Evaluating Robustness of LLMs to Typographical Noise in Yorùbá QA

Paul Okewunmi[1,2*] **Favour James**[1,2] **Oluwadunsin Fajemila**[1,2]
[1]ML Collective
[2]Obafemi Awolowo University
{ptokewunmi, fujames, oefajemila}@student.oauife.edu.ng

Abstract

Generative AI models are primarily accessed through chat interfaces, where user queries often contain typographical errors. While these models perform well in English, their robustness to noisy inputs in low-resource languages like Yorùbá remains underexplored. This work investigates a Yorùbá question-answering (QA) task by introducing synthetic typographical noise into clean inputs. We design a probabilistic noise injection strategy that simulates realistic human typos. In our experiments, each character in a clean sentence is independently altered, with noise levels ranging from 10% to 40%. We evaluate performance across three strong multilingual models using two complementary metrics: (1) a multilingual BERTScore to assess semantic similarity between outputs on clean and noisy inputs, and (2) an LLM-as-judge approach, where the best Yorùbá-capable model rates fluency, comprehension, and accuracy on a 1–5 scale. Results show that while English QA performance degrades gradually, Yorùbá QA suffers a sharper decline. At 40% noise, GPT-4o experiences over a 50% drop in comprehension ability, with similar declines for Gemini 2.0 Flash and Claude 3.7 Sonnet. We conclude with recommendations for noise-aware training and dedicated noisy Yorùbá benchmarks to enhance LLM robustness in low-resource settings.

Figure 1: The top conversation represents a correct response, while the bottom conversation illustrates errors due to typographical noise. The question in the bottom example contains multiple error types, which includes replacement and transposition. As a result, the model fails to understand the query and responds with confusion.

1 Introduction

Large Language Models (LLMs) have transformed the landscape of Natural Language Processing (NLP), enabling advanced reasoning and question-answering (QA) capabilities. These models perform exceptionally well in high-resource languages like English, where extensive training data and noise-handling mechanisms enhance robustness. However, their effectiveness in low-resource languages like Yorùbá remains underexplored (Inuwa-Dutse, 2025).

A key challenge affecting LLM robustness is sensitivity to input variations. Minor typographical errors, such as omitted letters or misplaced diacritics, can significantly degrade model performance. Prior research (Moradi and Samwald, 2021; Vaibhav et al., 2019) has analyzed this phenomenon in English QA tasks, revealing how slight distortions mislead models. However, little is known about its effects in Yorùbá, a tonal language heavily reliant

*Corresponding author.

Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025), pages 195–202

on diacritics to convey meaning. Misplaced or omitted diacritics can alter words entirely—e.g., "Ògún" (a deity) vs. "ogun" (war) vs. "ogún" (twenty), presenting an even greater risk of misinterpretation (Jimoh et al., 2025).

Despite the linguistic importance of diacritics, Yorùbá text is often written without them in electronic media, most often due to keyboard limitations or user habits, resulting in significant information loss (Jimoh et al., 2025). As illustrated in Figure 1, typographical distortions can lead to misinterpretations that affect model performance in QA tasks. LLMs trained predominantly on high-resource languages may struggle with these nuances, raising a critical question: **How well do LLMs handle typographical errors in Yorùbá question answering?**

Handling noisy text is crucial for real-world applications, particularly in multilingual settings. While typographical perturbations and adversarial attacks have been studied extensively in English, systematic evaluations for Yorùbá are lacking—despite the language being spoken by over 40 million people. Understanding how well LLMs handle noisy Yorùbá input is essential for improving their reliability across diverse linguistic contexts.

To address this gap, we construct a controlled Yorùbá QA dataset with synthetic typographical noise using a probabilistic noise modeling approach. Characters in clean sentences are independently altered at noise levels ranging from 10% to 40%, introducing errors such as insertions, replacements, and transpositions (swapping) based on keyboard adjacency. We also explore a variant where error types are randomly selected, incorporating leet replacements (e.g., 'e' → '3', 'o' → '0', 's' → '$')(Zhang et al., 2022). Model responses to noisy inputs are evaluated against clean text using semantic similarity metrics such as BERTScore (Zhang et al., 2020) and an LLM-as-judge evaluation framework (Zheng et al., 2023).

