Grammatical Error Correction for Low-Resource Languages: The Case of Zarma

Anonymous ACL submission

Abstract

Zarma is a Nilo-Saharan language spoken predominantly in West Africa. The limited availability of annotated data and the need for standardized orthography make grammatical error correction (GEC) particularly challenging for Zarma. This study presents a comparative analysis of GEC methods for Zarma, exploring classical GEC approaches such as rule-based methods, machine translation (MT) models, and state-of-the-art large language models (LLMs). Through rigorous evaluations, we compare the strengths and limitations of each method, assessing their effectiveness in identifying and correcting errors in Zarma texts. Our findings highlight the promising potential of both LLMs and MT models to significantly enhance GEC capabilities for low-resource languages, paving the way for developing more inclusive and robust NLP tools for African languages.

1 Introduction

GEC is an essential task in NLP that aims to improve the quality and readability of texts by correcting grammatical errors. GEC tools are important for enhancing written materials, significantly impacting educational outcomes, professional opportunities, and access to information. This is particularly relevant in under-resourced settings where limited access to academic resources and formal training can exacerbate language proficiency disparities. Additionally, communities in these settings often rely on local languages for communication and knowledge transmission. GEC tools are well-developed for high-resource languages, especially English, where extensive annotated datasets and standardized writing systems are available (Napoles et al., 2019). However, GEC presents significant challenges for low-resource languages.

Zarma, a Nilo-Saharan language spoken by over 5 million people across Niger and neighboring countries (Lewis et al., 2016), exemplifies the difficulties faced by many low-resource languages. The lack of standardized orthography and the limited availability of annotated data make it challenging to develop practical GEC tools. These challenges are not unique to Zarma but are shared by a broad class of low-resource languages, which include many African and indigenous languages worldwide. Addressing these challenges requires innovative approaches working with minimal data and non-standardized texts.

The emergence of LLMs has ushered in a new era in NLP, empowering machine learning (ML) models to understand and generate human-like text across many languages (Devlin et al., 2018; Brown et al., 2020). LLMs exhibit remarkable zero-shot and few-shot learning capabilities (Wan et al., 2023), potentially beneficial for low-resource languages. These capabilities allow LLMs to perform effectively with minimal data, making them valuable for languages with limited annotated datasets. However, the use of LLMs is also challenging in low-resource settings. They are primarily trained on data from high-resource languages, which may limit their performance when applied to low-resource languages. The lack of representative training data can lead to errors and biases, reducing their effectiveness. Additionally, the significant computational resources required to fine-tune and deploy LLMs can be a barrier in resource-constrained environments.

This study investigates the potential of LLMs and traditional models to improve GEC for Zarma by comparing conventional rule-based methods, MT-based models, and the novel application of LLMs for Zarma GEC. With the goal of studying the performance of these models on other low-resource languages, we replicate our Zarma GEC experiments with Bambara, a West African language spoken in Mali. Through a comprehensive case study, we highlight the strengths and weaknesses of each method, aiming to bridge the linguistic gap in NLP for low-resource languages like

Zarma and Bambara.

Our research addresses the following questions:

RQ1: *Do state-of-the-art LLM models outperform conventional rule-based and MT-based models on GEC for Zarma texts?*

RQ2: *What are the specific strengths and limitations of each GEC approach in low-resource settings, considering the lack of standardized orthography and limited annotated data?*

The main contributions of this paper are the following:

- A comprehensive evaluation of Zarma's three distinct GEC approaches (rule-based, MT-based, and LLM-based). Our findings show that the MT-based approach delivered the highest accuracy with a detection rate of 96.30%, a suggestion accuracy of 92.59%, and an acceptable performance in zero-shot scenarios.

- Reproduction of the experiments with additional West African languages (Bambara) to confirm replicability and broaden the study's scope beyond Zarma.

- Development and public release (upon acceptance) of models fine-tuned for GEC of the tested languages.

2 Related Work

Cissé and Sadat (2023) present an approach for spellchecking Wolof, a language primarily spoken in Senegal. Their algorithm combines a dictionary lookup with an edit distance metric Levenshtein distance algorithm (Levenshtein, 1966) to identify and correct spelling errors.

Vydrin and collaborators developed Daba, a software package for grammar and spellchecking in Manding languages (Vydrin, 2014). Their work employs a rule-based system focusing on morphological analysis, addressing the agglutinative nature of Manding languages.

