PeerJ Computer Science

Evaluating large language models' Arabic grammar error corrections and explanations

Kousar Mohi, Imtiaz Ahmad and Sa'ed Abed

Computer Engineering Department, College of Engineering and Petroleum, Kuwait University, Al-Shadadiya, Kuwait

ABSTRACT

Grammar Error Correction and Explanations (GECE) is considered a challenging task for under-resourced languages. Arabic is one such language as it lacks linguistic materials such as annotated corpuses, language supporting models, and even Natural Language Processing (NLP) tools. The study reported in the article was designed to evaluate the performance of Large Language Models (LLMs) in GEC and in generating adequate and relevant explanations for these corrections. The study explored the potential of the LLMs, GPT, Gemini, and Llama by using fine-tuning and two prompting techniques. The study also evaluated Arabic-specific LLM, ALLaM, using two prompting techniques. Additionally, the study compares the performance of LLMs with existing system called LanguageTool. The research examined whether prompting and fine-tuning techniques affected the quality of the explanations generated for the development of LLMs as useful tools in language learning. Human evaluation was applied to evaluate the quality and usefulness of the generated explanations. Our findings revealed that GPT-4o outperformed the other models based on the evaluation metrics used. The fine-tuned version of GPT-4o achieved the highest score of 78% in the Bilingual Evaluation Understudy (BLEU) metric, followed by the fine-tuned version of Llama and ALLaM's version uses few-shot prompting, which both scored 74%. The F0.5 metric of the Chunk-LEvel Multi-reference Evaluation (CLEME) indicated that the fine-tuning technique significantly increased the metrics for GPT-4o, Gemini, and Llama, which had precision scores of 45%, 25%, and 29%, respectively. Furthermore, the fine-tuned version of Llama, ALLaM using few-shot prompting, and the fine-tuned version of GPT-4o achieved the highest average Character Error Rate (CER) of 10%, 10%, and 11%, respectively. Overall, our study shows that targeted training, starting with examples and progressing to fine-tuning, leads to significant gains in grammar error correction accuracy and explanation quality. Accordingly, LLMs can serve as a reliable resource to teach the Arabic Language and automate the editing process.

Submitted 18 June 2025
Accepted 26 November 2025
Published 8 January 2026

Corresponding author
Sa'ed Abed, s.abed@ku.edu.kw

Academic editor
Nicole Nogoy

Additional Information and Declarations can be found on page 29

DOI **10.7717/peerj-cs.3486**

Subjects Artificial Intelligence, Computational Linguistics, Data Mining and Machine Learning, Natural Language and Speech, Text Mining

Keywords Grammar error correction and explanations, Natural language processing, Large language models, GPT-4o, Arabic language, Gemini, Llama, Zero-shot, Few-shot

INTRODUCTION

As described in *Miller (1951)* and *Dasopang (2025)*, language is the ability to speak. Developed from early childhood, it is an essential part of human expression and

communication. In contrast, machines lack the intrinsic ability to understand or generate human language without the assistance of complex artificial intelligence (AI) techniques as stated by *Raiaan et al. (2024)*. However, as mentioned in *Turing (2009)*, for decades, researchers have focused on empowering computers with human-level reading, writing, and conversational skills. Deep learning, computational advances, and the availability of large text corpora have all contributed to the development of LLMs. These models can recognize complicated language patterns and create text that closely mimics human conversation because they employ neural architectures with billions of parameters and self-supervised training on vast unlabeled datasets (*Shen et al., 2023*). Recently, Large Language Models (LLMs) have shown remarkable progress in the field of Natural Language Processing (NLP). Models such as Generative Pretrained Transformer (GPT), Gemini, and Llama, have been applied in various NLP tasks like text generation, translation, question answering, and classifications as stated by *Raheja et al. (2024)* and *Kobayashi, Mita & Komachi (2024a)*. A recent, comprehensive review by *Fan et al. (2024)* discussed the research trends in the period from 2017 to 2023, which included enhancements in essential algorithms, NLP tasks, and applications in fields such as medicine, engineering, social sciences and humanities.

OpenAI announced the first version of its GPT in 2018, called GPT-1, which is a transformer decoder-based model as stated by *Ghojogh & Ghodsi (2020)*. Older versions of GPT have influenced subsequent models, including GPT-4 and GPT-4o, which resulted in significant advances in language processing and creation (*Fan et al., 2024*). As mentioned by *Hurst et al. (2024)*, the GPT 4o version includes audio and video inputs in addition to the other inputs and was trained up to October 2023 on enormous datasets from various sources and materials. Meta introduced Llama3 in 2024, which is a herd of models that supports multiple languages, coding, reasoning and other features (*Grattafiori et al., 2024*). Meta improved the quantity and quality of data used for pre- and post-training Llama3 models. Llama3 was trained on 15T multilingual tokens. It has 8 billion, 70 billion, and 405 billion learnable parameters. Meta also developed extensions that support image and face recognition, in addition to speech understanding capabilities. Another multilingual LLM is Gemini, a family of transformer-decoder models, developed by Google. It has various versions: Ultra, Pro, Nano, and Flash, as mentioned by *Anil et al. (2023)*. New enhancements to Gemini models include audio and video support (*Georgiev et al., 2024*). Finally, the Arabic Large Language Model (ALLaM) is a well-known LLM developed by the Saudi Data & AI Authority (SDAIA) to support fluency with understanding of Arabic and English languages (*Bari et al., 2024*). It was trained on a model from scratch with seven billion parameters and three models initialized by Llama2 on scales of seven billion, 13 billion, and 70 billion parameters.

Owing to the critical role of deep learning techniques and the processing of enormous volumes of data, such models have displayed exceptional capabilities in handling a variety of languages. However, since there are more than 7,000 spoken languages, current research is concentrating on scaling LLMs' multilingual capabilities to handle more languages in various tasks as mentioned by *Lai, Mesgar & Fraser (2024)*, *Dang et al. (2024)* and *Mothe (2024)*. Arabic is one of the most challenging languages due to the complexity and the

richness of Arabic morphology, as stated by *Kwon et al. (2023)*. Moreover, Arabic is a collection of diverse languages and dialects in addition to Modern Standard Arabic (MSA).

As LLMs develop, investigation is now turning to linguistically complicated languages such as Arabic, whose complex morphology and distinct orthographic rules provide significant barriers to NLP. For around 300 million people, Arabic is their native language, and it is officially recognized in 27 states, as stated by *Saiegh-Haddad & Henkin-Roitfarb (2014)*. Also, being the language of the Qur'an, it serves as the global holy and liturgical medium for Muslims. As highlighted in *Ghazzawi (1992)*, there are different varieties of Arabic including classical Arabic, colloquial Arabic, and MSA. Classical Arabic is the language of the Qur'an and early Islam literature. MSA is the modernized form of classical Arabic and is the form used in media sources, speeches, academic writing, and so forth. Colloquial Arabic is the local dialect spoken in various countries. Arabic is highly organized and derivational, with a strong emphasis on morphology and syntax (*AlOyaynaa & Kotb, 2023*). Generally, while other language grammars are considered complex, Arabic grammar is unusually so, which makes grammar-checking a hard task. According to *Selim (2018)*, learning grammar is critical for two reasons: first, to prevent the Quranic language from corruption, and second, to provide non-native speakers with a baseline from which to build a correct grasp of the language.

This study investigated LLMs in order to enhance their performance in two distinguished NLP tasks involving MSA grammar. The first task was grammar error correction (GEC), which basically entails identifying textual grammatical errors and correcting them (*Zhang et al., 2023b*; *Wang & Yuan, 2024*). Applying GEC to Arabic reveals challenges due to the complexity of Arabic grammar and features (*Kwon et al., 2023*). Grammatical mistakes are ordinary for anyone writing in any language. These mistakes can disturb readers and lead to miscommunication (*Ingólfsdóttir et al., 2023*). Therefore, GEC is considered essential for anyone, especially non-native speakers, to guide and provide them with instant feedback to facilitate their individual learning journey (*Davis et al., 2024*). The second task was Grammar Error Correction Explanation (GECE) which is associated with the GEC task. In it, the system explains the reasons for the grammatical corrections applied (*Song et al., 2023*). Generating explanations for grammatical error corrections is helpful for readers to get a deeper understanding of grammar rules of MSA (*Song et al., 2023*). Also, it clarifies concepts and reduces confusion by identifying and understanding mistakes. Exploring GECE in LLMs and enhancing it in several languages can result in LLMs fostering more effective learning experiences (*Ye et al., 2024*).

The purpose of the study was to investigate and evaluate the performance of four language models, GPT, Gemini, Llama, and ALLaM, in handling Arabic grammar correction and generating reliable explanations. Furthermore, the grammatical error correction performance of LLMs was compared against the existing tool, LanguageTool (https://github.com/languagetool-org/languagetool). We used two datasets, anual Arabic Spelling-Error Correction *Corpus* (*Saty, Aouragh & Bouzoubaa, 2023*) and the Arabic Grammar Corrections Dataset (https://huggingface.co/datasets/s3h/arabic-grammar-corrections), to train LLMs and evaluate their efficiency for Arabic GEC and explanation

generation. This is achieved by fine-tuning the language models and adapting different prompting techniques (*i.e.*, zero-shot and few-shot). The metrics used to evaluate these LLMs include Cosine Similarity, Bilingual Evaluation Understudy (BLEU), Levenshtein Distance, Word Error Rate (WER), CER (Charachter Error Rate), Chunk-LEvel Multi-reference Evaluation (CLEME), Generalized Language Evaluation Understanding (GLEU), and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) as well as conducting human evaluation.

This research study was designed to answer the following questions:

1. How efficient can LLMs be in correcting Arabic grammar and explaining the corrections?
2. Do different prompting techniques affect the quality and clarity of explanations?
3. Which technique has the best performance: prompts or fine-tuning?

The main contributions of this article can be summarized as follows:

- It investigates the performance of GPT-4, Gemini, Llama, and ALLaM language models in dealing with Arabic GEC and generating helpful explanations for the produced corrections.
- It sheds light on strategies to enhance language model performance, such as prompt-based techniques and fine-tuning.

The remainder of the article will review the relevant literature on LLM, GEC and explanation, and different prompt engineering and fine-tuning techniques in the 'Related Work'. 'Methods' details the methodology and research pipelines, including descriptions of the models, datasets, and evaluation metrics. 'Discussion' presents the data analysis and discusses the results for each model using different techniques. Finally, 'Conclusions' offers the study's findings and conclusions.

RELATED WORK

LLMs are transformer-based AI algorithms trained on numerous datasets. They use deep learning to be capable of handling various tasks as mentioned in *Marvin et al. (2023)*. Recent studies, such as *La Cava & Tagarelli (2025)*, *Yu et al. (2023)*, *von Schwerin & Reichert (2024)* and *Liu et al. (2024)*, have discussed the difference between open-source and closed-source LLMs. According to *La Cava & Tagarelli (2025)*, open-source LLMs, such as Llama, can be used freely for any purpose, whereas closed-source models, such as GPT and Gemini, limit interactions to API access and do not allow access to the pretraining data (*von Schwerin & Reichert, 2024*). LLMs have generally demonstrated superior capabilities to understand and generate languages, outperforming previous systems in various NLP tasks as stated by *Kobayashi, Mita & Komachi (2024a)*.

Besides contributing significantly to recent NLP research, LLMs can also produce high-quality corrections in GEC like the GPT models (*Creutz, 2024*; *Loem et al., 2023*; *Zhang et al., 2023a*; *Park et al., 2024*). Additionally, recent studies have shown noticeable improvements in LLMs in producing relevant and meaningful explanations for GEC

including GPT models, Qwen, DeepSeek and Llama as referred to by *Kaneko & Okazaki (2023)*, *Song et al. (2023)*, *Li et al. (2024)* and *Ye et al. (2024)*. However, no research has explored GECE for Arabic yet. This is because Arabic GEC is quite challenging owing to the ambiguity of Arabic at the orthographic, morphological, syntactic, and semantic levels (*Kwon et al., 2023*; *Alhafni et al., 2023*; *Ingólfsdóttir et al., 2023*).

Grammar error correction

According to *Bryant et al. (2023)*, writing is a learned skill and an essential form of communication that can be challenging for non-native language speakers. *Fei et al. (2023)* mentioned that the evolution of NLP applications can assist non-native speakers in improving their writing skills. Within any given sentence, a GEC task can automatically identify and rectify grammatical, orthographic, and semantic errors as highlighted by *Kobayashi, Mita & Komachi (2024b)*. Previously, various approaches were implemented for GEC, such as classifiers, machine translation, edit-based approaches, and LLMs as detailed by *Bryant et al. (2023)*. Current research focuses on applying GEC in LLMs, which is challenging according to *Tang, Qu & Wu (2024)*. In *Creutz (2024)*, three LLMs were evaluated, namel GPT-3.5, GPT-4, and Claude v1, using a prompt-based approach in correcting grammatical errors in beginner-level Finnish learner texts on different temperature settings due to the non-deterministic nature of LLMs. GPT-4 outperformed GPT-3.5 and Claude v1 on GEC task.

Similarly, several research articles such as *Loem et al. (2023)*, *Zhang et al. (2023a)* and *Park et al. (2024)*, have evaluated the performance of GPT-3 in GEC task by using prompt-based methods such as zero-shot and few-shot settings. As was detailed by *Loem et al. (2023)*, they found that the performance of GPT-3 was effective when using appropriate instructions and clear examples. In addition to evaluating the performance of LLMs, *Zhang et al. (2023a)* evaluated their tolerance on texts containing different levels of noise/errors. The results showed that the level of noise can affect LLMs performance: the performance declines as the noise increases. Additionally, according to *Park et al. (2024)*, LLMs performance increases when few-shot techniques are applied (increased number of examples).