Our contributions are as follows:

1. We propose a probabilistic noise generation method that simulates human typographical errors in Yorùbá.

2. We systematically evaluate the impact of typographical noise on Yorùbá QA performance using GPT-4o, Gemini 2.0 Flash and Claude 3.7 Sonnet.

3. We provide insights to inform noise-aware training, develop evaluation datasets, and establish benchmarks for assessing typographical robustness in Yorùbá NLP.

2 Related Work

Given the increasing prevalence of chat-based language models facilitating text-based interaction between users and language models, several studies have explored how user-generated typographical errors influence model performance. Previous research has utilized artificially generated noisy datasets created through various simulation methodologies (Kumar et al., 2020; Cai et al., 2022). Specifically, these studies introduced noise by randomly altering a percentage of characters based on proximity within the QWERTY keyboard layout, effectively simulating typical typing errors encountered in real-world interactions.

However, much of this research has primarily concentrated on monolingual settings, predominantly English, neglecting the assessment of multilingual language models with diverse multilingual test scenarios (Moradi and Samwald, 2021; Wang et al., 2023). Consequently, investigations into textual noise have largely been restricted to English-language contexts. Despite impressive performances by large multilingual models across various tasks and languages, their effectiveness tends to diminish significantly when applied to languages other than English, particularly low-resource languages (Etxaniz et al., 2023).

Additionally, existing literature has mainly evaluated transformer-based models such as BERT, suggesting a research gap regarding larger, recently popularized language models (Cooper Stickland et al., 2023). Previous studies demonstrated the robustness of models like BERT, XLM-Roberta, and XLNet against textual noise, noting their commendable performance despite their relatively modest sizes, typically under 0.3 billion parameters. This highlights a clear distinction from contemporary LLMs, which frequently possess parameter counts in the billions, underscoring the necessity for further investigations into their resilience to noisy inputs.

This study addresses the gap between contemporary chat-based LLMs and authentic typographical errors observed in practical usage. It examines the robustness of large language models with multilingual capabilities, specifically using noisy, real-

Error	Example Sentence
None	Kí ló mu ki ẹrọ kọmpútà fi ṣe pàtàkì púpọ ní ayé òde òní?
Replacement	lí ló mu ji ẹfọ kọmpútà fi ṣe pàtàkì púpọ ní ayé òde ònk?
Insertion	Kí ló mu kiu ẹẹrọ kọmpútà fi ṣe pàtàqkì púpọ ní ayé òde òoní?
Transposition	Kí ló mu ki ẹọr kmọpútà fi ṣe pàtàkì púpọ ní ayé òde òin?
Random	Kí l0 mu k1 ẹrọ ọkmpútà fi $e pàtàkì púpọ ní ayé pde 0ní

Table 1: Yorùbá text with different error types.

world Yorùbá datasets.

3 Methodology

3.1 Typographical Error Types

To effectively replicate real-world user interactions, we focus on modifying words in ways that reflect common typing errors made during chatbot conversations with LLMs. To assess their impact, we introduce four primary categories of typographical errors using a probabilistic modeling approach:

- **Insertion Errors:** An extra character, either the same as the intended one (double typing) or an adjacent key from a QWERTY keyboard, is inserted immediately after the original character. This simulates accidental keystrokes common in rapid typing.
- **Replacement Errors:** The intended character is replaced with a neighboring key based on the QWERTY layout, mimicking mistyped characters.
- **Transposition (Swap) Errors**: Two adjacent characters swap positions, replicating common finger-slips where typists accidentally invert the order of two neighboring characters.
- **Random Errors:** A combination of insertion, replacement, transposition, and character-to-symbol substitutions (leetspeak errors, e.g., replacing 'e' with '3', 'o' with '0') is applied. This mixed-error category closely reflects real-world, unstructured typing mistakes.

These error types collectively represent realistic erros that can substantially affect the performance of language models, especially in a linguistically sensitive context such as Yorùbá question and answering tasks. Table 1 shows examples of these errors in a sentence.

3.2 Noise Injection Strategy

To precisely evaluate the impact of typographical errors, we employ a probabilistic noise injection approach. Given a clean text sequence of length N, we introduce errors at a predefined rate p, modifying a fraction of characters to simulate real-world typing mistakes.

The number of modified characters, N_e, is determined as:

$$N_e = \lfloor p \times N \rfloor$$

where p is the error rate (e.g., 10%, 20%, 40%).