Researchers have explored the use of LLMs for language-specific tasks, including GEC, demonstrating their adaptability beyond high-resource languages. For instance, a study by Palma Gomez et al. (2023) showed that the MT5 model, a multilingual transformer model pre-trained on a massive dataset of text and code, could be effectively fine-tuned for GEC in Ukrainian. Song et al. (2023) present an innovative application of LLMs—specifically GPT-4 (OpenAI, 2023)—for generating explanations for grammatical errors. Another promising approach to GEC leverages pre-trained multilingual MT models. The study by (Luhtaru et al., 2024) introduced the "No Error Left Behind" approach, which uses models like MT5 (Xue et al., 2020) and NLLB (Team et al., 2022). The fine-tuning process involves adapting the pre-trained MT model to treat error correction as a "**translation**" task, where the source language is the incorrect sentence and the target language is the corrected sentence. However, the research also identified a fundamental challenge: the trade-off between precision and recall when training with synthetic data. This finding shows the need for further research to optimize these models for low-resource GEC tasks, potentially through improved data augmentation techniques or developing specialized architectures for error correction.

3 Methods

3.1 GEC with LLMs

LLMs have significantly advanced NLP, exhibiting capabilities in multitasking, few-shot learning, and multilingual understanding. These models, extensively pre-trained on diverse datasets, demonstrate a remarkable ability to grasp nuanced aspects of language, reasoning, and context (Brown et al., 2020; Raffel et al., 2020). This section outlines our methodology for leveraging LLMs to develop a GEC tool for Zarma. The proposed GEC tool is designed to function independently of predefined grammar rules or lexicons, utilizing the models' few-shot learning capabilities for enhanced efficiency in low-resource scenarios.

3.1.1 Implementation

We employed two distinct approaches for LLM-based GEC:

Instruction and Error Explanation Fine-tuning:

This method involves embedding training data within a contextual sentence structure, enhancing the model's reasoning abilities, and facilitating effective learning from the examples. We use the following instruction prompt:

Prompt: "Zarma sentence: [*Incorrect Sentence*], Correct the zarma sentence: [*Correct Sentence*] **Error Causes:** : [*Error Cause*]." (174)

This format, particularly the error explanation component, is crucial for aiding the model's comprehension of Zarma's grammatical contexts and patterns, as demonstrated in previous research (Schick and Schütze, 2021; Wei et al., 2021).

Non-Prompt Fine-tuning Using Aligned Sentences:

This approach involves fine-tuning the model directly on parallel data of incorrect and correct Zarma sentences without explicit prompts. This leverages the model's ability to learn the implicit mapping between incorrect and correct forms.

3.2 GEC with MT Models

Exploring MT models for GEC represents a promising avenue for addressing GEC challenges in low-resource languages like Zarma. MT models, particularly those pre-trained on multilingual datasets, offer strong capabilities for understanding and processing text across diverse linguistic frameworks.

3.2.1 Implementation

We chose the M2M100 model (Fan et al., 2020) for its demonstrated ability to translate between many languages, indicating its potential to capture cross-linguistic patterns relevant to GEC. To adapt M2M100 for Zarma GEC, we fine-tuned it on a corrupted corpus designed to reflect common errors in Zarma text. This corpus was generated by applying a custom noise script, described in Section 3.4. The script introduces various errors, ensuring the training data effectively represents realistic challenges in real-world Zarma text. This corrupted corpus, paired with the original correct sentences, serves as the training data for M2M100, enabling the model to learn the mapping from incorrect to correct Zarma. Detailed training settings and evaluation metrics for this MT-based GEC approach can be found in Tables 3, 5, and 6.

3.3 Rule-based GEC for Zarma

As a baseline, we designed a rule-based GEC process, based on Levenshtein distance and the Bloom filter (Bloom, 1970), for Zarma (Figure 1). Additionally, we implemented a tool and API in Python. The tool—the first of its kind for Zarma, to our knowledge—is designed to cater to a wide range of users. It offers command-line and graphical user interfaces (GUI) and is much less computationally intensive than the LLM-based approach. Moreover, our results show that it provides a competitive spell correction performance compared to the LLMs- and MT-based approaches. To ensure the tool's accessibility, we plan a public release upon acceptance.