Grammar error correction explanations

As discussed in the previous section, GEC can improve writing by detecting and correcting textual errors. However, understanding the reason for a particular correction and identifying the type of error in a GEC system will help language learners to continuously improve their skills by learning from their mistakes following effective feedback (*i.e.*, explanations provided by GEC systems or LLMs) as described by *Park et al. (2024)*. As noted by *Song et al. (2023)*, GECE in LLMs is the task of explaining the reason for an applied correction. *Kaneko & Okazaki (2023)* proposed a method called controlled generation with prompt insertion (PI). In this method, corrected tokens are sequentially inserted in the LLM's explanation output as prompts to guide the LLM to generate more useful and illustrative explanations. Their study showed that using the PI method, there was a notable increase in the LLM's performance in explaining the reasons for corrections.

Another study by *Song et al. (2023)*, developed a two-pipeline stage for LLMs to generate an explanation for each grammar correction as a pair of erroneous and corrected sentences. Human evaluation indicated that more than 93% of the explanations produced by their pipeline method for German and Chinese were correct. Whereas in *Li et al.*'s *(2024)* article, the author used LLMs as explainers to train and provide explanations for their models to enhance their performance. Also, they used LLMs as evaluators to produce more reasonable Chinese GEC evaluations. One of their findings showed that their SEmantic-incorporated Evaluation framework displayed a significant performance, which made it a suitable evaluation tool for GEC in LLMs. Furthermore, *Ye et al. (2024)* introduced a benchmark featuring the design of hybrid edit-wise explanations. Each edit is structured as follows: error type, error severity level, and an error description that helps learners and guides them to clearly understand why and how the grammatical error was corrected.

Prompt engineering and fine-tuning

Prompt engineering is the process of crafting and optimizing prompts to acquire the desired responses from LLMs, as outlined by *Marvin et al. (2023)*. According to *White et al. (2023)*, a prompt is a series of instructions that unlocks the full potential of LLM by customizing and enhancing its capabilities. Prompts are crucial for leading LLMs to create meaningful and relevant content. Techniques such as fine-tuning, in-context learning (ICL), zero-shot and few-shot learning, tailor LLMs for specific tasks, as mentioned by *Marvin et al. (2023)*.

As emphasized by *Pajak & Pajak (2022)*, the fine-tuning technique uses a supervised learning process to train language models to perform effectively faster and with less power consumption in a specific task. ICL is a technique where the descriptions of tasks are provided in the prompt, as well as a few annotated task examples as described by *Yao et al. (2024)*. *Wei et al. (2021)* created an instruction tuning method to improve both zero-shot and few-shot ICL. Moreover, *Yao et al. (2024)*, through enhancing the construction of multiple ICL prompts, developed a new technique that produces confident predictions. By contrast, zero-shot prompting is a technique of plainly describing the information of a task without providing examples as outlined by *Allingham et al. (2023)*. In addition to task information, the few-shot prompting technique includes multiple examples (*Chen et al., 2023*). Recent research has studied the application of prompt-based approaches in applying LLMs to GEC, concentrating on developing effective prompts that produces corrected sentences as detailed by *Zeng et al. (2024)*. For example, in *Kwon et al.*'s *(2023)* study, they found that in-context few-shot learning effectively improved the performance of GPT-4. Conversely, *Davis et al. (2024)* observed that in some settings, zero-shot prompting is as competitive as the few-shot technique. Other studies, like that of *Kaneko & Okazaki (2023)*, looked into different prompting strategies that improve LLMs' explanation of corrections. *Kaneko & Okazaki (2023)* created a prompt insertion method to enhance the explanation generation of GPT-3 and ChatGPT models. Finally, *Ye et al. (2024)* used a fine-tuning strategy for edit extraction and a few-shot prompting technique to prompt GPT-4 to generate edit-wise explanations.

To our knowledge, this is the first study to discuss and evaluate Arabic grammar correction explanations using LLMs. We developed and compared different prompting techniques (zero-shot, few-shot) and targeted fine-tuning to encourage LLMs to deliver concise, well-structured explanations in Arabic. By systematically comparing various techniques and measuring both corrective performance and explanatory clarity, our study fills a fundamental gap and sets a framework for understandable, educationally relevant GEC systems for Arabic.

MATERIALS AND METHODS

Experiment setting

In this article, to run and analyze the data, we used a Lenovo YOGA 9i with 16 GB RAM and 1T storage using a Windows operating system. Additionally, we utilized Python3 on Google Colab, running data on both CPU and T4 GPU environments.

Model selection

Selection of the language models in this study was based on the performance evaluations of recent studies (*Lai, Mesgar & Fraser, 2024*; *Kwon et al., 2023*; *Raheja et al., 2024*; *Zhang et al., 2023b*). Our study involved prompting and fine-tuning several well-known models, including Open AI GPT-4o, Google Gemini, Meta Llama3, and SDAIA ALLaM. OpenAI newly developed GPT-4o (https://platform.openai.com/docs/models) family such as GPT 4o, GPT-4o mini, gpt-4-turbo, *etc*., that can be accessed *via* OpenAI API requests for prompting and fine-tuning purposes. Similarly, Google has developed new Gemini models (https://ai.google.dev/gemini-api/docs/models): Ultra, Pro, Flash, and Nano, which are designed for specific applications and can be accessed through Google AI Studio for both prompting and fine-tuning. Notably, Google documentation indicates that Gemini 1.5 Flash is the only model in the Gemini family currently available for fine-tuning; it imposes constraints on input size, limiting it to 40,000 characters, and output size, restricting it to 5,000 characters per training example. Meta (https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2/) has been iteratively enhancing its Llama models. This has culminated in the current version, Llama 3, which incorporates specific modifications in the prompt format. Furthermore, SDAIA and the National Center for AI (NCAI) have collaboratively developed a series of LLMs specifically tailored to support the Arabic language namely ALLaM (https://www.ibm.com/docs/en/SSYOK8/wsj/analyze-data/assets/ALLaM-1-13b-instruct-model-card.pdf). ALLaM is a pretrained model derived from Llama that has three variants: ALLaM-7B, ALLaM-13B, and ALLaM-70B. GPT has shown remarkable performance in GEC, as has Llama (*Lai, Mesgar & Fraser, 2024*; *Kwon et al., 2023*; *Raheja et al., 2024*; *Zhang et al., 2023b*). However, there is no research using Gemini or ALLaM for GEC. Furthermore, the study compares the LLMs performance against the existing tool, LanguageTool, which is a multilingual AI-based grammar checker that supports Arabic language.

Datasets

In this study, we used two datasets, a manual Arabic spelling-errors correction *corpus* (https://lindat.mff.cuni.cz/repository/xmlui/handle/11372/LRT-4763) and the Hugging Face

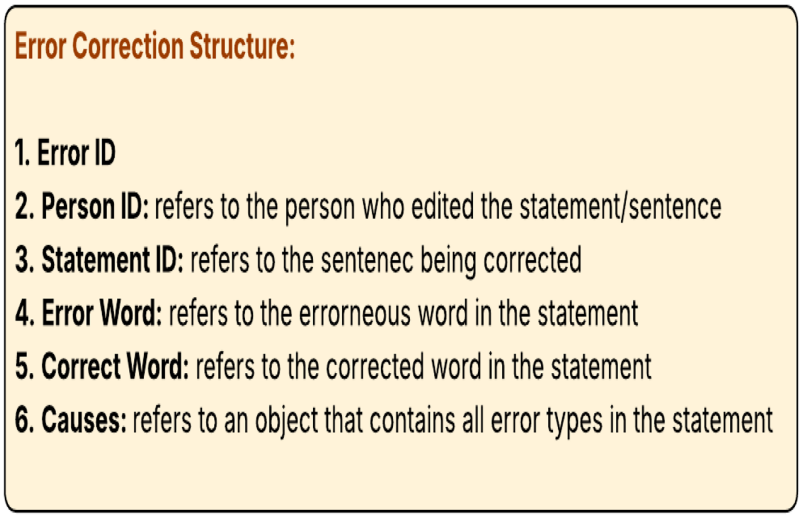

Figure 1 **Data structure available in manual Arabic spelling-errors correction *corpus*.**
Full-size DOI: 10.7717/peerj-cs.3486/fig-1

Arabic GEC dataset (https://huggingface.co/datasets/s3h/arabic-grammar-corrections). Manual Arabic spelling-errors correction is a text *corpus* designed for Arabic spell-checking; it was compiled from various files edited by a group of individuals and published by Sudan University of Science and Technology (*Saty, Aouragh & Bouzoubaa, 2023*). The *corpus* serves Arabic NLP by providing a comprehensive and open Arabic spell check resource ready for further exploration and analysis. The *corpus* consists of 11,098 words containing 1,888 errors and 20 error types, structured into several sections starting with the person, the documents he/she edited, types of errors, and the specific errors made. Each section contains data that elaborate on its content, which assists researchers in extracting valuable insights. The *corpus* has an array of error objects, and each object has the keys shown in Fig. 1.

We chose this dataset to pre-train the base models—GPT-4o, Gemini Pro and Llama3—to create our fine-tuned models. According to the LLMs' documentations, an average of 100 examples is generally sufficient to yield promising results. After data preprocessing, we compiled approximately 300 examples, meeting the recommended sample size outlined in each model's documentation for pre-training the models. Figure 2 shows the 20 types of errors on which the LLMs were trained. These types of errors were used in the training *corpus*, the Manual Arabic Spelling-Errors Correction *corpus*.

The second dataset was the Arabic GEC dataset shared on the Hugging Face platform. It has over 390,000 erroneous sentences and their corrections (reference sentences). We chose the first 2,000 records to evaluate the ability of both base and fine-tuned models in correcting erroneous sentences and explaining the purpose of corrections. Figure 3 presents a sample of the dataset.

Upon evaluating the fine-tuned models, we found that GPT-4o and Llama reproduced the training output format nearly identically. However, Gemini failed to generate the structured output on which it was trained, providing only error analysis without corrected

No.	Error Type in English	Error Type in Arabic
1	add space	اضافة مسافة
2	adjacent letter	الضغط على الحرف المجاور
3	context-sensitive error	الكلمة صحيحة ولكن استخدمت في غير موضعها
4	dialect error	اخطاء لهجات محلية
5	deletion error	حذف حرف
6	double letter	تكرار الحرف
7	double word	تكرار كلمة
8	exchange latter	تبديل الأحرف
9	forget press on space	حذف المسافة
10	Hamza error	أخطاء الهمزة
11	insertion error	إضافة حرف
12	lam shmsia	اللام الشمسية
13	phonetic error	أخطاء مخارج أو نطق
14	press adjacent letters at same time	الضغط على زرين متجاورين في نفس الوقت
15	permanent error	خطأ تبديل
16	shift keyboard	اختيار أحد الحرفين في زر واحد بالضغط على زر الأعلى
17	speed error	خطأ نتيجة للسرعة
18	taa marbota error	أخطاء التاء المربوطة
19	shape error	أخطاء تشابه الأحرف
20	Alef magsora and Yaa confusion	الإلتباس بين الألف المقصورة والياء في نهاية الكلمة

Figure 2 **Error types in the Manual Arabic Spelling-Errors Correction *corpus*.** Full-size DOI: 10.7717/peerj-cs.3486/fig-2

Erroneous Sentence	Corrected Sentence
الاول ومرشد ناصر الريامي من نادي الووحدة صاحب المركز الثاني عل صدارة السباق وجاء في المركز ثلثالث المتسابيق سليمان بن سعيد العلوي من نادي ينقل ول رابعا الدراج عمير بن .	الاول ومرشد ناصر الريامي من نادي الوحدة صاحب المركز الثاني على صدارة السباق وجاء في المركز الثالث المتسابق سليمان بن سعيد العلوي من نادي ينقل وحل رابعا الدراج عمير بن .
خلفان الرواحي من نادي مسقط وجاء في الركز اخامس المتسابق احمد بن سيف بالرحيى من نادي الوحدة .	خلفان الرواحي من نادي مسقط وجاء في المركز الخامس المتسابق احمد بن سيف الرحبى من نادي الوحدة .
وفي الترتيب العام للاندية جاء في امركز اولاول اناي الوحدة برصيد 40 نقطة وفي المركز الثاني نادي نزوى برصيد 30 نقطة وحل الث نادي النصر وفي نهاية السباق اقام راعي .	وفي الترتيب العام للاندية جاء في المركز الاول نادي الوحدة برصيد 40 نقطة وفي المركز الثاني نادي نزوى برصيد 30 نقطة وحل ثالث نادي النصر وفي نهاية السباق قام راعي .
المناسبة بوزيع الدايا والجوائز على اصحابا الماكاكز التقدمة .	المناسبة بتوزيع الهدايا والجوائز على اصحاب المراكز المتقدمة .
وبعد اختام اشاد حمد بن الياس فقيير بالمنافسة الوية وبالمستوى التنظيمى الرائع للبطولة والمشاركة ةالكبيةرة من مختلف اندية محافظات ومناطق السلطنة والتي تدل على وجود اهتمام كبير لهذه الرياضة بالسلطنة .	وبعد الختام اشاد محمد بن الياس فقير بالمنافسة القوية وبالمستوى التنظيمى الرائع للبطولة والمشاركة الكبيرة من مختلف اندية محافظات ومناطق السلطنة والتي تدل على وجود اهتمام كبير لهذه الرياضة بالسلطنة .

