GPT-3.5 and GPT-4) and Bard AI for Dialectal Arabic Sentiment Analysis, revealing GPT-4's superior performance over GPT-3.5 and Bard AI.

LLMs, as demonstrated by (Khondaker et al., 2023; Alyafeai et al., 2023; Kwon et al., 2023), still fall short when compared to SoTA models fine-tuned on Arabic data.

Other studies have investigating evaluating smaller Arabic language models (Abu Farha and Magdy, 2021; Inoue et al., 2021; Alammary, 2022; Nagoudi et al., 2023; Elmadany et al., 2023b; Elmadany et al., 2023a).

3.3 Arabic Punctuation

In various languages, punctuation functions as a marker for delineating sentence boundaries. However, the interpretative clarity of this punctuation is often compromised, notably evident in instances involving acronyms or abbreviations. When the need arises to segregate sentences, it is imperative to employ a punction prediction technique adept at resolving such ambiguities. Recent research has made significant progress in punctuation prediction. (Zhou et al., 2022) and (Wu et al., 2016) have proposed models that outperform traditional methods for speech recognition, with Zhou's joint ASR-punctuation model showing notable promise. Similarly, (Yi et al., 2020) tackled the class imbalance issue in punctuation prediction training by incorporating focal loss, resulting in improved performance. Collectively, these studies underscore the potential of deep learning in enhancing punctuation prediction accuracy.

A range of studies have explored the prediction of punctuation and diacritics in the Arabic

language. Both (Aboutaib et al., 2023), (Sunkara et al., 2020), and (Mansour et al., 2023) reported high accuracy in punctuation prediction. (Sunkara et al., 2020) and model utilized BERT based pretrained language models, exhibiting robustness against automatic speech recognition errors. (Mansour et al., 2023) utilized a pre-trained transformer-based model such as ELECTRA and BERT. (Al-Najjar et al., 2020) concentrated on diacritization in Medieval Arabic utilizing a character-level neural machine translation approach. (Sakr and Torki, 2023) propose a new punctuation dataset and concluded that XLM-RoBERTa outperformed other transformer-based models in punctuation restoration.

4 Methodology

This section outlines the methodology employed in our study. In section 3.1, we detail the dataset utilized, including its composition and preparation. Subsection 3.2 describes the models employed in this research.

4.1 Dataset

In this research, we use a dataset sourced from the King Salman Global Academy for Arabic Language (KSAA), which includes 25 books. Since the data is taken from published books, it has been proofread for grammar and punctuation by linguistic experts to ensure accuracy and consistency.The data is available from the corresponding author on request.

To prepare the dataset, the books were preprocessed by automatically removing footnotes, indexes, and references. Following this, the text was divided into smaller paragraphs using tab delimiters and then saved as an Excel file for further preparation.

Each paragraph was carefully reviewed and cleaned manually by one annotator, involving:

- The removal of titles and non-paragraph elements (e.g., Footnotes and their reference numbers).
- Combining rows that were contextually related to form complete paragraphs.

In total, 10,046 data points were generated, each limited to a maximum length of 512 tokens after tokenization.

Next, the data is split into training, validation, and test sets. The training set contains 8,569 data points, the validation set contains 962 data points, and the test set contains 515 data points.

For each book, 90% of the content was used for the training set, while the remaining 10% was allocated to the validation set. We designated one book exclusively for testing, and its data is not included in either the training or validation sets.

The training data will be used to fine-tune the AraBERT model, with its performance assessed using the validation set. Once fine-tuned, AraBERT, along with all other language models mentioned in this study, will be evaluated on the test data to compare their effectiveness in the given task.

In this study, we focus on the prediction of six Arabic punctuation marks: period (.), comma (،), colon (:), semicolon (؛), question mark (؟) and exclamation mark (!). Table 1 shows the punctuation distribution among data splitting.

Marks	Train (85%)	Val (10%)	Test (5%)
.	23,156	2,612	1,245
،	42,287	4,472	2,931
:	6,517	622	321
؛	3,445	370	195
؟	568	82	27
!	124	15	5

Table 1: Punctuations distribution.