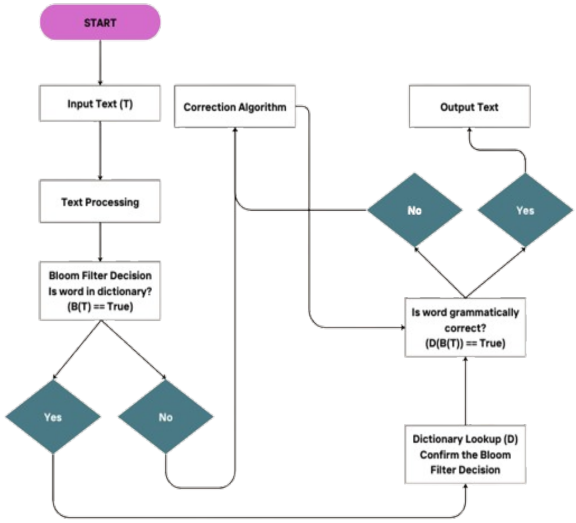

Figure 1: Rule-Based GEC tool Workflow

The GEC process for Zarma text includes a sequence of logical steps, beginning with the input text T, moving through a Bloom filter decision B, leading to a dictionary lookup outcome D, and culminating in the final correction C:

$$C(T) = \begin{cases} T & \text{if } D(B(T)) = \text{True}, \\ \text{Correct}(T) & \text{if } D(B(T)) = \text{False}. \end{cases} \tag{1}$$

The detailed process includes:

Text Processing The system begins by processing the text, where the Zarma content is segmented into words, punctuation, and spaces using regular expressions. This initial step ensures that every part of the text is ready for individual examination—a necessity given Zarma's linguistic intricacies.

Bloom Filter The Bloom filter—known for its space and time efficiency—performs probabilistic checks on whether a word might be in the dictionary. For a given word w, it uses a set of hash functions to probe various positions in a bit array. If all checked positions are flagged, w might be in the dictionary:

$$B(w) = \bigwedge_{i=1}^{k} \text{bit}[h_i(w)], \tag{2}$$

Given the extensive Zarma lexicon sourced from the Feriji Dataset (kei, 2024), this component makes the process faster and more efficient, totaling 9902 unique words.

Dictionary Lookup At the core of the system is the trie-based dictionary lookup. This setup manages Zarma's intricate word forms, confirming word accuracy after the Bloom filter. This step is crucial to ensure the text adheres to Zarma's linguistic norms and serves as a post-Bloom filter double-checking process.

Grammar Rules The system incorporates a set of Zarma grammar rules, a task made challenging by the lack of a standard writing format. This phase scrutinizes elements like consonant rules and vowel lengths, essential for keeping the text accurate to Zarma's grammatical essence despite its diverse phonology and absence of a writing standard.

Correction Algorithm When an error is detected, the correction algorithm is activated. This component leverages the Levenshtein distance L to identify the smallest number of edits—insertions, deletions, or substitutions—required to rectify the erroneous word. For illustration, given a corrupted sentence "A sindq biri," the Levenshtein algorithm's operation can be represented as:

$$\begin{gathered}\textbf{Corrupted Sentence}\text{: "A sindq biri"}\\ \downarrow\\ \textbf{LevSuggest}(\text{"}sindq\text{"})\\ \downarrow\\ \{\text{"sind", "sinda"}\}\end{gathered} \tag{3}$$

Here, "sindq" undergoes comparison with the lexicon, and "sind" and "sinda" are suggested based on their proximity in terms of minimal edit distance. In this context, $L(a, b)$ calculates the distance between the incorrect string a and a potential correct string b, considering their lengths i and j, with $1_{(a_i \neq b_j)}$ serving as an indicator function to highlight discrepancies. This mechanism ensures the algorithm's efficiency in providing appropriate corrections, thus enhancing the GEC tool's overall accuracy for Zarma text.

3.4 Data Preparation

This section details the process of gathering and preparing data to train our GEC models for Zarma. We created two distinct datasets, which were then combined to form a larger, more comprehensive training dataset.

3.4.1 Synthetic Data

The first dataset was generated using a custom corruption script applied to the Feriji Dataset. The script was designed to introduce realistic typographical and grammatical errors into Zarma sentences. To ensure the introduction of plausible errors that mimic human mistakes, the corruption process is guided by a defined formula:

$$CSZ(S, V, C) = N \circ SZ(S, V, C) \tag{4}$$

Where **CSZ** denotes the corrupted Zarma sentence resulting from applying the noise function **N** to the original sentence, **SZ**. Here, **(S, V, C)** represents a sentence's subject, verb, and complement, respectively. The noise function **(N)**, consists of four operations: deletions δ, insertions μ, substitutions σ, and transpositions τ. The script was meticulously crafted to ensure the introduced noise did not inadvertently create another grammatically valid Zarma sentence.