Figure 3 **Sample of Hugging Face Arabic GEC dataset.** Full-size DOI: 10.7717/peerj-cs.3486/fig-3

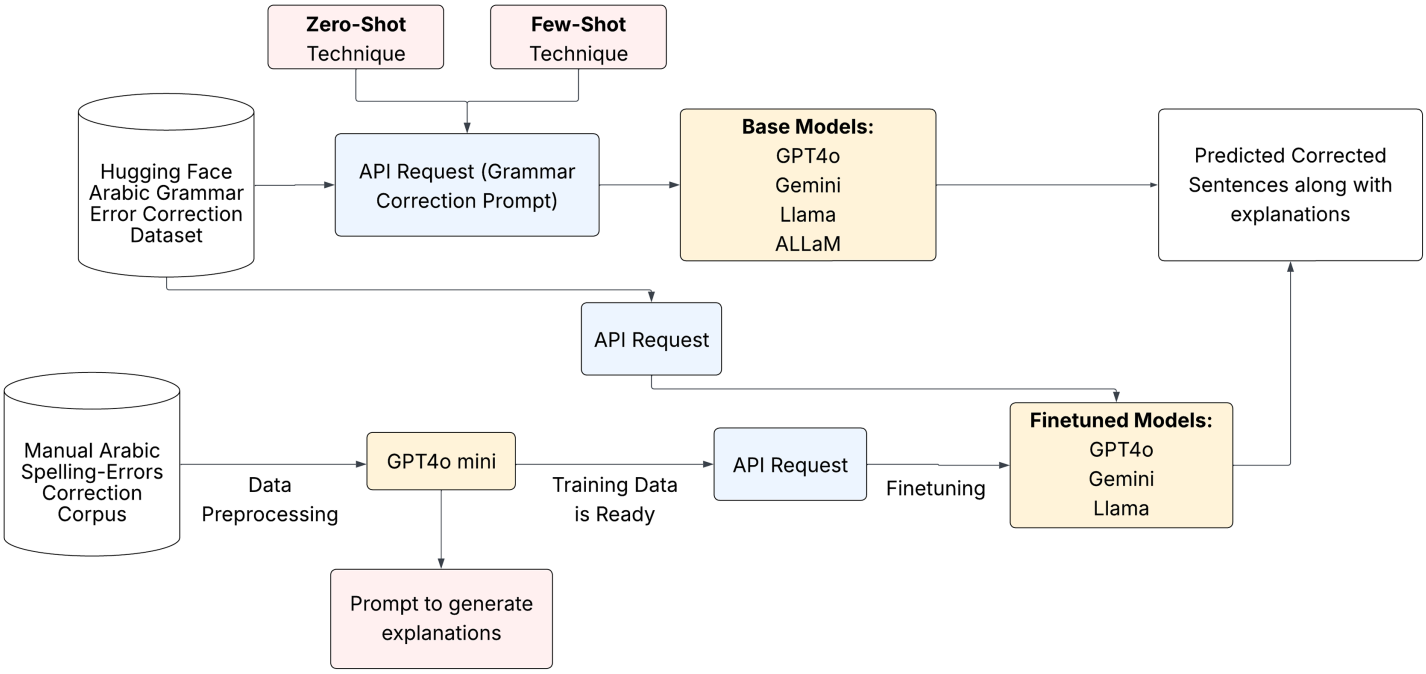

Figure 4 **Experiment pipeline.** Full-size DOI: 10.7717/peerj-cs.3486/fig-4

sentences. This output prevented many rows from being evaluated by our similarity metrics. As a result, we filtered the Gemini outputs to include only those records containing corrected sentences, resulting in a cleaned dataset of 379 entries used in the analysis.

Approach

We aimed to prompt and fine-tune the chosen LLMs to evaluate their capabilities for Arabic GEC and GECE tasks. Figure 4 shows the pipeline of our full approach. We used particular prompts to produce precise corrections and meaningful explanations of the language models.

Data preprocessing

For fine-tuning, we first preprocessed the Manual Arabic Spelling Errors Correction *corpus*. Specifically, we adopted its output structure while excluding the first three keys: 'documentID', 'statementID', and 'PersonID'. Each model requires a different training data format; however, they all follow the same principle of providing an input along with its corresponding output. To enhance the training data, we prompted GPT-4o-mini to include an additional key called "explanation" that provides the rationale behind each correction. This makes the output both informative, comprehensive and offers a complete understanding of the error correction process. Figure 5 shows the prompt used for generating explanations.

Figure 6 illustrates a sample of the training data. The first column represents the erroneous sentence, while the second column reflects the desired output structure as shown in Fig. 7.

Add Explanation Prompt:

Explain why do we need to correct ${erroneuos_word} to this $ {corrected_word} where the Arabic grammatical error type is related to ${errorType_English} (${errorType_Arabic}). Summarize the explanation in one paragraph without points

Figure 5 **Add explanation prompt.** Full-size DOI: 10.7717/peerj-cs.3486/fig-5

كيف أطور مهارتي في الكتابة؟	Number of Errors Found: 1 Corrected sentence:كيف أطور مهارتي في الكتابة؟ - Errors Found - Error ID: 1 - Erroneous Word: - Error Types (English): Hamza error - Error Types (Arabic): أخطاء الهمزة - Correction: - Explanation: The correction of ""اطور"" to "" أطور""" is necessary due to the rules governing the use of the Hamza in Arabic. In this case, the word should begin with a Hamza to indicate the proper pronunciation and to adhere to the grammatical structure of the language. The Hamza serves as a glottal stop and is essential for distinguishing the correct form of the verb, which in this instance is in the first-person singular present tense of ""to develop"" or ""to evolve."" Without the Hamza, the word not only loses its intended meaning but also becomes grammatically incorrect, highlighting the importance of proper Hamza usage in Arabic writing and communication.

Figure 6 **Sample of the training data.** Full-size DOI: 10.7717/peerj-cs.3486/fig-6

Output Structure:

1. Number of errors found in the sentence
2. Final corrected sentence
3. For each error found:
   - Error ID
   - Erroneous Word
   - Error Types (in English)
   - Error Types (in Arabic)
   - Correction
   - Explanation

Figure 7 **Output structure.** Full-size DOI: 10.7717/peerj-cs.3486/fig-7

Zero-Shot Prompt:

Correct the following Arabic sentence and provide only the following:
1. Corrected Sentence:
2. Number of errors found to be explained
3. For each error found, write the
1. corrected word,
2. define the error type in English,
3. define error type in Arabic,
and 4. explain why do we need to correct the errors based on the error types for each error separately: ${erroneous_sentence}

Figure 8 **Zero-shot prompt.** Full-size DOI: 10.7717/peerj-cs.3486/fig-8

Few-Shot Prompt:

Correct the following Arabic sentence and provide only the following:
1. Corrected Sentence:
2. Number of errors found to be explained
3. For each error found, write the
1. corrected word,
2. define the error type in English,
3. define error type in Arabic,
and 4. explain why do we need to correct the errors based on the error

types for each error separately: كيف اطور مهارتيفي الاستماع؟

Corrected Sentence: كيف أطور مهارتي في الاستماع؟

Number of Errors Found: 2
Errors Found

Figure 9 **Few-shot prompt-1.** Full-size DOI: 10.7717/peerj-cs.3486/fig-9

Model fine-tuning and prompting techniques

Fine-tuning is the process of training a pre-trained model on a more detailed dataset to enhance its performance on a certain task. As highlighted in *Mathav Raj et al. (2024)*, researchers have shown that fine-tuning technique yields promising and more accurate results than creating a model from scratch. After preparing the training data for each model, we fine-tuned each model using its respective platform. For Llama, we used Laminiai, a third-party application to fine-tune the model, where a number of hyperparameters were adjusted using the default settings in order to improve stability and performance. The default learning rate used in Laminiai is 0.0009. Gemini was fine-tuned using Google AI Studio at a learning rate of 0.001 and the other default settings, while GPT-4o was fine-tuned using the recommended hyperparameters of OpenAI using API requests. For prompting, we employed two common techniques: zero-shot and few-shot prompting. The zero-shot approach prompts LLMs without providing examples to measure their natural capabilities. We asked each LLM to correct the erroneous sentence

Few-Shot Prompt (continue):

- Error ID: 1
- Erroneous Word: اطور
- Error Types (English): Hamza error
- Error Types (Arabic): أخطاء الهمزة
- Correction: أطور
- Explanation: The correction of "" اطور"" to "" أطور "" is necessary due to the rules governing the use of the Hamza in Arabic. In this case, the word should begin with a Hamza to indicate the proper pronunciation and to adhere to the grammatical structure of the language. The Hamza serves as a glottal stop and is essential for distinguishing the correct form of the verb, which in this instance is in the first-person singular present tense of ""to develop"" or ""to evolve."" Without the Hamza, the word not only loses its intended meaning but also becomes grammatically incorrect, highlighting the importance of proper Hamza usage in Arabic writing and communication.

- Error ID: 2
  Erroneous Word : مهاراتيفي
  Error Types (English): forget press on space
  Error Types (Arabic): حذف المسافة
  Correction: مهاراتي في
  Explanation: The correction from "" مهاراتيفي "" to "" مهاراتي في "" is necessary because the original phrase suffers from a spacing error where the words are incorrectly combined. In Arabic, proper spacing between words is crucial for clear communication and grammatical accuracy. "" مهاراتي "" means ""my skills,"" while ""في"" means ""in."" Without the appropriate space, the meaning becomes obscured, and the phrase loses its intended clarity. Ensuring correct spacing not only adheres to grammatical standards but also enhances readability and understanding in the Arabic language.

Correct the following Arabic sentence and provide only the following:

1. Corrected Sentence:
2. Number of errors found to be explained
3. For each error found, write the
1. corrected word,
2. define the error type in English,
3. define error type in Arabic,
and 4. explain why do we need to correct the errors based on the error types for each error separately:

اللغة نظام كليّ يتكون من مجموعة من الانظمة الفرعية، وكل نظام فرعي يتكون من مجموعة من المستويات، وكلما كانت نظرتنا كلية شاملة الي اللغة تعلمناها بشكل افضل وشمولي.

Corrected Sentence:

اللغة نظام كليّ يتكون من مجموعة من الأنظمة الفرعية، وكل نظام فرعي يتكون من مجموعة من المستويات، وكلما كانت نظرتنا كلية شاملة إلى اللغة تعلمناها بشكل أفضل وشمولي.

Number of Errors Found: 2
Errors Found
- Error ID: 1
  Erroneous Word: الانظمة
  Error Types (English): Hamza error
  Error Types (Arabic): أخطاء الهمزة
  Correction: الأنظمة

Figure 10 **Few-shot prompt-2.** Full-size DOI: 10.7717/peerj-cs.3486/fig-10

Few-Shot Prompt (continue):

Explanation: The correction of "" الانظمة "" to "" الأنظمة "" addresses a common Hamza error in Arabic grammar. In this case, the word "" الانظمة "" (the systems) requires the Hamza (ء) to indicate the proper pronunciation and grammatical structure, as it follows the definite article ""ال"" The absence of the Hamza in "" الانظمة "" leads to a misreading and misunderstanding of the word's form and meaning. Correctly placing the Hamza ensures that the word conforms to standard Arabic orthography and accurately reflects its intended meaning, maintaining the integrity of the language.

- Error ID: 2
  Erroneous Word: افضل
  Error Types (English): Hamza error
  Error Types (Arabic): أخطاء الهمزة
  Correction: أفضل
  Explanation: The correction of "" افضل "" to "" أفضل "" is necessary due to the proper placement of the Hamza, which is a critical aspect of Arabic orthography. In Arabic, the word "" أفضل "" (meaning ""better"" or ""best"") requires a Hamza on the letter 'I' to indicate the correct pronunciation and to distinguish it from similar words. The absence of the Hamza in "" افضل "" alters both the meaning and the grammatical correctness of the word, as it fails to adhere to the rules of Arabic script that dictate the use of Hamza in certain contexts. Thus, ensuring the proper placement of Hamza is essential for conveying the intended meaning accurately and maintaining the integrity of the language.

Now, correct the following Arabic sentence and provide only the following:
1. Corrected Sentence
2. Number of errors found to be explained
3. For each error found,
write the 1. corrected word,
2. define the error type in English,
3. define error type in Arabic,
and 4. explain why do we need to correct the errors based on the error types for each error separately: ${erroneous_sentence}

Figure 11 **Few-shot prompt-3.** Full-size DOI: 10.7717/peerj-cs.3486/fig-11

and provide the corrected sentence, the number of errors found in the sentence (and for each error, the corrected word and the type of error, identified in English and Arabic), and finally, a detailed explanation for the corrections applied. Figure 8 shows the zero-shot prompt.

In contrast, the few-shot approach uses a few examples in the prompts to supervise the models to generate the desired output format and style. In this approach, we used the same prompt used in zero-shot in addition to two examples as shown in Figs. 9, 10, 11.

Evaluation method

Method

In this article, we used the cross-dataset evaluation method in which the dataset used to train the LLMs is different from the dataset used to evaluate the performance and accuracy of both base and fine-tuned models. As mentioned in the previous sections, we used the Manual Arabic spelling-errors correction *corpus* to train the fine-tuned models and the Hugging Face Arabic GEC dataset to evaluate them.