4.2 Model

In this section, a fine-tuned AraBERT model and several LLMs are introduced as the primary tools for tackling the task of punctuation prediction in Arabic texts.

4.2.1 Fine-tuning AraBERT for Arabic Punctuation Prediction

To fine-tune AraBERT v0.2, each text will be fed to the model along with its label. To prepare the text, we discard all punctuation marks, then, each text was then broken down into smaller units, typically words or subwords, referred to as tokens.

In contrast, to prepare the labeling, we first tokenize the text. Then, followed the method outlined in (Mansour et al., 2023), each token was encoded using underscores and punctuation marks: words without punctuation were replaced by underscores (_), while words followed by punctuation were substituted by the corresponding punctuation mark. We focused on encoding only

words that contain one of the six Arabic punctuation marks discussed earlier.

We ensured that the length of each tokenized text exactly matched the length of its label sequence, maintaining a one-to-one correspondence between tokens and their respective labels.

Simultaneously, labels indicating the presence or absence of punctuation for each token were converted into numerical indices through label encoding, following the mapping {"_": 0, ".": 1, "،": 2, "؛": 3, "؟": 4, "!": 5, ":": 6}. This transformation made the categorical label data suitable for model training, with each index corresponding to a specific punctuation mark. Thus, the input to the model consisted of the tokenized text without any punctuation, while the labels encoded the corresponding numerical label, as shown in Table 2.

Original text	أكل الولد الخبز، وشرب الماء.
No punct. tokenized text	[أكل, الولد, الخبز, وشرب, الماء]
Encoded label	._،__
Numerical label	[1,0,2,0,0]

Table 2: Fine-tune AraBERT input.

The tokenized texts were padded to ensure uniform length across batches. After padding, the tokens were embedded into dense vectors. We fine-tuned AraBERT v0.2 by adjusting several training parameters to optimize its performance for Arabic text processing. Specifically, we used a learning rate of 5e-5 and a batch size of 8, utilizing the AdamW optimizer in conjunction with a linear learning rate scheduler that employed zero warm-up steps. The model was trained for 5 epochs, a duration deemed sufficient for effective learning while minimizing the risk of overfitting.

4.2.2 LLMs for Arabic Punctuation Prediction

We utilized various LLMs, specifically GPT4-o, Gemni-1.5-flash-latest, jais-13b, AceGPT-13b, SILMA-9B-Instruct-v1.0, allam-1-13b-instruct, and command-r-plus-08-2024 in both zero-shot and few-shot scenarios. In the zero-shot approach, the models relied entirely on their pretraining knowledge without any additional fine-tuning. In the few-shot setting, they were provided with two examples of punctuation patterns, which improved their performance and demonstrated their ability to adapt to limited data scenarios.. We provided explicit directives against adding or deleting any word or letter from the original content to ensure effective implementation of the missing punctuation marks in the texts and enable its straightforward evaluation. We included two examples as an addition to the few-shot step. We ran the model on an NVIDIA A100 40GB GPU for efficient large-scale computation.

5 Results and Discussion

To assess the performance of the fine-tuned AraBERT model, we evaluate the model's performance using metrics: precision, recall and F1 score using the validation data. In addition, we analyze the overall accuracy of AraBERT in comparison to other LLMs mentioned in this study using the testing data, providing a comprehensive evaluation of model performance in predicting punctuation for Arabic text.

5.1 AraBERT Results

We investigated the performance of the fine-tuned AraBERT model on the evaluation dataset. As shown in Table 3, the model excels in recognizing some punctuation marks like the comma (،) and colon (:) but faces difficulties with others such as the semicolon (؛) and exclamation mark (!). The exclamation mark has a much lower F1 score than the other punctuation marks. The dataset has a highly uneven distribution of punctuation marks, potentially resulting in performance disparities. For instance, the exclamation mark is rarely used in comparison to commas, impacting the model's capacity to generalize and contributing to the small training size. The overall accuracy for the testing data reaches 29.78% among all punctuation marks. There is a significant contrast in performance for certain marks like the period (.), with precision at 54.87% and recall at 87.62%, suggesting the model accurately detects fewer true positive periods but has more false positives. Interestingly, the dataset size for the period is large compared to other

Marks	Precision	Recall	F1
_ (no punc)	98.63%	99.30%	98.97%
.	54.87%	87.62%	67.48%
،	86.02%	84.22%	85.11%
:	91.60%	87.97%	89.75%
؛	79.50%	46.58%	58.74%
؟	79.58%	90.66%	84.76%
!	69.57%	12.90%	21.77%

Table 3: Fine-tuned AraBERT result on validation set.

punctuation marks, which may influence this performance discrepancy.