For example, consider **SZ** as ***"A go koy fuo"*** which translates to ***"He is going home"***. Applying the noise function N, such as a substitution σ that changes "go" to "ga," results in the sentence **CSZ**: ***"A ga koy fuo"***. While "ga" is a valid Zarma word, in this context, it alters the meaning of the sentence to ***"He will go home"***, introducing a grammatical error.

$$\begin{aligned}&\textbf{SZ} = \text{"}A\,go\,koy\,fuo\text{"}\\ &\textbf{N}(\textbf{SZ}) = \textbf{SZ} \xrightarrow{\sigma(\text{"}go\text{"}\rightarrow\text{"}ga\text{"})} \text{"}A\,ga\,koy\,fuo\text{"}\\ &\textbf{CSZ} = \text{"}A\,ga\,koy\,fuo\text{"}\\ &\textbf{SZ} \xrightarrow{N} \textbf{Correct CSZ}\end{aligned} \tag{5}$$

Furthermore, the script creates four corrupted variants for each correct sentence in the dataset, enriching the learning material with diverse linguistic nuances. The dataset is illustrated in Table 1.

3.4.2 Human-Annotated Data

The second dataset—referred to as the Gold Data 2—was curated by human annotators. Annotators introduced grammatical and logical errors into Zarma sentences and provided justifications for each alteration. This dataset structure exposes the models to a broader array of error types— precisely logical and grammatical errors—and corresponding corrections, thereby enhancing the model's capacity to generalize and accurately correct unseen

Correct Sentence	Corrupted Sentences
Sintina gaa Irikoy na beena da ganda taka.	Sintina gaa Irikog na beena da ganda taka. Sintina gaa Irikoy na been da ganda taka. Sintina aga Irikoy na beena da ganda taka. Sintina gaa Irikoy na beena ea ganda taka.

Table 1: Snapshot of The Synthetic Data

texts in Zarma, particularly in zero-shot or few-shot learning scenarios.

Incorrect Sentence	Correction	Error Explanation
Souba, Ay koy Niamey	Souba, Ay ga koy Niamey	"**Souba**" means tomorrow, and therefore the tense must be in the future using the future tense marker "**ga**" after the subject "**Ay**."
...	...	...

Table 2: Snapshot of the Gold Data

4 Experiment

We selected three models for training based on their demonstrated proficiency in multilingual tasks and their aptitude for few-shot learning: Gemma (Team et al., 2024), MT5, and M2M100. The training was conducted on Google Cloud, utilizing a Deep Learning Virtual Machine. Due to resource constraints, QLoRA quantization (Dettmers et al., 2023) was applied to Gemma, while smaller variants of MT5 and M2M100 were used. For LLM training, the combined dataset was structured as shown in 3.1.1, while for the MT method, we adopted an MT task-specific format, using aligned sentences without error explanations. The detailed training settings for each model are presented in Table 3.

5 Results & Comparative Analysis

To assess the effectiveness of each GEC method for Zarma GEC, we tested them in two ways: fixing words and logic/grammar (LG) problems in sentences. We used two sets of 100 sentences each from the Feriji Dataset. Sample A was for fixing single words, and Sample B was for fixing LG errors. We also used a third set, Sample C, with 27 sentences to test how well the models could handle new words and LG problems they had not seen before (zero-shot testing). For Sample A, we used our script to add 71 typos—common mistakes people make when writing Zarma—-and for Sample B, our Zarma annotators added grammar errors —logic—and sentence structure. This gave us a good mix of mistakes to test the models.

5.1 Word-Level Correction Metrics

To compare the methods for fixing single words, we looked at these things:

- **Detection**: How many errors did the method find and correct?
- **Suggestion**: How many corrections suggested were correct?
- **F1-Score**: A score that combines detection and suggestion, giving us a balanced view of how well the method worked

As shown in Table 4, the rule-based method achieved a perfect score in this test. It found all the errors and suggested the proper correction every time. However, this was a controlled test with common typos. In real-world situations, the rule-based method might not work well if it encounters new words or errors it has not seen before. The M2M100 model did the best among the models, with high scores for detection—100%, suggestion—91%, and F1-score—0.95. This model learns from many different languages, which helps it understand and fix errors in Zarma even though it is a low-resource language.