Performance metrics

Several performance metrics were used to measure the effectiveness of LLM in the generation of Arabic GEC and GECE. What distinguishes these metrics is their ability to capture various aspects of error correction while ensuring an inclusive evaluation of the language models' performance.

CLEME is a reference-based metric used for evaluating GEC systems. It aims to provide unbiased F0.5 scores as referenced by *Ye et al. (2023)*. CLEME works on avoiding bias in GEC multi-reference assessment by converting the source, hypothesis, and references into consistent chunk sequences categorized as unchanged, corrected, or dummy. To measure the alignment of edits with these chunk boundaries across multiple references, CLEME calculates three statistics—the In-Corrected-Chunk (ICC) ratio, In-Unchanged-Chunk (IUC) ratio, and Cross-Chunk (CC) ratio-using Eqs. (1), (2), and (3):

$$ICC = \frac{1}{M}\sum_{i=1}^{M} f_1(e_i) \tag{1}$$

$$IUC = \frac{1}{M}\sum_{i=1}^{M} f_2(e_i) \tag{2}$$

$$CC = 1 - ICC - IUC \tag{3}$$

where M is the number of edits in a remaining reference, and e represents a single edit. f_1 returns 1 if an edit e_i is included in a corrected or dummy chunk, whereas f_2 returns 1 if e_i is included in an unchanged chunk.

Another metric was ROUGE, as outlined by *Lin (2004)*, it is a collection of measurements for evaluating the quality of computer-generated summaries by comparing them to human-created reference summaries. It computes the overlap of n-grams, word sequences, and word pairs in the produced and reference summaries using Eq. (4):

$$\text{ROUGE}-\text{N} = \frac{\sum_{S\in\{\text{ReferenceSummaries}\}}\sum_{gram_n\in S}\text{Count\_match}(gram_n)}{\sum_{S\in\{\text{ReferenceSummaries}\}}\sum_{gram_n\in S}\text{Count}(gram_n)} \tag{4}$$

where n represents the length of the n-gram, $gram_n$, and Count_match ($gram_n$) is the maximum number of n-grams overlapping in a candidate summary and a set of reference summaries. GLEU, designed for GEC, is a variant of BLEU considering both the source and reference results in a more accurate representation of human judgment, as discussed by *Napoles et al. (2015)*. It evaluates the n-gram overlap between the corrected output and reference sentences, penalizing superfluous alterations that do not correspond to the

reference. The GLEU score is calculated as a weighted precision of n-grams, with a shortness penalty comparable to the BLEU to account for recall. Equation (5) shows the formula for GLEU:

$$GLEU(C, R, S) = BP \exp\left(\sum_{n=1}^{N} w_n \log p_n{}'\right) \tag{5}$$

where C are the sentences, R are the references, S is the source, BP represents the brevity penalty, $p_n{}'$ represents the modified n-gram precision and w_n is the weight of n-gram precision.

For Arabic sentences, Cosine Similarity is a well-known metric typically implemented using an Arabic tokenizer. Using Eq. (6), we computed the cosine of the angle between the sentence vectors A and B, which contain lexical and contextual information. Similarity scores approaching 1 indicate a high degree of similarity.

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|}. \tag{6}$$

BLEU rates translation units-typically sentences-by comparing them with high-quality reference translations as described by *Sallam & Mousa (2024)*. BLEU calculates an n-gram overlap score for each segment of the *corpus*, then averages these scores over all segments, as outlined by *Papineni et al. (2002)*. It works by identifying contiguous sequences of n words in the candidate and reference texts; higher values indicate greater overlap and, thus, better fidelity to the reference, as outlined in Eq. (7).

$$\text{BLEU}_w(\hat{S}; S) := \text{BP}(\hat{S}; S) \cdot \exp\left(\sum_{n=1}^{\infty} w_n \log p_n(\hat{S}; S)\right) \tag{7}$$

where BP is the brevity penalty, w is the weights for each n-gram and p is the precision of n-grams.

In analysis of variance (ANOVA), the means of three or more groups are compared to determine whether they differ significantly from one another as mentioned by *Keselman et al. (1998)*. In Eq. (8), the total variability of the study is decomposed into variance between groups and variance within groups, which are then compared using an F-statistic calculated from the variance within and between groups. As in *Keselman et al. (1998)*, it shows that the resulting p-value indicates whether the observed differences in group means are statistically significant. In this study, ANOVA was employed to assess the significance of our findings across the various analyses and tests conducted.

$$\text{F-statistic (ANOVA Coefficient)} = \frac{\text{Mean Sum of Squares due to Treatment (MST)}}{\text{Mean Sum of Squares due to Error (MSE)}}. \tag{8}$$

WER measures the percentage of word-level errors—substitutions S, deletions D, and insertions I-needed to convert a system's output into the reference transcript, normalized by the reference length N as shown in Eq. (9). As stated in *Hanamaki, Kirishima & Narumi (2024)*, it computes the number of errors in the sentence corrected by the model compared to the reference. Due to its ability to capture both insertions and deletions, this metric is

widely recognized as effective for evaluating grammar correction systems (*Salhab & Abu-Khzam, 2024*; *Li et al., 2025*).

$$WER = \frac{S + D + I}{N}. \tag{9}$$

CER follows the same principle at the character level, counting character substitutions S_c, deletions D_c and insertions I_c over the total reference characters M as shown in Eq. (10) from *Hanamaki, Kirishima & Narumi (2024)*. This kind of evaluation makes it useful for morphologically rich languages or very short texts. It is beneficial for identifying errors at the character level, which can be critical for detecting fine spelling variations and minor correction inaccuracies. Smaller scores indicate closer fidelity to the reference.

$$CER = \frac{S_c + D_c + I_c}{M}. \tag{10}$$

The Levenshtein Distance (LD) between two sentences s and t is the smallest number of character-level insertions I, deletions D, or substitutions S needed to turn sentence s into sentence t as shown in Eq. (11). It measures how different two strings are from each other. This metric aids GEC systems in assessing the similarity of sentences, where minor word variations must be identified and fixed as outlined by *Naziri & Zeinali (2024)* and *Mehta et al. (2021)*.

$$\mathrm{LD}(s, t) = min_{\substack{\text{all edit sequences} \\ \text{transforming } s \to t}} (I + D + S). \tag{11}$$

Finally, Fleiss' Kappa is an inter-rater reliability metric that measures the degree of agreement between two or more raters. It judges n subjects independently, through a scale consisting of q categories as referenced by *Moons & Vandervieren (2023)* and *Falotico & Quatto (2015)*. This metric indicates which LLM achieved the highest agreement among all raters and in which criteria (*i.e.*, fluency, grammar correction, *etc.*). $\bar{P}$ means the mean of the overall observed proportion of agreement and P_e is the expected proportion of agreement by chance as shown in Eq. (12).

$$\kappa = \frac{\bar{P} - P_e}{1 - P_e}. \tag{12}$$

DISCUSSION

In this study, we evaluated two main strategies for Arabic GEC with explanations: prompting techniques (*i.e.*, zero-shot and few-shot prompting) and fine-tuning of pretrained models (GPT-4o, Gemini, and Llama).

Automatic evaluation

Automatic evaluation metrics are necessary to evaluate the performance of LLMs in the GEC task.

Similarity metrics

In order to measure the correction accuracy of our models' corrected sentences, we used the following similarity metrics: BLEU, GLEU, ROUGE, Cosine Similarity, and CLEME.

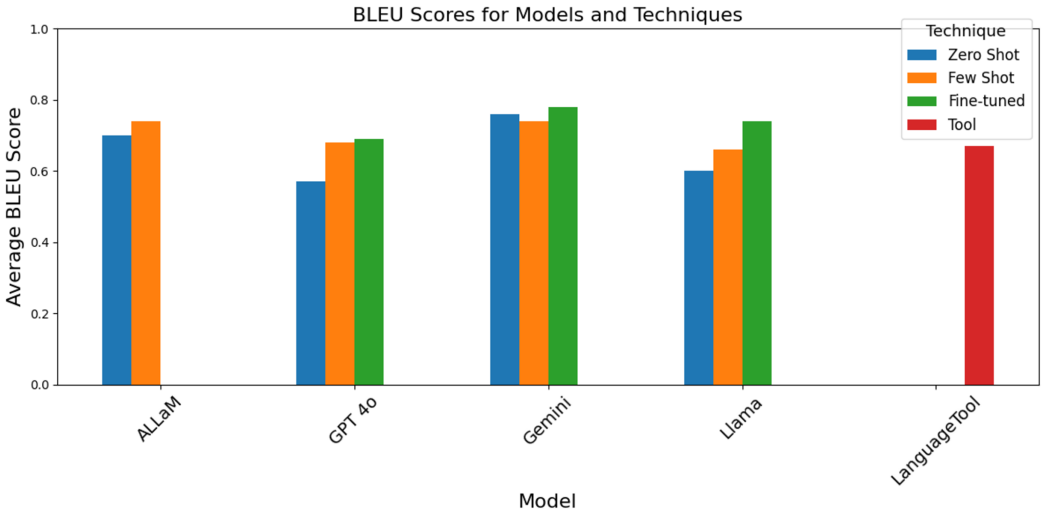

Figure 12 **BLEU scores grouped by the model and technique.**
Full-size DOI: 10.7717/peerj-cs.3486/fig-12

The BLEU evaluation, compared the corrected sentences produced by our models to the gold-standard (baseline) sentences by checking how closely the model's correction fits to the human edits. As illustrated in Fig. 12, the fine-tuned version of GPT-4o outperformed the other four models in addition to LanguageTool, achieving a BLEU score of 78%. It was closely followed by the fine-tuned Llama and ALLaM (few-shot), both with BLEU scores of 74%. Also, we note that both the fine-tuning and the few-shot techniques improved the GEC of Gemini where the BLEU score have increased from 57% (zero-shot) to 69% and 68% respectively. Additionally, LanguageTool's performance is on par with fine-tuned Gemini and Llama when prompting techniques are used.

Notably, the few-shot prompting consistently improved over zero-shot baselines, though the magnitude varied by model. This can be noticed as well in the fine-tuned versions of the models which implies that with dedicated training, the models can be improved. However, GPT-4o using zero-shot prompting outperformed GPT-4o using few-shot prompting, unlike other models. We noticed that around 250 records resulted from GPT-4o using few-shot prompting that yielded generic rejection messages as shown in Fig. 13.

GPT-4o users have observed this behavior and reported it in OpenAI community chats. According to OpenAI's documentation, when using structured outputs with user-generated input, the model may occasionally refuse to fulfill the request for unknown safety reasons. The zero-shot setting, without customized prompts, leads to more consistent corrections and higher GPT-4o scores when using prompting techniques. Table 1 shows the full BLEU scores for the models and techniques. Better quality is indicated by higher BLEU scores, which measure how similar the corrected sentence generated by the model is to the reference sentence. That is, a higher BLEU score indicates that the model's output is closer to the reference output in GEC. This means that the model made fewer mistakes and produced more accurate and grammatically correct results.

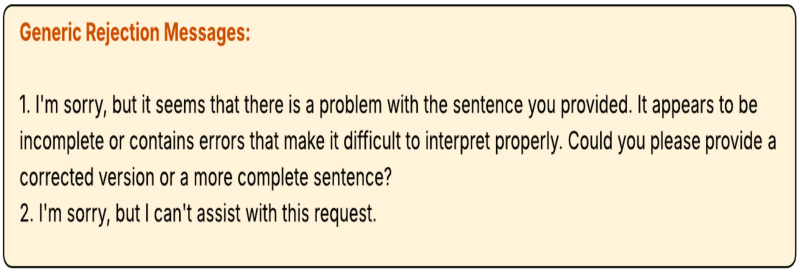

Figure 13 **Generic rejection messages.** Full-size DOI: 10.7717/peerj-cs.3486/fig-13

Table 1 **BLEU scores for all models and techniques.**

Model name	Technique	Average score
ALLaM	Zero shot	0.70
	Few shot	0.74
Gemini	Fine-tuned	0.69
	Zero shot	0.57
	Few shot	0.68
GPT-4o	Fine-tuned	0.78
	Zero shot	0.76
	Few shot	0.74
Llama	Fine-tuned	0.74
	Zero shot	0.60
	Few shot	0.66
LanguageTool	NA	0.67

As shown in Table 2, the fine-tuning technique improved the average Cosine Similarity scores to 84% for GPT-4o, 75% for Llama, and 72% for Gemini. Also, it is notable that our fine-tuned models outperformed LanguageTool. Remarkably, ALLaM, which is a pretrained model derived from Llama, reached 75% when given a few prompt examples, whereas our own fine-tuned Llama hit 75% outright in a zero-shot setting. As a result, our model matches ALLaM's best performance without relying on additional examples, indicating that parameter-level tuning is just as effective as prompt-based adaptation and possibly even more effective with additional training. In contrast to the BLEU evaluation, the original and cleaned Gemini fine-tuned outputs yielded the same cosine score because embedding models do not map empty inputs to zero vectors. Instead, they assign a learned neutral embedding (often near the centroid of the embedding space), so including those empty cases leaves the average Cosine Similarity essentially unchanged.

Figure 14 reveals the clear, gradual improvement in model performance as we moved from generic to more task-focused training. Introducing just a few illustrative examples *via* few-shot prompting yielded a slight boost over the zero-shot baseline, and fine-tuning each model to correct the sentences and explain the corrections drove the highest scores. This pattern underscores the value of progressively more targeted training—first by example, then by direct adaptation—for specialized language processing tasks.