Several factors may explain the high F1 score of 84.76% for the question mark (؟), despite its infrequent occurrence. Initially, question marks are commonly found in particular syntax and meaning situations, frequently in conjunction with question terms or formations, aiding in the model's understanding of where they belong. Furthermore, question marks are used more clearly and with less ambiguity than other punctuation marks, making them a more effective learning aid. The elevated F1 score could also be due to the model's increased recall, indicating it accurately detects most questions even if it sometimes mistakes other sentences as questions. Therefore, the model's strong performance on question marks is attributed to a combination of strong contextual cues, clear usage patterns, and high recall, despite the low training frequency.

5.2 LLM REsults

Upon investigating the models, we found that GPT4-o and Command R+ complied substantially with the majority of the proposed guidelines. In several instances, the models enhanced the quality of the text by inserting additional words do not present in the original content, as observed with Gemini 1.5. Conversely, other models exhibited some discrepancies in adhering to the given directions or generated completely different content, leading to the deletion of some original textual information. These elements complicated the process of evaluation.

As shown in Figure 1, the results reveal that GPT4-o and Command R+ performed substantially well in terms of adhering to the proposed guidelines, demonstrating higher accuracy compared to other models. However, Gemini 1.5 introduced additional words that were not part of the original content, complicating the evaluation process.

The few-shot method consistently improved the performance of most models, with GPT4-o achieving an accuracy of 66.57%, significantly higher than the zero-shot method. In contrast, models like SILMA and JAIS struggled with lower accuracy levels across both learning scenarios. Notably, JAIS took the longest time to complete the tasks, whereas SILMA was the fastest, highlighting the variability in processing efficiency. The results highlight that while LLMs show potential for punctuation prediction, their performance varies depending on the task and method, with some models requiring further refinement to improve consistency and adherence to the original text.

When analyzing the results by punctuation mark, the period (.) achieved the highest accuracy, as illustrated in Table 4. Notably, both the AceGPT and JAIS models showed significant improvement after employing the few-shot method. However, in comparison to AraBERT's performance, these models demonstrated stronger results. As shown in Table 4, AraBERT showed weaker performance relative to the LLMs and a decline in performance from the validation (Table 3) to the test set, reflecting its limited generalization capability.

Interestingly, even though the few-shot prompts did not include any question marks in the examples, the results still displayed some enhancements in the prediction accuracy of question marks, underscoring the potential of few-shot learning to improve performance across different punctuation marks.

5.3 Error Analysis

We aim to examine the errors made by these LLMs during the processing of Arabic text based on the test data. We aim to provide valuable insights that can contribute to the refinement of punctuation prediction LLMs, ultimately enhancing the efficiency of Arabic text processing.

We outline the main types of errors found in the test data:

- **Formatting or Sample Division:** In the original text, the phrase "كانوا يتكلمون اللغة العربية قبل ظهور الإسلام،واللغة العربية" had the word (الإسلام،) attached to the word (واللغة). The models GPT4-o and Gemini 1.5. separated