5.2 LG Improvement Metrics

For Sample B, five Zarma speakers rated how well each method fixed LG errors using a scale from 1 to 5. 1 means the correction was terrible, and 5 means it was perfect. We also examined how well the methods did with different error types, like verb tense errors, subject-verb agreement errors, and missing words. See Table 5.

Again, the rule-based method struggled with LG errors because it needed help understanding the context of the sentences. It could only fix problems based on its predefined rules rather than based on how the words were used in the sentence. The

Models	Parameters	Training Details			
		QLoRA	GPU Used	Lr	Loss
Gemma 2b	2 billion	Applied	NVIDIA P100	2×10^{-4}	1.2613
MT5-small	300 million	Not Applied	NVIDIA T4x2	2×10^{-5}	0.0345
M2M100	418 million	Not Applied	NVIDIA P100	2×10^{-5}	0.0214

Table 3: Training Settings for the models

Methods	Word Level Metrics		
	Detection	Suggestion	F1-Score
Rule-based	100%	100%	1.00
Gemma 2b	92%	66%	0.77
MT5-small	95%	64%	0.76
M2M100	100%	91%	0.95

Table 4: Word-Level Correction Performance Metrics

Methods	Context Level Avg(1-5)	
	Logical Errors Correction	Sentence Improvement
Rule-based	0.4	0
Gemma 2b	1	0
MT5-small	1.7	1
M2M100	3	2.5

Table 5: LG Improvement Metrics

M2M100 model performed better than the other methods, getting higher scores for fixing logical errors—3/5, and improving sentence structure—2.5/5 as shown in Table 5. This shows that learning from many languages helps MT models understand the context of sentences and make better corrections. We also noticed that the models had more trouble with some LG errors than others. For example, they were better at fixing verb tense than subject-verb agreement errors. This tells us that we need more training data with different kinds of mistakes to help the models learn how to fix them. Recent research has shown that training models on diverse error types, including synthetic errors that reflect real-world linguistic variations, can significantly enhance their performance in LG correction tasks (Napoles et al., 2017).

5.3 Zero-Shot Performance

In the zero-shot test (Sample C), we looked at how well the models could handle new words and LG errors they had not seen before. Table 6 shows the results.

As expected, the rule-based method could not suggest corrections for new words because it did not have them in its dictionary. The M2M100 model again performed best, showing its ability to generalize from its multilingual training data to handle new Zarma words and LG errors it had never seen before—with an accuracy of 96.30% for detection, 92.59% for suggestion, 2.4/5 for logical error correction, and 2.3/5 for sentence improvements. These results strongly suggest that MT models, especially those trained on diverse, multilingual data, hold significant potential for improving GEC in low-resource languages. This aligns with recent research highlighting the effectiveness of MT models for cross-lingual transfer learning in various NLP tasks (Conneau et al., 2018). However, more research is needed to explore the optimal training strategies and data requirements for further maximizing the performance of MT models in low-resource GEC scenarios. To validate the reproducibility and robustness of our methods, we conducted further experiments with the Bambara language, which belongs to a different linguistic family. The results of this experiment are detailed in Section A of the appendix.

6 Discussion

Our comparative analysis, detailed in Tables 4, 5 and 6, indicates that the M2M100 model—leveraging the MT approach—yielded the most promising results among the tested models. This was particularly evident in its superior suggestion accuracy and ability to handle zero-shot words effectively. This strong performance is likely attributable to M2M100's design, which leverages a balanced approach to translation tasks across multiple languages, making it adept at understanding and correcting errors within a multilingual context.

6.1 Methods' Strengths and Limitations

6.1.1 Rule-Based Methods

Rule-based approaches are highly effective in addressing predictable and previously encountered error patterns. Our controlled tests showed that these methods achieved perfect detection and sug-

Methods	Evaluation Metrics			
	Word Level		Context Level Avg(1-5)	
	Detection	Suggestion	Logical Errors Correction	Sent.Improvement
Rule-based	100%	81.48%	1	0
Gemma 2b	92.59%	40.74%	0.5	0
MT5-small	92.59%	48.15%	1.2	0.6
M2M100	96.30%	92.59%	2.4	2.3

Table 6: Correction Performance Metrics (Zero-Shot Dataset)

gestion scores. Their strength lies in their reliance on a comprehensive set of predefined rules and a detailed target language dictionary. However, this reliance also presents a significant limitation—inflexibility. Rule-based methods struggle to handle new or unexpected errors, which is common in dynamic language use. This limitation becomes particularly pronounced in zero-shot scenarios, where the system encounters words or grammatical constructions not included in its defined patterns or dictionary. This inherent dependency on extensive and carefully curated linguistic resources restricts the scalability of rule-based methods, especially for low-resource languages like Zarma, where such resources are often limited or incomplete, as highlighted in (Scannell, 2007).