Table 2 Cosine similarity scores for all models and techniques.

Model name	Technique	Average score
ALLaM	Zero shot	0.71
	Few shot	0.75
Gemini	Fine-tuned	0.72
	Zero shot	0.59
	Few shot	0.69
GPT-4o	Fine-tuned	0.84
	Zero shot	0.81
	Few shot	0.81
Llama	Fine-tuned	0.75
	Zero shot	0.63
	Few shot	0.68
LanguageTool	NA	0.67

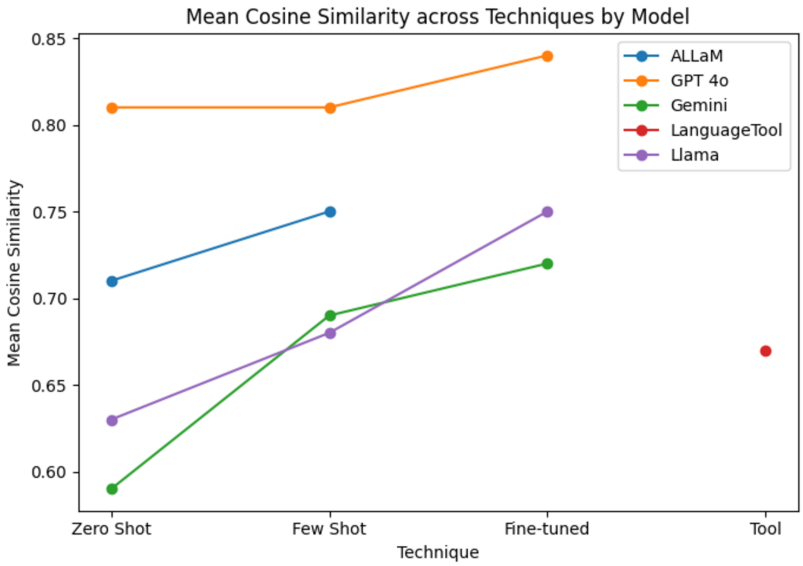

Figure 14 Cosine similarity scores grouped by the model and technique.
Full-size DOI: 10.7717/peerj-cs.3486/fig-14

Figure 15 and Table 3 indicate that the fine-tuned GPT-4o delivered the highest sentence-level overlap of any model, technique, and even the existing tool. LanguageTool's score is similar to both ALLaM and Gemini when prompting techniques are used. Few-shot prompting also resulted in consistent gains for both ALLam and Llama, reflecting the benefit of even a small number of examples. In contrast, Gemini's original fine-tuned version produced an abnormally low average due to empty or mismatched outputs, but cleaning those records raised its GLEU from just 10% to 51%, bringing it back in line with the other methods.

The fine-tuned version of GPT-4o exhibited the strongest lexical overlap with a ROUGE score of 74%, with its zero-shot variant nearly matching it, while few shot prompting

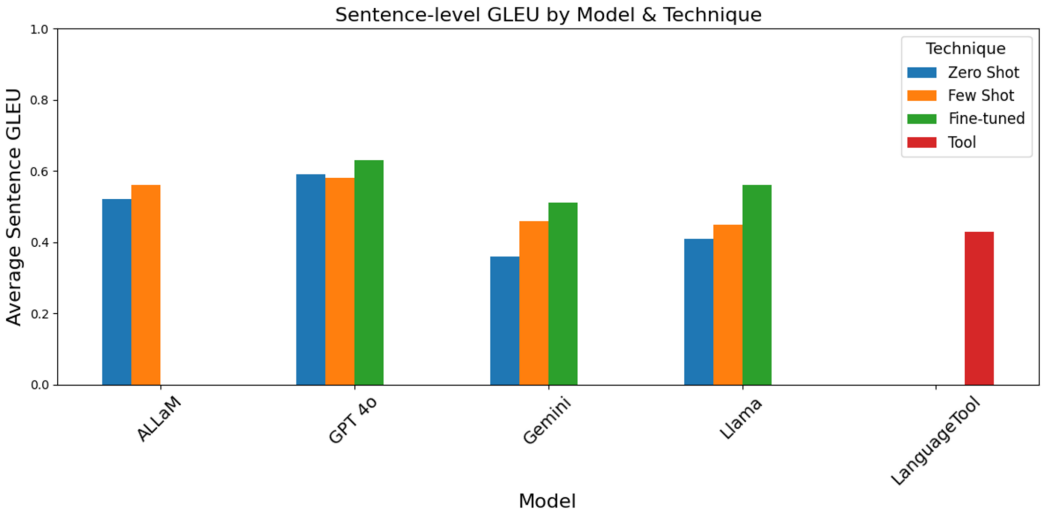

Figure 15 **GLEU scores grouped by the model and technique.**
Full-size DOI: 10.7717/peerj-cs.3486/fig-15

Table 3 **GLEU scores for all models and techniques.**

Model name	Technique	Average score
ALLaM	Zero shot	0.52
	Few shot	0.56
Gemini	Fine-tuned	0.51
	Zero shot	0.36
	Few shot	0.46
GPT-4o	Fine-tuned	0.63
	Zero shot	0.59
	Few shot	0.58
Llama	Fine-tuned	0.56
	Zero shot	0.41
	Few shot	0.45
LanguageTool	NA	0.43

slightly boosted ALLaM and Llama scores to be 71% and 65%, respectively, as shown in Table 4. The score of Gemini's cleaned fine-tuning version was 70%, bringing it on par with ALLaM and Llama few-shot prompting and greatly closing the gap from GPT-4o. Overall, the observations show that fine-tuning generally improved ROUGE scores across models.

As shown in Table 5, in all the models, CLEME results revealed that the fine-tuning technique consistently produced the strongest error correction performance. Particularly, the fine-tuned GPT-4o achieved the highest $F_{0.5}$ score of 45% and accuracy 72% by substantially increasing precision from 26% to 40% while maintaining higher recall 84%. Comparatively, zero-shot prompting achieved very high recall 95% for both GPT-4o and Gemini but struggled with precision 26% and 10%, respectively. This combined result

Table 4 ROUGE scores for all models and techniques.

Model name	Technique	Average sScore
ALLaM	Zero shot	0.68
	Few shot	0.71
Gemini	Fine-tuned	0.70
	Zero shot	0.56
	Few shot	0.63
GPT-4o	Fine-tuned	0.74
	Zero shot	0.74
	Few shot	0.71
Llama	Fine-tuned	0.71
	Zero shot	0.60
	Few shot	0.65
LanguageTool	NA	0.62

Table 5 CLEME results for all models and techniques.

Model name	Technique	No. of sample	$F_{0.5}$	Accuracy	Precision	Recall
ALLaM	Zero shot	2,000	0.26	0.60	0.23	0.72
	Few shot	2,000	0.28	0.61	0.25	0.61
Gemini	Fine-tuned	379	0.25	0.62	0.22	0.68
	Zero shot	1,919	0.14	0.53	0.11	0.95
	Few shot	1,998	0.18	0.55	0.15	0.93
GPT-4o	Fine-tuned	1,999	0.45	0.72	0.40	0.84
	Zero shot	1,999	0.31	0.63	0.26	0.95
	Few shot	1,995	0.29	0.61	0.25	0.94
Llama	Fine-tuned	1,999	0.29	0.61	0.25	0.61
	Zero shot	1,995	0.12	0.52	0.10	0.51
	Few shot	1,994	0.12	0.54	0.10	0.32
LanguageTool	NA	2,000	0.07	0.51	0.06	0.28

indicates that the models identified most errors but introduced many false edits (false negatives).

After a few illustrative examples were provided in few-shot prompting, the models yielding modest precision gains where ALLam rising from 23% to 25% and Gemini's from 11% to 15%, at the expense of slightly lower recall, resulting in small $F_{0.5}$ improvements. Filtering out empty records for Gemini's fine-tuned runs raised its $F_{0.5}$ from 13% to 25%. Among the fine-tuned models, Llama's precision jumped and its $F_{0.5}$ score more than doubled 12% to 29%. ALLam also benefited, which indicated that training on a focused task delivers the greatest overall gains. Although LanguageTool achieved a similar accuracy range with Llama and Gemini, surprisingly, it has the lowest $F_{0.5}$ and precision. Overall, we can conclude that fine-tuning matters since it raises the $F_{0.5}$ for all models significantly above zero- and few-shot performance, up to 45% for GPT-4o, and

Table 6 Levenshtein distance results for all models and techniques.

Model name	Technique	Average distance	Average ratio
ALLaM	Zero shot	18.73	0.85
	Few shot	13.48	0.88
Gemini	Fine-tuned	19.28	0.85
	Zero shot	27.47	0.78
	Few shot	18.98	0.84
GPT-4o	Fine-tuned	18.89	0.85
	Zero shot	11.43	0.89
	Few shot	14.65	0.86
Llama	Fine-tuned	13.48	0.88
	Zero shot	43.82	0.78
	Few shot	18.96	0.84
LanguageTool	NA	13.31	0.88

provided the strongest and most consistent improvements in error correction quality. Higher $F_{0.5}$ numbers means producing few false corrections while still capturing a good portion of real errors.

Error rate analysis

Levenshtein distance is the minimum number of character insertions, deletions, or substitutions needed to turn one string into another. Levenshtein distance is usually read as the number of edits, which is why we normalized it to a similarity ratio from 0 to 1 as shown in Table 6. The number of edits per sentence by itself can be misleading; what matters is the sentence's length and what proportion of each sentence is changing. For example, in regard to GPT-4o's fine-tuned model, on average, each sentence required around 19 character edits to match the reference; it is hard to decide if that is a good or bad number. However, its ratio shows that 85% of characters were correctly placed which indicates that the model outputs preserved most of the reference's character sequence, which means high semantic similarity. Overall results showed ratios above 80% as shown in Fig. 16.

Better corrections mean lower WER and CER which indicate that fewer errors remain. It's notable that the WER and CER results lined up well with the other evaluation metrics used. GPT-4o's fine-tuned model achieved an average WER of 23% and average CER of 11%. On average, then, only about 23% of the words, and 11% of the characters were erroneous, which is excellent for a grammar correction task. In second place come ALLaM zero-shot prompting and fine-tuned Llama, which both hit averages of 35% for WER and 10% for CER-in other words, 65% word accuracy and 85% character accuracy. Thus, overall, we can say that if a model has a WER of $<=25\%$ and a CER of $<=10\%$, that means it produces strong corrections, which is the case with GPT-4o's fine-tuned model, Llama's fine-tuned model and ALLaM's few-shot prompting technique, as shown in Table 7.

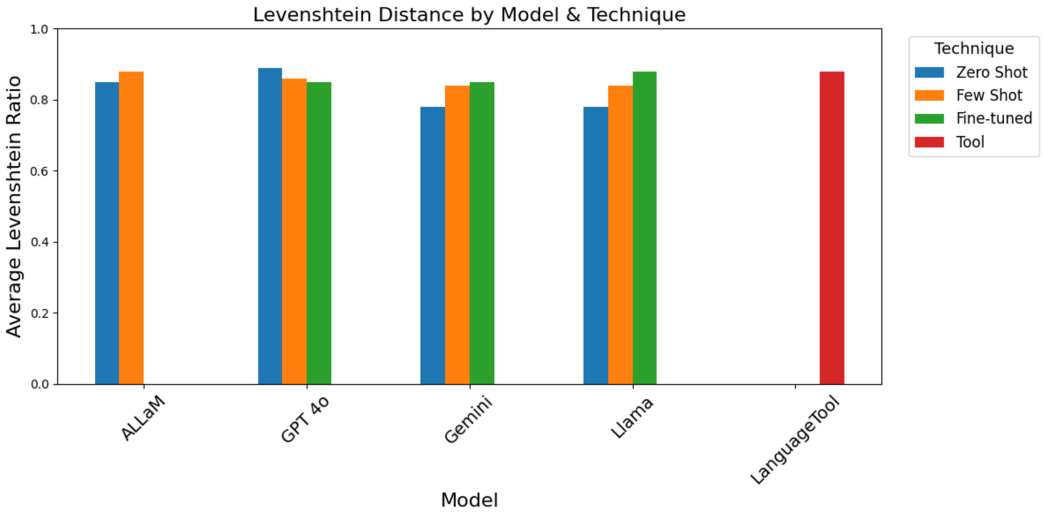

Figure 16 **Levenshtein distance results grouped by the model and technique.**
Full-size DOI: 10.7717/peerj-cs.3486/fig-16

Table 7 **Word and character error rates (WER and CER) for all models and techniques.**

Model name	Technique	Average WER	Average CER
ALLaM	Zero shot	0.39	0.16
	Few shot	0.35	0.10
Gemini	Fine-tuned	0.33	0.15
	Zero shot	0.52	0.25
	Few shot	0.44	0.15
GPT-4o	Fine-tuned	0.23	0.11
	Zero shot	0.35	0.11
	Few shot	0.37	0.14
Llama	Fine-tuned	0.35	0.10
	Zero shot	0.58	0.35
	Few shot	0.42	0.15
LanguageTool	NA	0.45	0.12

Human evaluation

In order to evaluate whether explanations of grammatical errors are effective in helping Arabic learners understand corrections, we randomly selected a sample of the generated outputs from the best performing LLMs to be evaluated, which are fine-tuned GPT-4o, fine-tuned Gemini, fine-tuned Llama, and ALLaM using few-shot prompting. Four native speakers and experts were recruited as annotators to evaluate the predicted corrected sentences and explanations based on the criteria shown in Tables 8 and 9. The following elements show the structure of the sheet provided for the annotators for evaluation: sentence no., erroneous sentence, corrected sentence, erroneous words, errors explanations, and the following to be rated from one to three: grammatical correctness (GC), fluency (F), meaning preservation (MP), clarity of explanation (CE), usefulness for learning (UL), and accuracy (ACC).