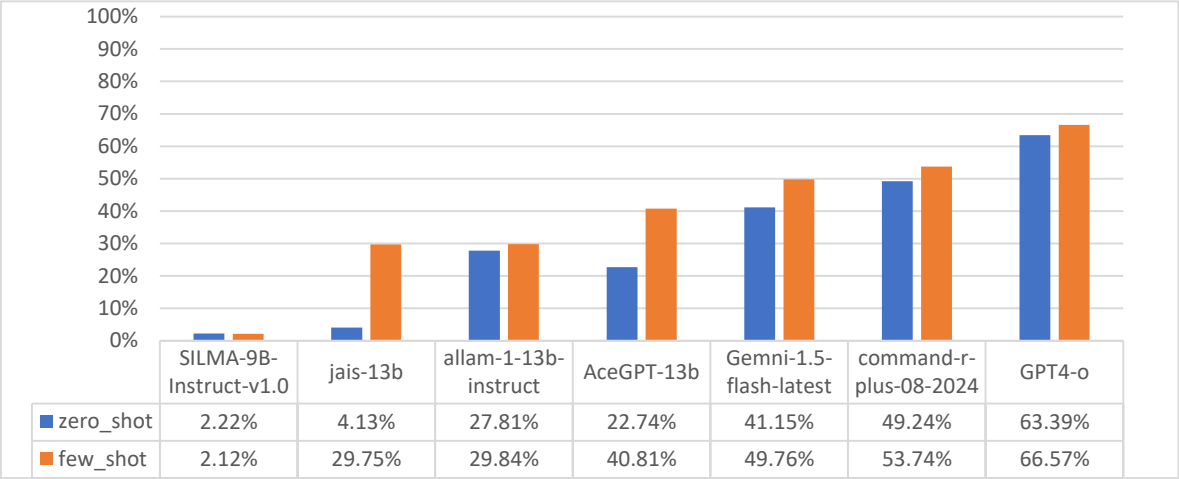

Figure 1: Average Accuracy among all punctuation marks.

Method	Model	.	،	؛	؟	!	:
Fine-tune	AraBERT	22.68%	31.36%	03.51%	26.66%	00.00%	34.28%
Zero-shot **Few-shot**	GPT4-o	**76.46%** 75.22%	59.74% **69.23%**	**34.58%** 19.92%	54.44% **63.33%**	20.00% 20.00%	**60.45%** 49.52%
	command-r-plus-08-2024	69.14% 72.91%	41.32% 47.16%	23.37% 17.06%	56.66% 50.00%	20.00% **40.00%**	55.93% 57.19%
	Gemni-1.5-flash-latest	50.95% 53.30%	39.57% 55.76%	29.11% 14.26%	56.66% 61.11%	20.00% 20.00%	19.10% 14.46%
	allam-1-13b-instruct	60.03% 66.15%	09.67% 08.70%	18.88% 17.51%	30.00% 36.66%	00.00% 00.00%	46.83% 46.87%
	AceGPT-13b	15.30% 58.49%	26.58% 32.31%	06.96% 18.03%	34.44% 43.33%	20.00% 00.00%	34.00% 48.16%
	jais-13b, AceGPT-13b	04.48% 32.55%	03.53% 27.50%	01.17% 12.11%	00.00% 20.00%	00.00% 20.00%	01.98% 27.87%
	SILMA-9B-Instruct-v1.0	00.77% 01.04%	02.97% 02.51%	00.00% 00.00%	00.00% 00.00%	00.00% 00.00%	02.64% 04.06%

Table 4: Average Accuracy par punctuation marks on test set.

these words while retaining the punctuation, but this was considered incorrect. Additionally, some text samples were very short and lacked context, which led to failures in punctuation, such as the phrase "ثالثًا الكتاب".

- **Writer's Mistakes:** It is important to note that an accurate score below 100% does not necessarily indicate a mistake by the model; in some cases, the model may be correcting errors in the original text. Consequently, a model achieving a perfect score (100%) might only signify alignment with the source text, even if that text contains inaccuracies. For example, most models corrected the original sentence: "... افتُتِحَت مجموعة من المعاهد العالية الإسلامية، التي كانت تستقبل الطلبة المتخرجين في ثانويات الأئمة والخطباء..." to: " افتُتِحَت مجموعة من المعاهد العالية الإسلامية التي كانت تستقبل الطلبة المتخرجين في ثانويات الأئمة والخطباء.". GPT4-o, the model that received accuracy of 100%, did not make this correction. Additionally, there were instances of complete loss of punctuation in the original text, as seen in the phrase: " العامية لغة العامة أما الفصحى فهي لغة الخاصة," which was corrected by Gemini1.5 to: " العامية لغة العامة، أما الفصحى فهي لغة الخاصة," yet it received a score of 0.