For each selected character position, one of the previously described error types is applied. The error type is either predetermined (for controlled experiments) or chosen randomly for greater variability.

The noise injection process follows these steps:

1. **Text Tokenization**: The input text is split into individual characters while preserving spaces.
2. **Error Injection**: A random subset of characters, determined by N_e, is selected, and an error type is applied.
3. **Text Reconstruction**: The modified sequence is reconstructed, ensuring that spacing and word boundaries remain intact.

Since the selection of characters to be modified is performed uniformly at random, each character in the text has an equal probability of being selected for modification. The probability that a specific character x_i is selected for modification is:

$$P(x_i \text{ is modified}) = \frac{N_e}{N} = p$$

This implies that every character has an independent probability p of being altered, regardless of its position in the sequence. The overall process is further illustrated in **Algorithm 1**.

Algorithm 1 Probabilistic Typo Injection

Require: Clean text sequence $X = \{x_1, x_2, ..., x_n\}$, error rate p, predefined error mapping T, noise function $\mathcal{N}$
Ensure: Noisy text sequence $X' = \{x'_1, x'_2, ..., x'_n\}$

Compute number of typo errors:

$$N_e = \lfloor p \times n \rfloor$$

Randomly select N_e character positions:

$$P = \text{RandomSample}(\{1, 2, ..., n\}, N_e)$$

for $i \in P$ **do**
    Retrieve predefined error type T_i from mapping T
    Apply noise function $\mathcal{N}$ based on T_i:
    if T_i = Insertion **then**
        Insert an adjacent or duplicate character
    else if T_i = Replacement **then**
        Replace character with a neighboring key
    else if T_i = Transposition **then**
        Swap adjacent characters
    else if T_i = Random **then**
        Apply a mix of predefined transformations
    end if
end for
Construct noisy text X' by modifying selected positions in X
return X'

4 Experimentation

4.1 Dataset

The dataset used in this study consists of 50 curated Yorùbá QA pairs, carefully selected to ensure a balance between culturally specific questions and general knowledge inquiries. The culturally peculiar questions focus on topics rooted in Yorùbá traditions, language, and history, while the general knowledge questions cover widely known facts that are not restricted to any specific cultural context. The average question length is about 15 words.

Each question in the dataset is structured to encourage detailed responses rather than one-word answers. This design choice ensures that evaluation is not based on exact matches but rather on the LLM's ability to understand the question and generate an accurate and contextually appropriate response.

4.2 Generating Noisy Variants from Dataset

To evaluate the impact of typographical noise on Yorùbá QA, we introduce controlled noise to create variations of the clean questions in the dataset. For each question, we introduce typographical errors at predefined rates. Every question undergoes modifications corresponding to the four error types, with error rates varying from 10% to 40% in increments of 10%. This range ensures that we capture a spectrum of real-world errors, from minor typos to more severe distortions. Increasing noise beyond this threshold could result in unnatural sentences, making evaluation less meaningful.

To account for variability, we generate three distinct variations for each error type at each noise level, ensuring that different subsets of characters are affected. This results in a total of:

$$\begin{aligned} &4 \text{ (error types)} \times 4 \text{ (error rates)} \\ &\times 3 \text{ (variations per rate)} = 48 \end{aligned}$$

noisy versions per sentence. Since we have 50 sentences in our dataset, we end up with a total of:

$$50 \times 48 = 2,400$$

sentences, allowing for a diverse evaluation of model robustness.

Having multiple variations per sentence enhances evaluation depth and reliability. First, it provides a comprehensive assessment of how different types and levels of noise impact model performance. Additionally, by generating multiple variations at the same noise level, we ensure that evaluation results are not biased by a specific character selection, reducing variance and improving statistical significance. Finally, this approach closely reflects real-world typing errors, as users rarely make the same mistake in a fixed pattern.

4.3 Models

Each noisy variation of the dataset is input into the models using the same system prompt to ensure consistency across evaluations. The prompt explicitly instructs the models to limit responses to a maximum of 25 words, balancing computational efficiency with response relevance.

To enforce deterministic outputs, we set the temperature to 0, ensuring a fixed response pattern for each input. The generated responses are logged for further evaluation, enabling direct comparisons between clean and noisy input variations.