Table 8 Human evaluation criteria for Arabic GEC.

Corrected sentence evaluation	
Criterion	**Description and Scale**
Grammatical Correctness (GC) (*Östling et al., 2023*)	1: Incorrect (major errors remain or new errors introduced)
	2: Partially correct (minor errors remain)
	3: Fully correct (native-like)
Fluency/Naturalness (F) (*Östling et al., 2023*)	1: Awkward/unreadable
	2: Understandable but slightly unnatural
	3: Smooth and natural Arabic
Meaning Preservation (MP) (*Östling et al., 2023*)	1: Meaning changed significantly
	2: Meaning mostly preserved with minor distortions
	3: Meaning fully preserved

Table 9 Human evaluation criteria for Arabic GECE.

Explanation evaluation	
Criterion	**Description and scale**
Clarity of Explanation (CE) (*Kaneko & Okazaki, 2023*)	1: Hard to understand or ambiguous
	2: Understandable with some effort
	3: Very clear and concise
Usefulness for Learning (UL) (*Kaneko & Okazaki, 2023*)	1: Not helpful (doesn't guide correction)
	2: Moderately helpful
	3: Very helpful and actionable
Accuracy (ACC) (*Kaneko & Okazaki, 2023*)	1: Incorrect explanation or misleading
	2: Partially correct explanation
	3: Fully accurate explanation

As shown in Table 10, the fine-tuned GPT-4o model consistently outperforms the other models across all dimensions, achieving the highest scores in the six categories: GC, F, MP, CE, UL, and ACC. Furthermore, ALLaM using the few-shot prompting technique demonstrates competitive performance with GPT-4o in GC, F, and MP surpassing Gemini and Llama. However, it performs significantly worse in terms of CE, UL and ACC, demonstrating its limitations in explaining the corrections. Both fine-tuned Gemini and Llama achieve balanced results. Gemini shows relatively low performance in grammar correction, but produces more comprehensible explanations. In contrast, Llama achieves better grammar correction results than Gemini but slightly weaker explanation results. These human evaluation results demonstrate that the fine-tuned GPT-4o outperforms other models, confirming its potential as a reliable supportive learning tool for educational purposes.

Furthermore, to ensure the objectivity of the human evaluation, we employed Fleiss' Kappa k which calculates the level of agreement among multiple raters. Table 11 shows Fleiss' Kappa values which indicate fair agreement among the raters across the models.

Table 10 Human evaluation results.

Model name	Technique	GC	F	MP	CE	UL	ACC
ALLaM	Few-shot	2.64	2.70	2.77	1.54	1.54	1.48
Gemini	Fine-tuned	1.76	1.66	1.82	2.36	2.25	2.26
GPT-4o	Fine-tuned	2.67	2.70	2.78	2.64	2.53	2.58
Llama	Fine-tuned	2.40	2.43	2.50	2.17	2.08	2.08

Table 11 Fleiss' kappa values *k*.

Categories	GC	F	MP	CE	UL	ACC
k	0.27	0.28	0.29	0.26	0.22	0.26

Table 12 ANOVA test among techniques.

Technique	F-statistic	*p*-value
Zero shot	18.63	7.98727616053634E−07
Few shot	142.53	3.5607348542138E−14
Fine-tuning	101.2583784	1.4297085278784E−09

ANOVA

ANOVA was performed to examine whether there were significant differences in performance between the four language models (ALLaM, Gemini, GPT-4o, and Llama) in various evaluation metrics (BLEU, GLEU, CS, ROUGE, LD, WER, and CER). when using the zero-shot technique, we obtained an F-value of 18.63 and a *p*-value of 7.99E−07, shown in Table 12, which is much lower than the critical F-value of 2.66. This indicated that there were statistically significant differences between the models scores. Likewise, the few-shot technique resulted in an F-value of 142.53 with a *p*-value of 3.56E−14, which is much smaller than the critical F-value of 2.66, meaning a significant difference. Similarly, the fine-tuning technique had an F-value of 101.26 with a *p*-value of 1.43E−09, which is significantly smaller than the critical F-value of 2.99, again indicating a strongly significant difference in performance between the models.

In short, the ANOVA for the three techniques, zero-shot, few-shot, and fine-tuning, revealed varying levels of significance. As shown above, the results indicated that all the techniques exhibited significant changes in performance among the models.

Figure 17 presents a better performance visualization of the evaluation metrics under the three adaptation techniques: fine-tuned, few-shot prompting, and zero-shot prompting in additional to LanguageTool. Radial placements closer to the center imply poorer performance, and those toward the outer edge show better outcomes. GPT-4o's polygon consistently covers the biggest region in all three charts, indicating its supremacy in similarity measurements and low error rates. ALLaM, using few-shot prompting, slightly outperformed Llama in similarity and error metrics, while maintaining Llama's exact results when fine-tuned. Gemini trailed behind but closed the gap on error rate, particularly when zero-shot prompting was used. Moreover, the visualization

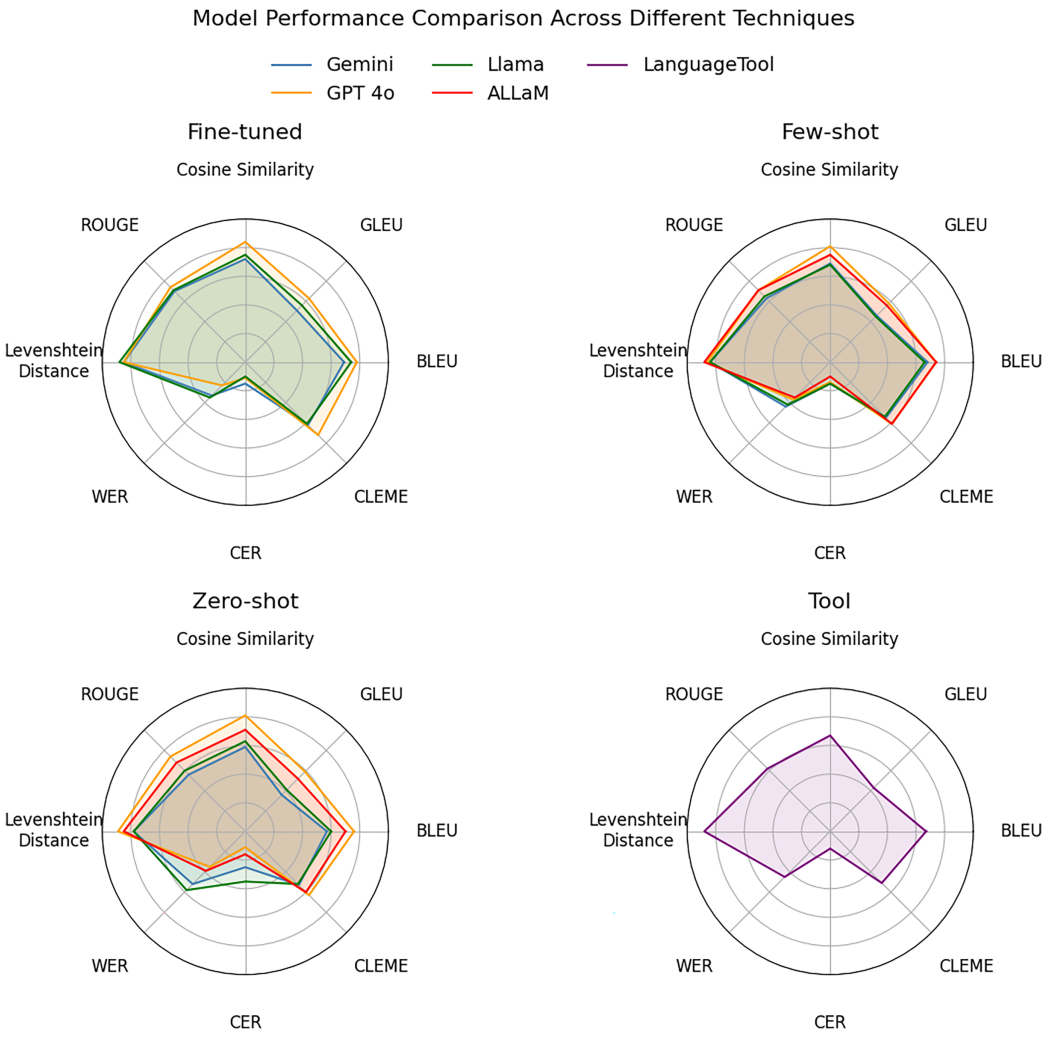

Figure 17 **Radar chart for each technique.** Full-size DOI: 10.7717/peerj-cs.3486/fig-17

illustrates more clearly that LanguageTool achieves scores comparable to Gemini and Llama, but its performance remains lower than GPT-4o. Figure 17 clearly displays GPT-4o's leading edge and the relative strengths and trade-offs of each model under varied techniques.

CONCLUSIONS

Concluding remarks

This work demonstrated that cutting-edge LLMs can potentially be efficiently tailored to solve the two challenges of correcting Arabic grammatical errors and producing clear, educationally relevant explanations. We conducted methodical experiments employing four LLMs: GPT-4o, Gemini, Llama and ALLaM and adapted two techniques, prompting techniques such as zero-shot and few-shot, and fine-tuning to answer the research questions. We also compared the performance of the selected LLMs with an existing AI-based tool called LanguageTool. On top of that, we conducted human evaluation to

examine the LLMs performance in correcting Arabic grammatical errors and explaining them.

When fine-tuned, GPT-4o obtained the highest average WER of 23% and CER of 11%, outperforming all other models in addition to LanguageTool. When zero-shot prompting was emploed, in terms of Levenshtein Distance, GPT-4o scored 11.43, which was the smallest amount of modification required to get the baseline sentence. Similarly, when fine-tuned, Llama achieved better scores across all assessment measures. ALLaM, which is an LLM developed on top of Llama, obtained scores identical to those of Llama's fine-tuned version when few-shot prompting was applied. Gemini's zero-shot prompting had the lowest score across all performance metrics. Also, LanguageTool performance was comparable to Llama and Gemini. Few-shot prompting outperformed fine-tuning on Cosine Similarity, GLEU, ROUGE, and WER but not on Levenshtein Distance or the CLEME measure.

All the techniques used, whether prompting or fine-tuning, led to significant performance changes, according to statistical analysis using one-way ANOVA, where all F-values exceeded critical levels at $p < 0.005$. According to these tests, the advantages we observed are notable and not the result of chance fluctuations. Taken together, our findings suggest that LLMs are strong tools that can explain Arabic grammatical correction and perform other NLP tasks. Prompting techniques and fine-tuning can unlock the full potential of models for Arabic, which is a difficult and under-resourced language.

Limitations

Despite the valuable findings, our study has several limitations. There are few publicly available Arabic corpora containing grammatically erroneous sentences with explicit mistake annotations and paired gold-standard corrections. Augmenting the sample size and training data may improve the robustness and reliability of the findings. Another limitation is the scarcity of Arabic-specific LLMs designed for answering inquiries or offering interpretations. Furthermore, we discovered that certain models, like Gemini, would occasionally fail to make any correction, resulting in empty records or "unable to respond" errors. Finally, the assessment of Arabic grammar correction and explanation employed several evaluation metrics that were not specifically designed for the Arabic language; a more accurate and more significant evaluation would be possible if metrics were developed or modified to account for Arabic's distinct morphology and syntax.

Future work

Future research should utilize more expansive and varied Arabic corpora alongside annotation schemes that incorporate error types, facilitating multitask learning and the creation of error-specific prompts. Furthermore, additional human assessments covering other aspects, particularly those from teachers and language learners, will provide a more comprehensive understanding of the quality of the explanation. Our techniques can be validated in real-world teaching scenarios by integrating the LLMs into real-world educational tools, especially those targeted at non-native speakers. Additionally, we can explore how different prompt designs and languages can affect the LLMs performance.

Finally, we can improve our understanding of the model performance by creating new evaluation metrics specifically designed to evaluate the precision and clarity of Arabic GECE.

Ethical considerations

Throughout this study, all datasets and models were sourced from publicly available academic resources. For closed-source LLMs, we use official APIs, so there are no ethical concerns about data ownership or access. The datasets contain no personally sensitive information. Some limitations remain, however: the data may not be representative of all types of grammatical errors, which may bias model performance towards certain patterns, and the generated explanations may sometimes be inaccurate or misleading. Therefore, we recommend using the system as a support along with human judgment rather than as a standalone replacement.

REPRODUCIBILITY

This article evaluates the performance of LLMs in correcting and explaining Arabic grammatical errors. In addition, the article explores different techniques such as fine-tuning, zero-shot prompting, and few-shot prompting techniques. Also, we used two datasets, a Manual Arabic Spelling Error Correction *corpus* and the Hugging Face Arabic GEC dataset. Additional information can be found in the README file at GitHub (https://github.com/kousar-Mohi/Evaluating-LLMs-Arabic-Grammar-Error-Corrections-and-Explanations).