- **Differences in Usage Across Languages:** The application of punctuation rules from other languages to Arabic led to several issues. For instance, the original text stated:

"هناك أسباب كثيرة أدت إلى ظهور العامية منها:
العرق: ...
العامل الجغرافي: ...
العامل الثقافي: ...
الاستعمار: ..."
The models transformed this into:
" هناك أسباب كثيرة أدت إلى ظهور العامية، منها: العرق: .....؛ العامل الجغرافي.....؛ العامل الثقافي ....؛ الاستعمار..... "
except for Command R+, AceGPT-13b, and ALLaM.

- **Limited and Emotional Use of Certain Punctuation Marks:** An example is the exclamation mark (!) which appeared in only five instances, three of which were complex usages combined with the question mark (؟!). The models Command R+, ALLaM, and AceGPT used it correctly in standalone contexts, while one instance was in an explicit exclamatory expression: "يالألحزن!" which was correctly utilized by the models Command R+, Gemini1.5, and GPT4-4o. However, one instance was in a highly personal context that none of the models managed to punctuate correctly.

- **Partial Diacritical Marking in Arabic Texts:** The inability of some models, e.g. AraBERT, to handle the presence or absence of diacritical marks leads to the exclusion of any marked words, resulting in grammatically incorrect text that the model fails to punctuate appropriately.

5.4 Findings

The study highlights that models such as GPT4-o and Geni1.5 demonstrated robust zero-shot and few-shot learning capabilities. These findings suggest potential for handling languages such as Pashto and Sindhi, which share script similarities with Arabic.. Pashto and Sindhi exhibit unique syntactic and semantic features, which differ from Arabic. For example, Pashto uses diacritics more consistently than Arabic, and Sindhi's punctuation conventions may require additional adaptation of model pretraining or fine-tuning. While the LLMs in the are promising, their effectiveness in Pashto or Sindhi would depend on additional fine-tuning and dataset enrichment tailored to these languages. For fine-tuning, embedding models such as E5, which is known for its multilingual support, covers Persian and could be extended to Pashto, Sindhi, and Uyghur with additional pretraining on relevant datasets.

The presence of partial diacritics in the dataset introduced inconsistency, creating ambiguity for models such as AraBERT when predicting punctuation. Models such as GPT4-o demonstrated stronger generalization in both zero-shot and few-shot scenarios, effectively handling diacritic-related complexities in punctuation prediction. AraBERT, while less accurate overall, benefited significantly from fine-tuning on diacritic-inclusive datasets, showing improved accuracy compared to when diacritics were excluded.

Errors occur due to improper text segmentation, such as attached punctuation marks (e.g., "الإسلام،واللغة") or short, context-lacking samples. Writer's mistakes, such as missing or incorrect punctuation, lead models to correct text but result in mismatches during evaluation. Multilingual training causes cross-linguistic interference, applying non-Arabic punctuation rules. Rare punctuation marks, like exclamation marks (!), are underrepresented, limiting generalization. Lastly, partial diacritical marking creates ambiguity, making it difficult for models to interpret and predict punctuation accurately. Moreover, Rare punctuation marks such as the exclamation mark (!) and semicolon (؛) posed significant challenges due to their low frequency in the dataset, which limited the models' exposure to these patterns during training. In addition, their usage often occurs in complex contexts, such as emotional expressions or structured lists, making it challenging for models to predict them accurately. For example, the exclamation mark is commonly combined with other punctuation marks, such as "؟!".

6 Conclusion

This study demonstrates the effectiveness of LLMs, for punctuation prediction in Arabic texts. Our findings highlight the importance of dataset alignment and suggest promising avenues for enhancing NLP applications. Future research should focus on fine-tune LLMs on our dataset for this task, in addition to extending a more balanced dataset to tackle the issue of uneven data distribution and enhance the model's performance across all punctuation marks. These efforts will significantly advance the automation and quality of Arabic text processing. Moreover, the future work should focus on augmenting datasets with Rare

punctuation marks such as the exclamation mark (!) and semicolon (؛) employing context-aware training techniques to improve model accuracy and robustness.