4.4 Evaluation Process

We pass the clean questions to the models, using their returned output as a gold standard for comparison. Next, we introduce typographical noise and compare the models' responses to their clean-input counterparts to measure performance degradation.

Model	Error Rate	LLM as Judge			Refusal	BertScore		
		Fluency	Comp.	Acc.	Rate (%)	P	R	F1
Google Gemini	10	4.9	4.9	4.8	0.7	82.4	82.2	82.3
	20	4.8	4.3	4.3	8.4	79.2	79.0	79.1
	30	4.7	3.2	3.1	29.4	76.0	75.3	75.6
	40	4.7	2.3	2.2	59.1	73.4	72.7	73.0
Claude sonnet 3.7	10	4.8	4.9	4.8	1.0	83.0	80.0	84.0
	20	4.7	4.5	4.5	8.0	80.0	78.0	80.0
	30	4.5	3.4	3.3	19.7	77.0	77.0	76.0
	40	4.0	2.1	2.0	35.0	71.0	72.0	73.0
GPT-4 Omni	10	4.9	4.8	4.7	0.4	85.9	85.8	85.9
	20	4.5	4.2	4.1	2.5	81.2	80.9	81.1
	30	4.1	3.1	3.0	13.4	77.1	76.7	76.9
	40	4.2	2.2	1.9	38.1	73.8	73.1	73.4

Table 2: Model Performance Across Error Rates: Fluency, Comprehension, Accuracy, Refusal Rate, and BERTScore

4.4.1 Metrics for Measuring Robustness

BERTScore for Semantic Similarity: To assess how typographical noise affects responses, we compute BERTScore between the model's outputs for clean and noisy inputs. Unlike BLEU (Papineni et al., 2002), which relies on n-grams, BERTScore leverages contextual embeddings from pre-trained models to measure semantic similarity.

However, BERTScore's effectiveness for Yorùbá is limited by the poor quality of its language embeddings in multilingual models, as low-resource languages often lack sufficient training data for robust representations. As a result, while it can measure similarity, it sometimes fails to reflect how dissimilar two Yorùbá sentences truly are, necessitating additional evaluation methods.

LLM-as-a-Judge Evaluation: Given BERTScore's limitations, we use an LLM-as-a-Judge approach, leveraging Google's Gemini 2.0 Flash for human-like evaluation. This method assesses whether the models maintain meaningful understanding despite noise. The system prompt provided to the LLM acting as judge is show in Appendix B.

The evaluation process follows these steps:

1. The clean question and the noisy-response pair are fed to the model.

2. The model scores the response, based on the following:

   - **Fluency**: Grammatical correctness and naturalness.
   - **Comprehension**: Understanding of the question.
   - **Accuracy**: Correctness of the response.

3. The model also classifies responses as either:

   - A valid attempt at answering the question.
   - A refusal or failure to understand, including responses like: *"Mo nílò àlàyé síwájú sí"* ("I need more clarification.") or *"Èmi kò lè dáhùn ìbéèrè yìí."* ("I can't provide an answer.").

This helps us to calculate the refusal rate:

$$\text{Refusal Rate(RR)} = \frac{\text{Number of refusals}}{\text{Total questions asked}}$$

By combining BERTScore with LLM-based evaluation, we obtain a more comprehensive assessment of model performance, capturing both semantic similarity and human-like judgment across varying levels of typographical noise.

5 Results and Findings

Table 2 presents the main results on the effect of varying levels of typographical noise in Yorùbá sentences on LLM, using different evaluation metrics across the three models.

5.1 Overall Performance Trend

The findings reveal that typographical noise severely affects comprehension and accuracy once

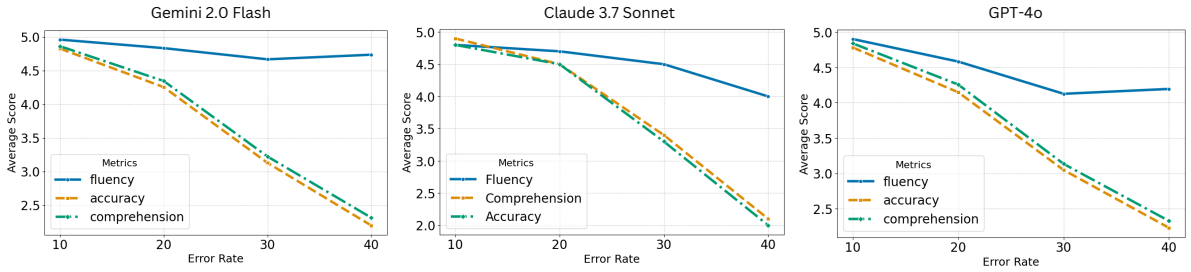

Figure 2: LLM-as-Judge Evaluation of Fluency, Comprehension, and Accuracy Across Error Rates