6.1.2 LLMs

The LLMs in our experiments—Gemma 2b and MT5—demonstrated adequate performance in controlled and zero-shot scenarios. A key strength of LLMs is their capacity to understand context, enabling them to generate corrections based on broader textual cues rather than relying solely on direct matches to known errors. However, LLM performance significantly depends on the diversity and quality of the training data. A critical limitation is that most pre-existing LLMs are primarily trained on data from high-resource languages, mainly Western languages. Consequently, their applicability to African languages like Zarma is often hindered by a need for more representative training examples (Bender et al., 2021). This results in lower suggestion accuracy and difficulties in effectively handling the unique linguistic complexities of these languages. Moreover, training LLMs necessitates substantial computational resources, posing a significant barrier in resource-constrained environments.

6.1.3 MT Models

In our case, the MT approach—using the M2M100 model—demonstrated exceptional performance in zero-shot scenarios, surpassing both rule-based methods and LLMs. The strength of this approach lies in the ability of these models to leverage multilingual translation mechanisms, effectively adapting to the nuances of diverse languages through their exposure to vast and varied training datasets. This characteristic makes MT models particularly suitable for GEC in low-resource languages, as they can infer corrections from patterns learned across multiple languages. This aligns with research highlighting the effectiveness of MT models for cross-lingual transfer learning in various NLP tasks (Conneau et al., 2018). However, a significant challenge in utilizing MT models for GEC in low-resource languages is the frequent scarcity of high-quality, parallel corpora for training. With sufficient data, the models may generate more accurate and contextually appropriate corrections (Tiedemann, 2020). Moreover, despite their strengths, MT models require fine-tuning and continuous updating to maintain their accuracy and relevancy, especially as language use evolves.

6.2 Recommendations for Improvement

To further enhance GEC systems for Zarma and other low-resource languages, we propose the following recommendations:

Increasing Dataset Size: Expanding datasets with more varied examples, including those representing zero-shot scenarios, can substantially improve model training, especially for LLMs and MT models. As noted by (Scannell, 2007), limited data availability is a significant challenge in developing resources for low-resource languages. Increasing the training data's volume and diversity could enable models to handle a broader range of linguistic variations and rare scenarios, enhancing overall accuracy and robustness.

Hybrid Approaches: Our findings suggest that combining the strengths of rule-based systems with the adaptability of LLMs and the robustness of MT models could yield a more powerful GEC system. Such a hybrid approach could utilize rule-based systems to handle standard, predictable errors and leverage machine learning models to address more complex, context-dependent errors. This approach aligns with research highlighting the effectiveness of integrating multilingual resources to improve language processing capabilities across different systems (Tiedemann, 2020).

Continuous Learning: Implementing mechanisms for models to learn continuously from new input and user-generated corrections can contribute to progressively improving their accuracy and adaptability. This aligns with the findings of (Bender et al., 2021), who emphasize the importance of continuous model updating and reevaluation to maintain their effectiveness, especially in rapidly evolving language use patterns.

7 Potential Applications

Our team visited Niamey to present the work to the local Zarma community and inquire about their feedback. The discussions provided valuable insights into potential applications for our GEC tool and broader language models.

Content Creation One critical comment we received was the potential use of our model to translate coding content and general educational materials for enthusiasts and students. There is a growing interest in technology and programming within the community, but a significant language barrier exists. By translating coding tutorials, textbooks, and other educational resources into Zarma, our model can help overcome this challenge, making these materials more accessible and encouraging non-western language speakers to engage in tech-related fields. Additionally, the GEC tool can be used to translate and produce general educational content in Zarma. This includes textbooks, instructional materials, and online courses across various subjects.

Communication Tools Integrating the GEC tool into communication platforms can facilitate seamless interaction in Zarma for users with varying levels of language proficiency. In addition, tools such as messaging apps and email clients could incorporate the GEC tool to provide real-time corrections, helping users learn and adopt proper language usage.