Acknowledgments

This work was supported by the Arabic AI Center (ARAI) at KSAA, which provided computational resources for model training and covered the costs associated with generating output from closed-source LLMs.

References

Abdelkarim Aboutaib, Ahmad El allaoui, Imad Zeroual, and El Wardani Dadi. 2023. Punctuation Prediction for the Arabic language. In *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security*, New York, NY, USA. Association for Computing Machinery.

Ibrahim Abu Farha and Walid Magdy. 2021. Benchmarking Transformer-based Language Models for Arabic Sentiment and Sarcasm Detection. In Nizar Habash, Houda Bouamor, Hazem Hajj, Walid Magdy, Wajdi Zaghouani, Fethi Bougares, Nadi Tomeh, Ibrahim Abu Farha, and Samia Touileb, editors, *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 21–31, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual Evaluation of Generative AI.

Ali Saleh Alammary. 2022. BERT Models for Arabic Text Classification: A Systematic Review. *Applied Sciences*, 12(11).

Ebtesam Almazrouei, Ruxandra Cojocaru, Michele Baldo, Quentin Malartic, Hamza Alobeidli, Daniele Mazzotta, Guilherme Penedo, Giulia Campesan, Mugariya Farooq, Maitha Alhammadi, Julien Launay, and Badreddine Noune. 2023. AlGhafa Evaluation Benchmark for Arabic Language Models. In Hassan Sawaf, Samhaa El-Beltagy, Wajdi Zaghouani, Walid Magdy, Ahmed Abdelali, Nadi Tomeh, Ibrahim Abu Farha, Nizar Habash, Salam Khalifa, Amr Keleg, Hatem Haddad, Imed Zitouni, Khalil Mrini, and Rawan Almatham, editors, *Proceedings of ArabicNLP 2023*, pages 244–275, Singapore (Hybrid). Association for Computational Linguistics.

Khalid Al-Najjar, Mika Hämäläinen, Niko Partanen, and Jack Rueter. 2020. Automated Prediction of Medieval Arabic Diacritics. *ArXiv*, abs/2010.05269.

Abdulmohsen Al-Thubaity, Sakhar Alkhereyf, Hanan Murayshid, Nouf Alshalawi, Raghad Alateeq, Rawabi Almutairi, Razan Alsuwailem, Manal Alhassoun, and Imaan Alkhanen. 2023. Evaluating ChatGPT and Bard AI on Arabic Sentiment Analysis. In Hassan Sawaf, Samhaa El-Beltagy, Wajdi Zaghouani, Walid Magdy, Ahmed Abdelali, Nadi Tomeh, Ibrahim Abu Farha, Nizar Habash, Salam Khalifa, Amr Keleg, Hatem Haddad, Imed Zitouni, Khalil Mrini, and Rawan Almatham, editors, *Proceedings of ArabicNLP 2023*, pages 335–349, Singapore (Hybrid). Association for Computational Linguistics.

Zaid Alyafeai, Maged S. Alshaibani, Badr AlKhamissi, Hamzah Luqman, Ebrahim Alareqi, and Ali Fadel. 2023. Taqyim: Evaluating Arabic NLP Tasks Using ChatGPT Models. *ArXiv*, abs/2306.16322.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan AlRashed, Faisal A. Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, et al. 2024. ALLaM: Large Language Models for Arabic and English.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of

Artificial General Intelligence: Early experiments with GPT-4.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023a. Octopus: A Multitask Model and Toolkit for Arabic Natural Language Generation. In Hassan Sawaf, Samhaa El-Beltagy, Wajdi Zaghouani, Walid Magdy, Ahmed Abdelali, Nadi Tomeh, Ibrahim Abu Farha, Nizar Habash, Salam Khalifa, Amr Keleg, Hatem Haddad, Imed Zitouni, Khalil Mrini, and Rawan Almatham, editors, *Proceedings of ArabicNLP 2023*, pages 232–243, Singapore (Hybrid). Association for Computational Linguistics.

AbdelRahim Elmadany, ElMoatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023b. ORCA: A Challenging Benchmark for Arabic Language Understanding. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9559–9586, Toronto, Canada. Association for Computational Linguistics.