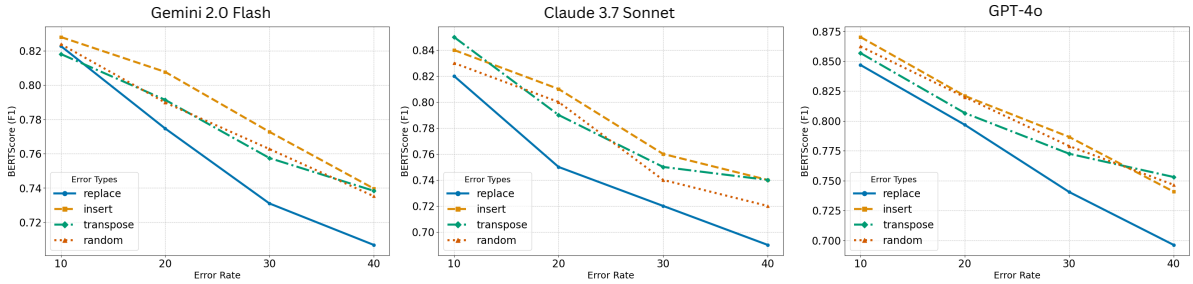

Figure 3: BERTScore (F1) Evaluation Across Error Rates and Error Types

it exceeds 20% - all models show comparable difficulty in extracting meaning from increasingly distorted inputs. Fluency remains relatively stable across all models, indicating that while the models can still generate well-formed sentences, they often misinterpret noisy inputs or in other cases simply say they cannot answer or need more information in a well-written sentence. Similarly, the refusal rate increases significantly after the 20% noise level, indicating that the models refuse to respond as the noise increases. This suggests that, past a certain threshold, models prioritize avoiding incorrect responses over attempting a response based on uncertain input.

5.2 Which type of error has the most significant effect on performance?

Different error types impact performance in different ways, as seen in Figure 3. From the graph, we note that insertion errors introduce minor noise, but do not significantly degrade comprehension. In contrast, replacement errors cause the most substantial drop, as they alter the core word structures. Random and swap errors produced mixed results, but followed a general downward trend.

5.3 Which of the models is more robust?

No one model stands out to be more robust, instead each exhibits some unique trends. For example, in Table 2, we note that at higher noise levels (30-40%) GPT-4o tends to attempt answering the question even when comprehension is very low, but Gemini tries to play it safe by declining to give an answer. From Figure 2, we can see that claude performs slightly better in comprehension than GPT-4o at lower noise levels (10-20%) but deteriorates faster at higher noise rates. Gemini maintains the highest stability in fluency, but its accuracy and comprehension decline significantly at 30% noise and beyond.

5.4 What kind of performance do we see for English

A similar evaluation was conducted on the English translations of the Yorùbá sentences using the same error injection strategy, revealing a stark contrast in model robustness. While Yorùbá comprehension drops rapidly with increasing noise levels, As expected, English maintains high accuarcy and comprehension scores, this is shown in Appendix A. This further illustrates the fact that LLMs are significantly more resilient to typographical noise in English due to greater training data exposure and familiarity with noisy text variations in high-resource languages.

6 Conclusion

This study highlights the critical challenge of maintaining robustness in LLMs under typographical noise within low-resource languages, specifically

focusing on Yorùbá, a tonal language highly sensitive to orthographic nuances such as diacritics. Our experimental results underscore the vulnerability of state-of-the-art models (GPT-4 Omni, Gemini 2.0 Flash, and Claude 3.7 Sonnet) to typographical errors in Yorùbá QA tasks. These findings highlight the urgent need for noise-aware training, emphasizing typographical robustness, particularly for low-resource languages like Yorùbá. We recommend for the creation of dedicated, noisy Yorùbá QA benchmarks and noise-aware training strategies to improve real-world robustness of multilingual LLMs.