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received no funding for this work.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Kousar Mohi conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Imtiaz Ahmad conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the article, investigation, Methodology, Supervision, and approved the final draft.
- Sa'ed Abed conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the article, investigation, Methodology, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The data and code are available at GitHub and Zenodo:

- https://github.com/kousar-Mohi/Evaluating-LLMs-Arabic-Grammar-Error-Corrections-and-Explanations.

- Mohi, K. (2025). Evaluating LLMs Arabic Grammar Error Corrections and Explanations [Data set]. In PeerJ (Version 1). Zenodo. https://doi.org/10.5281/zenodo.17737799.

REFERENCES

Alhafni B, Inoue G, Khairallah C, Habash N. 2023. Advancements in Arabic grammatical error detection and correction: an empirical investigation. ArXiv DOI 10.48550/arXiv.2305.14734.

Allingham JU, Ren J, Dusenberry MW, Gu X, Cui Y, Tran D, Liu JZ, Lakshminarayanan B. 2023. A simple zero-shot prompt weighting technique to improve prompt ensembling in text-image models. In: *International Conference on Machine Learning*. PMLR, 547–568.

AlOyaynaa S, Kotb Y. 2023. Arabic grammatical error detection using transformers-based pretrained language models. *ITM Web of Conferences* **56**:4009 DOI 10.1051/itmconf/20235604009.

Anil R, Borgeaud S, Alayrac J-B, Yu J, Soricut R, Schalkwyk J, Dai AM, Hauth A, Millican K, Silver D, Johnson M, Antonoglou I, Schrittwieser J, Glaese A, Chen J, Pitler E, Lillicrap T, Lazaridou A, Firat O, Molloy J, Isard M, Barham PR, Hennigan T, Lee B, Viola F, Reynolds M, Xu Y, Doherty R, Collins E, Meyer C, Rutherford E, Moreira E, Ayoub K, Goel M, Krawczyk J, Du C, Chi E, Cheng H-T, Ni E, Shah P, Kane P, Chan B, Faruqui M, Severyn A, Lin H, Li Y, Cheng Y, Ittycheriah A, Mahdieh M, Chen M, Sun P, Tran D, Bagri S, Lakshminarayanan B, Liu J, Orban A, Güra F, Zhou H, Song X, Boffy A, Ganapathy H, Zheng S, Choe H, Weisz G, Zhu T, Lu Y, Gopal S, Kahn J, Kula M, Pitman J, Shah R, Taropa E, Al Merey M, Baeuml M, Chen Z, El Shafey L, Zhang Y, Sercinoglu O, Tucker G, Piqueras E, Krikun M, Barr I, Savinov N, Danihelka I, Roelofs B, White A, Andreassen A, von Glehn T, Yagati L, Kazemi M, Gonzalez L, Khalman M, Sygnowski J, Frechette A, Smith C, Culp L, Proleev L, Luan Y, Chen X, Lottes J, Schucher N, Lebron F, Rrustemi A, Clay N, Crone P, Kocisky T, Zhao J, Perz B, Yu D, Howard H, Bloniarz A, Rae JW, Lu H, Sifre L, Maggioni M, Alcober F, Garrette D, Barnes M, Thakoor S, Austin J, Barth-Maron G, Wong W, Joshi R, Chaabouni R, Fatiha D, Ahuja A, Tomar GS, Senter E, Chadwick M, Kornakov I, Attaluri N, Iturrate I, Liu R, Li Y, Cogan S, Chen J, Jia C, Gu C, Zhang Q, Grimstad J, Hartman AJ, Garcia X, Pillai TS, Devlin J, Laskin M, de Las Casas D, Valter D, Tao C, Blanco L, Puigdomènech Badia A, Reitter D, Chen M, Brennan J, Rivera C, Brin S, Iqbal S, Surita G, Labanowski J, Rao A, Winkler S, Parisotto E, Gu Y, Olszewska K, Addanki R, Miech A, Louis A, Teplyashin D, Brown G, Catt E, Balaguer J, Xiang J, Wang P, Ashwood Z, Briukhov A, Webson A, Ganapathy S, Sanghavi S, Kannan A, Chang M-W, Stjerngren A, Djolonga J, Sun Y, Bapna A, Aitchison M, Pejman P, Michalewski H, Yu T, Wang C, Love J, Ahn J, Bloxwich D, Han K, Humphreys P, Sellam T, Bradbury J, Godbole V, Samangooei S, Damoc B, Kaskasoli A, Abbas Z, et al. 2023. Gemini: a family of highly capable multimodal models. ArXiv DOI 10.48550/arXiv.2312.11805.

Bari MS, Alnumay Y, Alzahrani NA, Alotaibi NM, Alyahya HA, AlRashed S, Mirza FA, Alsubaie SZ, Alahmed HA, Alabduljabbar G, Alkhathran R, Almushayqih Y, Alnajim R, Alsubaihi S, Al Mansour M, Alrubaian M, Alammari A, Alawami Z, Al-Thubaity A, Abdelali

A, Kuriakose J, Abujabal A, Al-Twairesh N, Alowisheq A, Khan H. 2024. ALLaM: large language models for Arabic and English. ArXiv DOI 10.48550/arXiv.2407.15390.

Bryant C, Yuan Z, Qorib MR, Cao H, Ng HT, Briscoe T. 2023. Grammatical error correction: a survey of the state of the art. *Computational Linguistics* **49(3)**:643–701 DOI 10.1162/coli_a_00478.

Chen B, Zhang Z, Langrené N, Zhu S. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. ArXiv DOI 10.48550/arXiv.2310.14735.

Creutz M. 2024. Correcting challenging Finnish learner texts with Claude, GPT-3.5 and GPT-4 large language models. In: *Workshop on Noisy and User-generated Text*. Association for Computational Linguistics (ACL), 1–10.

Dang J, Ahmadian A, Marchisio K, Kreutzer J, Üstün A, Hooker S. 2024. RLHF can speak many languages: unlocking multilingual preference optimization for LLMs. ArXiv DOI 10.48550/arXiv.2407.02552.

Davis C, Caines A, Andersen Ø, Taslimipoor S, Yannakoudakis H, Yuan Z, Bryant C, Rei M, Buttery P. 2024. Prompting open-source and commercial language models for grammatical error correction of English learner text. ArXiv DOI 10.48550/arXiv.2401.07702.

Dasopang G. 2025. The Analysis of General Communication in English Language Teaching. *Explora: Journal of English Language Teaching & Education* **11(1)** DOI 10.51622/explora.v11i1.2717.

Falotico R, Quatto P. 2015. Fleiss' kappa statistic without paradoxes. *Quality & Quantity* **49(2)**:463–470 DOI 10.1007/s11135-014-0003-1.

Fan L, Li L, Ma Z, Lee S, Yu H, Hemphill L. 2024. A bibliometric review of large language models research from 2017 to 2023. *ACM Transactions on Intelligent Systems and Technology* **15(5)**:1–25 DOI 10.1145/3664930.

Fei Y, Cui L, Yang S, Lam W, Lan Z, Shi S. 2023. Enhancing grammatical error correction systems with explanations. ArXiv DOI 10.48550/arXiv.2305.15676.

Georgiev P, Lei VI, Burnell R, Bai L, Gulati A, Tanzer G, Vincent D, Pan Z, Wang S, Mariooryad S, Ding Y, Geng X, Alcober F, Frostig R, Omernick M, Walker L, Paduraru C, Sorokin C, Tacchetti A, Gaffney C, Daruki S, Sercinoglu O, Gleicher Z, Love J, Voigtlaender P, Jain R, Surita G, Mohamed K, Blevins R, Ahn J, Zhu T, Kawintiranon K, Firat O, Gu Y, Zhang Y, Rahtz M, Faruqui M, Clay N, Gilmer J, Co-Reyes J, Penchev I, Zhu R, Morioka N, Hui K, Haridasan K, Campos V, Mahdieh M, Guo M, Hassan S, Kilgour K, Vezer A, Cheng H-T, de Liedekerke R, Goyal S, Barham P, Strouse D, Noury S, Adler J, Sundararajan M, Vikram S, Lepikhin D, Paganini M, Garcia X, Yang F, Valter D, Trebacz M, Vodrahalli K, Asawaroengchai C, Ring R, Kalb N, Baldini Soares L, Brahma S, Steiner D, Yu T, Mentzer F, He A, Gonzalez L, Xu B, Lopez Kaufman R, El Shafey L, Oh J, Hennigan T, van den Driessche G, Odoom S, Lucic M, Roelofs B, Lall S, Marathe A, Chan B, Ontanon S, He L, Teplyashin D, Lai J, Crone P, Damoc B, Ho L, Riedel S, Lenc K, Yeh C-K, Chowdhery A, Xu Y, Kazemi M, Amid E, Petrushkina A, Swersky K, Khodaei A, Chen G, Larkin C, Pinto M, Yan G, Puigdomenech Badia A, Patil P, Hansen S, Orr D, Arnold SMR, Grimstad J, Dai A, Douglas S, Sinha R, Yadav V, Chen X, Gribovskaya E, Austin J, Zhao J, Patel K, Komarek P, Austin S, Borgeaud S, Friso L, Goyal A, Caine B, Cao K, Chung D-W, Lamm M, Barth-Maron G, Kagohara T, Olszewska K, Chen M, Shivakumar K, Agarwal R, Godhia H, Rajwar R, Snaider J, Dotiwalla X, Liu Y, Barua A, Ungureanu V, Zhang Y, Batsaikhan B-O, Wirth M, Qin J, Danihelka I, Doshi T, Chadwick M, Chen J, Jain S, Le Q, Kar A, Gurumurthy M, Li C, Sang R, Liu F, Lamprou L, Munoz R, Lintz N, Mehta H, Howard H, Reynolds M, Aroyo L, Wang Q, Blanco L, Cassirer A, Griffith J, Das D, Lee S, Sygnowski J, Fisher Z, Besley J,

Powell R, Ahmed Z, Paulus D, Reitter D, Borsos Z, Joshi R, Pope A, Hand S, Selo V, Jain V, Sethi N, Goel M, Makino T, May R, Yang Z, Schalkwyk J, Butterfield C, Hauth A, Goldin A, Hawkins W, Senter E, Brin S, et al. 2024. Gemini 1.5: unlocking multimodal understanding across millions of tokens of context. ArXiv DOI 10.48550/arXiv.2403.05530.

Ghazzawi S. 1992. *The Arabic language*. Washington, D.C.: Center for Contemporary Arabic Studies.

Ghojogh B, Ghodsi A. 2020. Attention mechanism, transformers, BERT, and GPT: tutorial and survey. *Open Science Framework, Dec.*

Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, Letman A, Mathur A, Schelten A, Vaughan A, Yang A, Fan A, Goyal A, Hartshorn A, Yang A, Mitra A, Sravankumar A, Korenev A, Hinsvark A, Rao A, Zhang A, Rodriguez A, Gregerson A, Spataru A, Roziere B, Biron B, Tang B, Chern B, Caucheteux C, Nayak C, Bi C, Marra C, McConnell C, Keller C, Touret C, Wu C, Wong C, Ferrer CC, Nikolaidis C, Allonsius D, Song D, Pintz D, Livshits D, Wyatt D, Esiobu D, Choudhary D, Mahajan D, Garcia-Olano D, Perino D, Hupkes D, Lakomkin E, AlBadawy E, Lobanova E, Dinan E, Smith EM, Radenovic F, Guzmán' F, Zhang F, Synnaeve G, Lee G, Anderson GL, Thattai G, Nail G, Mialon G, Pang G, Cucurell G, Nguyen H, Korevaar H, Xu H, Touvron H, Zarov V, Arrieta Ibarra I, Kloumann I, Misra I, Evtimov I, Zhang J, Copet J, Lee J, Geffert J, Vranes J, Park J, Mahadeokar J, Shah J, van der Linde J, Billock J, Hong J, Lee J, Fu J, Chi J, Huang J, Liu J, Wang J, Yu J, Bitton J, Spisak J, Park J, Rocca J, Johnstun J, Saxe J, Jia J, Alwala KV, Prasad K, Upasani K, Plawiak K, Li K, Heafield K, Stone K, El-Arini K, Iyer K, Malik K, Chiu K, Bhalla K, Lakhotia K, Rantala-Yeary L, van der Maaten L, Chen L, Tan L, Jenkins L, Martin L, Madaan L, Malo L, Blecher L, Landzaat L, de Oliveira L, Muzzi M, Pasupuleti M, Singh M, Paluri M, Kardas M, Tsimpoukelli M, Oldham M, Rita M, Pavlova M, Kambadur M, Lewis M, Si M, Singh MK, Hassan M, Goyal N, Torabi N, Bashlykov N, Bogoychev N, Chatterji N, Zhang N, Duchenne O, Çelebi O, Alrassy P, Zhang P, Li P, Vasic P, Weng P, Bhargava P, Dubal P, Krishnan P, Koura PS, Xu P, He Q, Dong Q, Srinivasan R, Ganapathy R, Calderer R, Cabral RS, Stojnic R, Raileanu R, Maheswari R, Girdhar R, Patel R, Sauvestre R, Polidoro R, Sumbaly R, Taylor R, Silva R, Hou R, Wang R, Hosseini S, Chennabasappa S, Singh S, Bell S, Kim SS, Edunov S, Nie S, Narang S, Raparthy S, Shen S, Wan S, Bhosale S, Zhang S, Vandenhende S, Batra S, Whitman S, Sootla S, Collot S, Gururangan S, Borodinsky S, Herman T, Fowler T, Sheasha T, Georgiou T, Scialom T, Speckbacher T, et al. 2024. The Llama 3 herd of models. ArXiv DOI 10.48550/arXiv.2407.21783.