Jessica Nayeli López Espejel, El Hassane Ettifouri, Mahaman Sanoussi Yahaya Alassan, El Mehdi Chouham, and Walid Dahhane. 2023. GPT-3.5, GPT-4, or BARD? Evaluating LLMs Reasoning Ability in Zero-Shot Setting and Performance Boosting Through Prompts. In

Yuan Gao, Ruili Wang, and Feng Hou. 2023. How to Design Translation Prompts for ChatGPT: An Empirical Study.

Aidan Gomez. 2024. Introducing Command R+: A Scalable LLM Built for Business.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023a. Not All Languages Are Created Equal in LLMs: Improving Multilingual Capability by Cross-Lingual-Thought Prompting.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2023b. AceGPT, Localizing Large Language Models in Arabic. *ArXiv*, abs/2309.12053.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is ChatGPT A Good Translator? Yes With GPT-4 As The Engine.

Karima Kadaoui, Samar M. Magdy, Abdul Waheed, Md Tawkat Islam Khondaker, Ahmed Oumar El-Shangiti, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. TARJAMAT: Evaluation of Bard and ChatGPT on Machine Translation of Ten Arabic Varieties.

Marzena Karpinska and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. GPTAraEval: A Comprehensive Evaluation of ChatGPT on Arabic NLP. *ArXiv*, abs/2305.14976.

Sang Yun Kwon, Gagan Bhatia, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Beyond English: Evaluating LLMs for Arabic Grammatical Error Correction. *ArXiv*, abs/2312.08400.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. ChatGPT Beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning.

Md Tahmid Rahman Laskar, M. Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets.

Hongyuan Lu, Haoyang Huang, Dongdong Zhang, Haoran Yang, Wai Lam, and Furu Wei. 2023. Chain-of-Dictionary Prompting Elicits Translation in Large Language Models.

Youssef Mansour, Ashraf Elnagar, and Sane Yagi. 2023. Punctuation Prediction for the Arabic Language. In Abhishek Swaroop, Vineet Kansal, Giancarlo Fortino, and Aboul Ella Hassanien, editors, *Proceedings of Fourth Doctoral Symposium on Computational Intelligence*, volume 726 of *Lecture Notes in Networks and Systems*, pages 579–592. Springer Nature Singapore, Singapore.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, Ahmed El-Shangiti, and Muhammad Abdul-Mageed. 2023. Dolphin: A Challenging and Diverse Benchmark for Arabic NLG.

Oluwatosin Ogundare and Gustavo Quiros Araya. 2023. Comparative Analysis of CHATGPT and the evolution of language models.

OpenAI. 2024. Hello GPT-4o.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards Making the Most of ChatGPT for Machine Translation.

Sundar Pichai and Demis Hassabis. 2024. Our next-generation model: Gemini 1.5.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a General-Purpose Natural Language Processing Task Solver?

Abdelrahman Sakr and Marwan Torki. 2023. AraPunc: Arabic Punctuation Restoration Using Transformers. *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*:1–6.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, et al. 2023. Jais and Jais-chat: Arabic-Centric Foundation and Instruction-Tuned Open Generative Large Language Models.

SILMA. 2024. Empowering Arabic Speakers with Cutting-Edge Generative AI Technologies.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation Metrics in the Era of GPT-4: Reliably Evaluating Large Language Models on Sequence to Sequence Tasks. In *Conference on Empirical Methods in Natural Language Processing*.

Monica Sunkara, S. Ronanki, Kalpit Dixit, S. Bodapati, and Katrin Kirchhoff. 2020. Robust Prediction of Punctuation and Truecasing for Medical ASR. *ArXiv*, abs/2007.02025.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-Level Machine Translation with Large Language Models.

Xueyang Wu, Su Zhu, Yue Wu, and Kai Yu. 2016. Rich punctuations prediction using large-scale deep learning. *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*:1–5.

Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Ye Bai, and Cunhang Fan. 2020. Focal Loss for Punctuation Prediction. In *Interspeech*.

Zhikai Zhou, Tian Tan, and Yanmin Qian. 2022. Punctuation Prediction for Streaming On-Device Speech Recognition. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*:7277–7281.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can Large Language Models Transform Computational Social Science?