Limitations

Our research has several limitations that future studies could address. Firstly, the use of synthetic typographical errors may not fully capture the complexity and variability of real-world user-generated typing errors. Collecting genuine noisy Yorùbá data would enhance ecological validity and applicability of findings. Additionally, although the dataset scales up to 2400 samples from an initial set of 50 QA pairs, incorporating more QA pairs would likely enhance generalizability and robustness assessments. Additionally, better semantic similarity metrics tailored specifically to Yorùbá should be developed, given the limitations of multilingual BERTScore. Lastly, periodic re-evaluation using updated LLMs is necessary to reflect continuous advancements in model robustness.

Acknowledgments

We would like to thank Abraham Owodunni, whose initial idea sparked the curiosity that led to this research. His guidance in refining the research question and validating key ideas was invaluable. I also appreciate the support of the MLC Nigeria community of independent researchers, whose encouragement and insights were instrumental throughout the research process.

References

Shanqing Cai, Subhashini Venugopalan, Katrin Tomanek, Ajit Narayanan, Meredith Morris, and Michael Brenner. 2022. Context-aware abbreviation expansion using large language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1261–1275, Seattle, United States. Association for Computational Linguistics.

Asa Cooper Stickland, Sailik Sengupta, Jason Krone, Saab Mansour, and He He. 2023. Robustification of multilingual language models to real-world noise in crosslingual zero-shot settings with robust contrastive pretraining. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1375–1391, Dubrovnik, Croatia. Association for Computational Linguistics.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2023. Do multilingual language models think better in english? *Preprint*, arXiv:2308.01223.

Isa Inuwa-Dutse. 2025. Naijanlp: A survey of nigerian low-resource languages. *Preprint*, arXiv:2502.19784.

Toheeb A. Jimoh, Tabea De Wille, and Nikola S. Nikolov. 2025. Bridging gaps in natural language processing for yorùbá: A systematic review of a decade of progress and prospects. *Preprint*, arXiv:2502.17364.

Ankit Kumar, Piyush Makhija, and Anuj Gupta. 2020. Noisy text data: Achilles' heel of bert. *Preprint*, arXiv:2003.12932.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.

Haoyu Wang, Guozheng Ma, Cong Yu, Ning Gui, Linrui Zhang, Zhiqi Huang, Suwei Ma, Yongzhe Chang, Sen Zhang, Li Shen, Xueqian Wang, Peilin Zhao, and Dacheng Tao. 2023. Are large language models really robust to word-level perturbations? *Preprint*, arXiv:2309.11166.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Yunxiang Zhang, Liangming Pan, Samson Tan, and Min-Yen Kan. 2022. Interpreting the robustness of neural NLP models to textual perturbations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3993–4007, Dublin, Ireland. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Model Performance on Translated English Questions

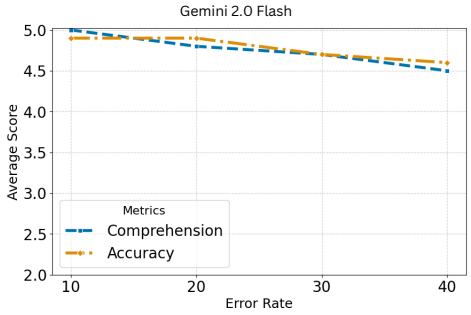

Figure 4: Gemini 2.0 Evaluation of Comprehension and Accuracy Across Error Rates

B System Prompt for LLM-as-Judge Evaluation

LLM-as-Judge System Prompt

You are an expert evaluator of Yoruba language responses. You will be shown a question in Yoruba and the response provided by an AI system. Your task is to rigorously assess the quality of the response.
Important Considerations:

- **Yoruba Language Expertise:** Assume the role of a native Yoruba speaker with deep linguistic knowledge.
- **25-Word Limit:** The AI's response is constrained to a maximum of 25 words.

1. Response Status (Choose One):

- **A. Direct Answer:** The AI provides an answer, even if incorrect.
- **B. Explicit Refusal/Uncertainty:** The AI explicitly refuses to answer or asks for clarification.

Evaluation Criteria (Score 1-5):

- **Fluency:** Is the response grammatically correct and natural?
- **Accuracy:** Does the response correctly address the question?
- **Comprehension:** Does the response demonstrate an understanding of the question?