Cultural Preservation Some feedback highlighted the importance of maintaining accurate written records of folklore, oral histories, and traditional knowledge. The GEC tool can support these efforts by providing a reliable tool for transcribing and publishing grammatically accurate texts.

8 Conclusion and Future Work

This research demonstrates the potential of LLMs and MT models to address the critical need for effective GEC tools in low-resource languages, explicitly focusing on Zarma. While previous studies have shown the effectiveness of these models in high-resource settings, their application to Zarma presents unique challenges due to data scarcity and a need for established benchmarks.

To overcome these challenges, we implemented a novel approach that combines three key elements: (1) a custom corruption script to generate synthetic training data, effectively addressing the limited availability of annotated Zarma text; (2) a human-annotated "Gold Data" set incorporating expert knowledge of Zarma grammar, providing a valuable benchmark for evaluating model performance on complex errors; and (3) the adaptation of advanced LLMs and MT models, such as Gemma or M2M100, for the specific task of Zarma GEC.

Our findings reveal the potential of LLM and MT models—particularly M2M100—in achieving high accuracy in Zarma GEC, even in zero-shot scenarios. This highlights their ability to leverage cross-lingual patterns learned from diverse, multilingual datasets to improve GEC in under-resourced languages. This research comprehensively evaluates different GEC approaches for Zarma and establishes a baseline for future work in this area. Further exploration of hybrid approaches that combine rule-based methods with the adaptability of LLMs and the robustness of MT models holds promise for creating even more effective GEC tools. Additionally, incorporating continuous learning mechanisms can enable these tools to adapt to evolving language use and user feedback, enhancing their accuracy and relevance.

Limitations

Despite the promising results obtained from our experiments, we observed several limitations. Firstly, while effective in controlled scenarios with known error patterns, the rule-based approach exhibited significant challenges when faced with unseen patterns. This is due to its dependence on predefined rules and extensive dictionaries, which could be better for languages with limited resources and writing standards.

Secondly, the LLMs we used—including Gemma 2b and MT5-small—also faced several challenges. One primary limitation was the models' reliance on the diversity and quality of their training data. These models— primarily built for high-resource languages– -may need more nuances and contextual understanding for low-resource languages like Zarma. In addition, the models are resource-hungry, which is a disadvantage in resource-constrained environments typical of low-resource language communities.

Thirdly, a significant challenge is the need for more quality annotated data for Zarma and other low-resource languages. While we created a synthetic dataset and a smaller human-annotated "Gold Data" set to mitigate this, these datasets may still not capture the full linguistic error patterns in language use. The reliance on synthetic data—though helpful for experiments—may introduce biases that do not entirely reflect real-world usage. Therefore, the generalizability of our findings is–constrained by the quality and representativeness of the available training data.

Lastly, the zero-shot performance highlighted challenges in achieving a good score across the approaches regarding LG errors and sentence improvements. The approaches showed variability in handling different LG errors, with some types being more challenging than others. This suggests that our current methodologies require further refinement and additional data to handle the wide range of errors.

References

2024. Feriji: A french-zarma parallel corpus, glossary & translator. *Preprint*, arXiv:2406.05888.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Burton H. Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Thierno Ibrahima Cissé and Fatiha Sadat. 2023. Automatic spell checker and correction for under-represented spoken languages: Case study on Wolof. In *Proceedings of the Fourth workshop on Resources for African Indigenous Languages (RAIL 2023)*, pages 1–10, Dubrovnik, Croatia. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denkowski, and Hervé Jégou. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *Preprint*, arXiv:2010.11125.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

M. Paul Lewis, Gary F. Simons, and Charles D. Fennig. 2016. Ethnologue: Languages of the world.

Agnes Luhtaru, Elizaveta Korotkova, and Mark Fishel. 2024. No error left behind: Multilingual grammatical error correction with pre-trained translation models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1209–1222, St. Julian's, Malta. Association for Computational Linguistics.

Courtney Napoles, Maria Nădejde, and Joel Tetreault. 2019. Enabling robust grammatical error correction in new domains: Data sets, metrics, and analyses. *Transactions of the Association for Computational Linguistics*, 7:551–566.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. The most frequent error correction scenarios: Evidence from l2 english writings. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 208–218, Copenhagen, Denmark. Association for Computational Linguistics.

OpenAI. 2023. Chatgpt4: Optimizing language models for dialogue. https://openai.com/chatgpt4.

Frank Palma Gomez, Alla Rozovskaya, and Dan Roth. 2023. A low-resource approach to the grammatical error correction of Ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 114–120, Dubrovnik, Croatia. Association for Computational Linguistics.