Hanamaki S, Kirishima N, Narumi S. 2024. Assessing audio hallucination in large multimodal models. DOI 10.31219/osf.io/f8ra7.

Hurst A, Lerer A, Goucher AP, Perelman A, Welihinda A, Radford A, Borzunov A, Carney A, Chow A, Paino A, Renzin A, Passos AT, Christakis A, Kamali A, Moyer A, Tam A, Crookes A, Tootoonchian A, Kumar A, Karpathy A, Mishchenko A, Cann A, Kondrich A, Tulloch A, Jiang A, Pelisse A, Woodford A, Gosalia A, Nayak A, Oliver A, Ghorbani B, Leimberger B, Wang B, Hoover B, Samic B, Guarraci B, Eastman B, Lugaresi C, Li C, Barette C, Voss C, Ding C, Zhang C, Beaumont C, Hallacy C, Koch C, Gibson C, Choi C, Hesse C, Wei C, Kappler D, Levin D, Levy D, Farhi D, Mely D, Sasaki D, Tsipras D, Li D, Nguyen DP, Findlay D, Wong E, Asdar E, Proehl E, Yang E, Peterson E, Sigler E, Brevdo E, Khorasani F, Zhang F, Oden G, Salmon G, Salman H, Bao H, Schmidt H, Ren H, Chung HW, Kivlichan I, O'Connell I, Osband I, Okuyucu I, Kostrikov I, Kanitscheider I, Coxon J, Crooks J, Lennon J, Park J, Teplitz J, Wei J, Wolfe J, Chen J, Harris J, Weng J, Tang J, Jang J, Ward J, McKay J, Kim JW, Gross J, Kaplan J, Jiao J, Lee J, Zhuang J, Fricke K, Karthik K, Hsu K, Howe K, Luther K, Kai L, Itow L, Chen L, Guy L, Mamitsuka L, Weng L, Ouyang L, Feuvrier L,

Kondraciuk L, Kaiser L, Doshi L, Aflak M, Simens M, Thompson M, Dukhan M, Zhang M, Litwin M, Zeng M, Johnson M, Gupta M, Glaese M, Janner M, Petrov M, Wu M, Fradin M, Pokrass M, Castro MOT, Pavlov M, Khan M, Bavarian M, Yesildal M, Gimelshein N, Staudacher N, Stathas N, Tezak N, Kudige N, Bundick N, Nachum O, Boiko O, Murk O, Godement O, Campbell-Moore O, Pronin P, Tillet P, Lim R, Troll R, Lin R, Lopes RG, Puri R, Miyara R, Leike R, Gaubert R, Zamani R, Honsby R, Ramchandani R, Carmichael R, Nigmatullin R, Cheu R, Culver S, Gray S, Grove S, Metzger S, Jain S, Zhao S, Wu S, Xia ST, Phene S, Papay S, Coffey S, Lee S, Lee S, Hall S, Balaji S, Broda T, Stramer T, Gogineni T, Sanders T, Cunninghman T, Dimson T, Raoux T, Zheng T, Kim C, Underwood T, Heywood T, Qi V, Monaco V, Fomenko V, Zheng W, Zhou W, Zaremba W, Patil Y, Qian Y, Kim Y, et al. 2024. GPT-4O system card. ArXiv DOI 10.48550/arXiv.2410.21276.

Ingólfsdóttir SL, Ragnarsson PO, Jónsson HP, Símonarson HB, orsteinsson V, Snæbjarnarson V. 2023. Byte-level grammatical error correction using synthetic and curated corpora. ArXiv DOI 10.48550/arXiv.2305.17906.

Kaneko M, Okazaki N. 2023. Controlled generation with prompt insertion for natural language explanations in grammatical error correction. ArXiv DOI 10.48550/arXiv.2309.11439.

Keselman HJ, Huberty CJ, Lix LM, Olejnik S, Cribbie RA, Donahue B, Kowalchuk RK, Lowman LL, Petoskey MD, Keselman JC, Levin JR. 1998. Statistical practices of educational researchers: an analysis of their ANOVA, MANOVA, and ANCOVA analyses. *Review of Educational Research* **68(3)**:350–386 DOI 10.3102/00346543068003350.

Kobayashi M, Mita M, Komachi M. 2024a. Large language models are state-of-the-art evaluator for grammatical error correction. ArXiv DOI 10.48550/arXiv.2403.17540.

Kobayashi M, Mita M, Komachi M. 2024b. Revisiting meta-evaluation for grammatical error correction. ArXiv DOI 10.48550/arXiv.2403.02674.

Kwon SY, Bhatia G, Nagoudi EMB, Abdul-Mageed M. 2023. Beyond English: evaluating LLMs for Arabic grammatical error correction. ArXiv DOI 10.48550/arXiv.2312.08400.

La Cava L, Tagarelli A. 2025. Open models, closed minds? On agents capabilities in mimicking human personalities through open large language models. *Proceedings of the AAAI Conference on Artificial Intelligence* **39(2)**:1355–1363 DOI 10.1609/aaai.v39i2.32125.

Lai W, Mesgar M, Fraser A. 2024. LLMs beyond English: scaling the multilingual capability of LLMs with cross-lingual feedback. ArXiv DOI 10.48550/arXiv.2406.01771.

Li Y, Qiao X, Zhao X, Zhao H, Tang W, Zhang M, Yang H. 2025. Large language model should understand pinyin for chinese asr error correction. In: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Piscataway: IEEE, 1–5.

Li Y, Qin S, Ye J, Ma S, Li Y, Qin L, Hu X, Jiang W, Zheng H-T, Yu PS. 2024. Rethinking the roles of large language models in Chinese grammatical error correction. ArXiv DOI 10.48550/arXiv.2402.11420.

Lin C-Y. 2004. Rouge: a package for automatic evaluation of summaries. In: *Text Summarization Branches Out*. Cham: Springer, 74–81.

Liu Q, Chen N, Sakai T, Wu X-M. 2024. Once: boosting content-based recommendation with both open-and closed-source large language models. In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. New York: ACM, 452–461.

Loem M, Kaneko M, Takase S, Okazaki N. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: a study on performance and controllability in prompt-based methods. ArXiv DOI 10.48550/arXiv.2305.18156.

Marvin G, Hellen N, Jjingo D, Nakatumba-Nabende J. 2023. Prompt engineering in large language models. In: *International Conference on Data Intelligence and Cognitive Informatics*. Cham: Springer, 387–402.

Mathav Raj J, Kushala VM, Warrier H, Gupta Y. 2024. Fine tuning LLM for enterprise: practical guidelines and recommendations. ArXiv DOI 10.48550/arXiv.2404.10779.

Mehta A, Salgond V, Satra D, Sharma N. 2021. Spell correction and suggestion using Levenshtein distance. *International Journal of Engineering Research & Technology* **8(8)**:1977–1981.

Miller GA. 1951. Language and communication. *McGraw-Hill*. DOI 10.1037/11135-000.

Moons F, Vandervieren E. 2023. Measuring agreement among several raters classifying subjects into one-or-more (hierarchical) nominal categories. A generalisation of Fleiss' kappa. ArXiv DOI 10.48550/arXiv.2303.12502.

Mothe J. 2024. Shaping the future of endangered and low-resource languages—our role in the age of LLMs: a keynote at ECIR 2024. In: *ACM SIGIR Forum*. Vol. 58. New York, NY, USA: ACM, 1–13.

Napoles C, Sakaguchi K, Post M, Tetreault J. 2015. Ground truth for grammatical error correction metrics. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 588–593.

Naziri A, Zeinali H. 2024. A comprehensive approach to misspelling correction with BERT and Levenshtein distance. ArXiv DOI 10.48550/arXiv.2407.17383.

Östling R, Gillholm K, Kurfalı M, Mattson M, Wirén M. 2023. Evaluation of really good grammatical error correction. ArXiv DOI 10.48550/arXiv.2308.08982.

Pajak K, Pajak D. 2022. Multilingual fine-tuning for Grammatical Error Correction. *Expert Systems with Applications* **200(8)**:116948 DOI 10.1016/j.eswa.2022.116948.

Papineni K, Roukos S, Ward T, Zhu W-J. 2002. BLEU: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318.

Park C, Koo S, Kim G, Lim H. 2024. Towards harnessing the most of ChatGPT for Korean grammatical error correction. *Applied Sciences* **14(8)**:3195 DOI 10.3390/app14083195.

Raheja V, Alikaniotis D, Kulkarni V, Alhafni B, Kumar D. 2024. mEDIT: multilingual text editing via instruction tuning. ArXiv DOI 10.48550/arXiv.2402.16472.

Raiaan MAK, Mukta MSH, Fatema K, Fahad NM, Sakib S, Mim MMJ, Ahmad J, Ali ME, Azam S. 2024. A review on large language models: architectures, applications, taxonomies, open issues and challenges. *IEEE Access* **12(8)**:26839–26874 DOI 10.1109/access.2024.3365742.

Saiegh-Haddad E, Henkin-Roitfarb R. 2014. The structure of Arabic language and orthography. In: *Handbook of Arabic Literacy: Insights and Perspectives*. Cham: Springer, 3–28.

Salhab M, Abu-Khzam F. 2024. AraSpell: a deep learning approach for Arabic spelling correction. ArXiv DOI 10.48550/arXiv.2405.06981.

Sallam M, Mousa D. 2024. Evaluating ChatGPT performance in Arabic dialects: a comparative study showing defects in responding to Jordanian and Tunisian general health prompts. *Mesopotamian Journal of Artificial Intelligence in Healthcare* **2024**:1–7 DOI 10.58496/mjaih/2024/001.

Saty AA, Aouragh SL, Bouzoubaa K. 2023. A new spell-checking approach based on the user profile. *International Journal of Computing and Digital Systems* **13(1)**:1.

Selim N. 2018. Arabic, grammar, and teaching: an Islamic historical perspective. *International Journal of Islamic Thought* **13**:80–89.

Shen Y, Heacock L, Elias J, Hentel KD, Reig B, Shih G, Moy L. 2023. ChatGPT and other large language models are double-edged swords. *Radiology* **307(2)**:e230163 DOI 10.1148/radiol.230163.

Song Y, Krishna K, Bhatt R, Gimpel K, Iyyer M. 2023. GEE! Grammar error explanation with large language models. ArXiv DOI 10.48550/arXiv.2311.09517.

Tang C, Qu F, Wu Y. 2024. Ungrammatical-syntax-based in-context example selection for grammatical error correction. ArXiv DOI 10.48550/arXiv.2403.19283.

Turing AM. 2009. *Computing machinery and intelligence*. Cham: Springer.

von Schwerin M, Reichert M. 2024. A systematic comparison between open-and closed-source large language models in the context of generating GPDR-compliant data categories for processing activity records. *Future Internet* **16(12)**:459 DOI 10.3390/fi16120459.

Wang Q, Yuan Z. 2024. Assessing the efficacy of grammar error correction: a human evaluation approach in the Japanese context. ArXiv DOI 10.48550/arXiv.2402.18101.

Wei J, Bosma M, Zhao VY, Guu K, Yu AW, Lester B, Du N, Dai AM, Le QV. 2021. Finetuned language models are zero-shot learners. ArXiv DOI 10.48550/arXiv.2109.01652.

White J, Fu Q, Hays S, Sandborn M, Olea C, Gilbert H, Elnashar A, Spencer-Smith J, Schmidt DC. 2023. A prompt pattern catalog to enhance prompt engineering with ChatGPT. ArXiv DOI 10.48550/arXiv.2302.11382.

Yao B, Chen G, Zou R, Lu Y, Li J, Zhang S, Sang Y, Liu S, Hendler J, Wang D. 2024. More samples or more prompts? Exploring effective few-shot in-context learning for LLMs with in-context sampling. In: *Findings of the Association for Computational Linguistics: NAACL 2024*. Stroudsburg: ACL, 1772–1790.

Ye J, Li Y, Zhou Q, Li Y, Ma S, Zheng H-T, Shen Y. 2023. CLEME: debiasing multi-reference evaluation for grammatical error correction. ArXiv DOI 10.48550/arXiv.2305.10819.

Ye J, Qin S, Li Y, Cheng X, Qin L, Zheng H-T, Xing P, Xu Z, Cheng G, Wei Z. 2024. EXCGEC: a benchmark of edit-wise explainable Chinese grammatical error correction. ArXiv DOI 10.48550/arXiv.2407.00924.

Yu H, Yang Z, Pelrine K, Godbout JF, Rabbany R. 2023. Open, closed, or small language models for text classification? ArXiv DOI 10.48550/arXiv.2308.10092.

Zeng M, Kuang J, Qiu M, Song J, Park J. 2024. Evaluating prompting strategies for grammatical error correction based on language proficiency. ArXiv DOI 10.48550/arXiv.2402.15930.

Zhang Y, Cui L, Zhao E, Bi W, Shi S. 2023b. RobustGEC: robust grammatical error correction against subtle context perturbation. ArXiv DOI 10.48550/arXiv.2310.07299.

Zhang X, Zhang X, Yang C, Yan H, Qiu X. 2023a. Does correction remain an problem for large language models? ArXiv DOI 10.48550/arXiv.2308.01776.