Colin Raffel et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21.

Kevin Scannell. 2007. The challenges of developing computational resources for minority languages. In *Proceedings of the ACL-2007 Workshop on Computational Approaches to Semitic Languages*, pages 65–72.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze questions for few shot text classification and natural language inference. In *Proceedings of EACL*.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2023. Gee! grammar error explanation with large language models. *arXiv preprint arXiv:2311.09517*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Pier Giuseppe Sessa, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Jörg Tiedemann. 2020. Tatoeba: Building and exploiting a multilingual database of sentence equivalents. In *Language Resources and Evaluation Conference*, pages 2020–2024.

Valentin Vydrin. 2014. Daba: a model and tools for manding corpora. *ResearchGate*.

Valentin Vydrin, Jean-Jacques Meric, Kirill Maslinsky, Andrij Rovenchak, Allahsera Auguste Tapo, Sebastien Diarra, Christopher Homan, Marco Zampieri, and Michael Leventhal. 2022. Machine learning dataset development for manding languages. urlhttps://github.com/robotsmali-ai/datasets.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514, Toronto, Canada. Association for Computational Linguistics.

Jason Wei et al. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

A Further Experiment with other Languages

After obtaining promising results for Zarma using LLM and MT-based approaches, we conducted further experiments to validate the reproducibility of these methods—using the M2M100 and Gemma models. We selected the Bambara language for this experiment because it belongs to a different linguistic family, allowing us to evaluate the performance of the approaches on a language outside the Nilo-Saharan family. We utilized the Bayelemabaga dataset (Vydrin et al., 2022) for Bambara. The same data preparation process described in the methodology section was followed; however, we excluded any human-annotated data to focus solely on word-level GEC performance. The results are presented in Table 7.

Methods	**Word Level Metrics**		
	Detection	**Suggestion**	**F1-Score**
Gemma 2b	87.45%	52.91%	0.6594
M2M100	94.64%	68.18%	0.7926

Table 7: Word-Level Correction Performance Metrics for Bambara

The Bambara experiment demonstrated that the MT-based approach outperformed the LLMs-based one regarding word-level correction metrics. The MT-based approach achieved a detection rate of 94.64% and a suggestion accuracy of 68.18%. In contrast, the LLMs-based approach detected 87.45% of errors and suggested corrections with 52.91% accuracy. The promising results from the Bambara experiment highlight the potential of both LLMs and MT models to improve GEC for low-resource languages significantly. However, they also emphasize the necessity for continued expanding and diversifying training datasets.

B Errors Being Addressed

In this section, we explain the types of errors our grammatical error correction (GEC) methods address. We categorize the errors into two main types: word-level correction (spellchecking) and context-level correction. The context-level correction is further divided into grammar errors, logical errors, and sentence improvement. Below, we define each error type and provide examples to illustrate them.

B.1 Word-Level Correction (Spellchecking)

Word-Level correction involves identifying and correcting typographical errors in individual words. These errors are usually due to misspellings, incorrect usage of characters, or typographical mistakes.

- **Example:**
  - **Incorrect**: *Sintina gaa Irikog na beena da ganda taka.*
  - **Correct**: *Sintina gaa Irikoy na beena da ganda taka.*

B.2 Context-Level Correction

Context-level correction involves errors that go beyond individual words and affect the overall structure and meaning of the sentence. We categorize these errors into logical errors and sentence improvement.

B.2.1 Logical Errors

Logical errors include incorrect verb conjugations, subject-verb agreement issues, incorrect use of grammatical markers, and logical inconsistencies within the sentence. These errors affect the grammatical correctness and logical coherence of the sentence.

- **Example:**
  - **Incorrect**: *Souba, Ay koy Niamey.* (The time indicator "Souba" means "tomorrow," but the verb "koy" indicates present tense.)
  - **Correct**: *Souba, Ay ga koy Niamey.* (The future tense marker "ga" matches the time indicator "Souba.")

B.2.2 Sentence Improvement

Sentence improvement involves enhancing the quality of the sentence by making it more precise, concise, or stylistically appropriate. This category addresses grammatically correct sentences that can be improved for better readability or style.

- **Example:**
  - **Original**: *I girbi honkuna i tun be.*
  - **Improved**: *I ga girbi suba.*

Figure 2: Images of the different GEC tool interfaces. The rule-based on the left and the other approaches on the right