HausaNLP at SemEval-2026 Task 7: A Locale-Conditional, Multi-Model Prompting Pipeline for Hausa Cultural Question Answering

Faisal Muhammad Adam[1] **Sani Aji**[2]
Lukman Jibril Aliyu[3] **Abdulhamid Abubakar**[4]
[1]**ACETEL, National Open University of Nigeria**
[2]**Department of Mathematics, Faculty of Science, Gombe State University, Gombe, Nigeria**
[3]**HausaNLP** [4]**Nasarawa State University, Keffi**
faisaladamm@gmail.com ajysani@yahoo.com
lukman.j.aliyu@gmail.com abdulhamid@ab-bkr.com

Abstract

We describe HausaNLP's submission to SemEval-2026 Task 7 Track 1 (short-answer cultural question answering). Our system is a training-free, prompt-based pipeline targeting native Hausa (ha-NG). Two design decisions distinguish it from a generic zero-shot baseline. We use *locale-conditional prompting*: ha-NG questions receive a system prompt instructing concise standard Hausa output with explicit Boko-script characters (ɓ, ɗ, ƙ, ɣ). Second, we use a *two-model fallback* pipeline: GPT-4o handles the primary pass, and Gemini 1.5 Flash retries any rows where the primary call returned an error or empty output, separating model-knowledge failures from API-availability failures. On the official development leaderboard, our best run reached 36.4 accuracy. Error analysis shows that a non-trivial fraction of failures are placeholder strings caused by API errors rather than incorrect generations, and that surface-level mismatches (verbosity, orthographic variation) account for many of the remaining errors. Code, prompts, and processing scripts are released for reproducibility.

1 Introduction

Large language models (LLMs) encode useful but uneven cultural knowledge across languages and communities (Naous et al., 2024; Durmus et al., 2023). This is especially challenging for low-resource languages, where knowledge is often thinner and reference answers may reflect narrow regional or generational norms (Joshi et al., 2020). SemEval-2026 Task 7 (Task 7 Organizers, 2026), built on BLEnD (Myung et al., 2024), evaluates this setting with short-answer cultural questions. Track 1 requires brief answers in the question language, scored by reference-string comparison, so we use the term *string-level scoring*.

For Hausa (ha-NG), three issues matter most: models may be too verbose, they may miss Boko characters (ɓ, ɗ, ƙ, ɣ), and API failures directly reduce accuracy because empty outputs are scored as wrong.

Our submission is a simple, training-free pipeline organized around these three issues. We do not fine-tune on BLEnD and do not use BLEnD instances as in-context examples, in line with task policy on hidden evaluation data. Our contributions are: (i) a locale-conditional prompt formulation for ha-NG; (ii) a two-model fallback pipeline that recovers from primary-API errors; (iii) an error analysis that separates generation errors from infrastructure-level placeholder failures.

2 Background and Related Work

A growing body of work probes the cultural alignment of large language models. Naous et al. (2024) show that LLMs frequently default to Western cultural defaults when prompted in non-Western languages, and Durmus et al. (2023) demonstrate that subjective opinions encoded in LLMs cluster around a narrow set of countries. BLEnD (Myung et al., 2024) provides a controlled benchmark for everyday cultural knowledge across 16 countries and 13 languages. SemEval-2026 Task 7 builds on BLEnD and adds a hidden test split with shared evaluation infrastructure (Task 7 Organizers, 2026).

Hausa is widely spoken in West Africa but remains under-represented in modern LLM evaluation. Prior African NLP work includes resources such as MasakhaNER (Adelani et al., 2021), AfriSenti (Muhammad et al., 2023), AfriBERTa (Ogueji et al., 2021), and AfroXLM-R (Alabi et al., 2022). Joshi et al. (2020) also identify Hausa as digitally underrepresented relative to its speaker base.

Few-shot and zero-shot prompting (Brown et al., 2020; Kojima et al., 2022) remain standard interfaces to general-purpose LLMs, including short-answer QA. In this setting, prompts must balance completeness and brevity. Constrained decoding (Lu et al., 2021) could help, but public APIs do not

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1701–1706

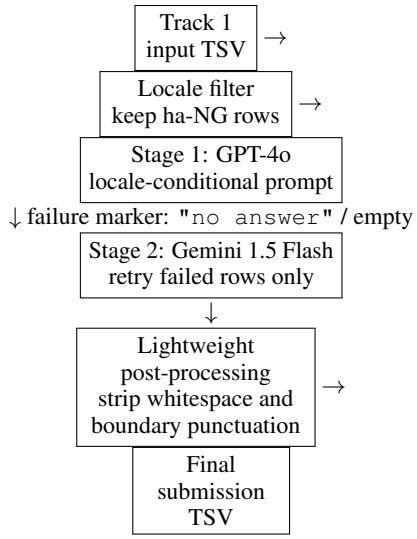

Figure 1: System overview of the two-stage ha-NG pipeline. Non-target rows are filled with "not applicable"; only rows with Stage 1 failure markers are sent to Stage 2.

always expose the needed controls, so we rely on prompting and lightweight post-processing.

3 System Description

3.1 Pipeline overview

The pipeline runs in two stages. **Stage 1** sends every ha-NG row to GPT-4o (OpenAI, 2023) with a locale-specific system prompt; transient API errors and empty responses are written as the placeholder string "no answer". **Stage 2** rereads the Stage 1 output and reissues only the rows whose prediction is a known failure marker ("no answer" or empty) to Gemini 1.5 Flash (Gemini Team, Google, 2023) with a different, more compact prompt. Stage 2 failures are written as "error_still_failed". Rows outside the ha-NG locale are filled with "not applicable" and not sent to any model.

This two-stage design is motivated by the observation that primary-model failures during prototype runs were dominated by rate limits and safety filtering rather than knowledge gaps, and that a second provider with different infrastructure recovered many of these rows. Figure 1 summarizes the full processing flow from locale filtering through fallback repair and final submission construction.

3.2 Models and decoding

Stage 1 (primary): OpenAI's GPT-4o (API model identifier `gpt-4o`), accessed via the OpenAI Chat Completions API. Decoding uses `temperature = 0.1` and `max_tokens = 20`.

Stage 2 (fallback): Google's Gemini 1.5 Flash (API model identifier `gemini-1.5-flash`), accessed via the Google GenAI API. Decoding uses `temperature = 0.1` (`max_tokens` left at the API default).

Low temperature is chosen to reduce surface variation across runs and improve adherence to the brevity instruction. We do not use top-p truncation, stop sequences, or any decoding-time constraint beyond the token budget.

3.3 Locale-conditional prompts

We used two prompts in total: One GPT-4o system prompt and One Gemini user prompt, conditional on locale. All prompts are reproduced verbatim below.

Prompt 1: GPT-4o prompt for ha-NG (Stage 1)
```
You are a wise elder and cultural expert from Northern Nigeria. Answer the question concisely in standard Hausa. IMPORTANT: Use the correct Boko script characters: ɓ, ɗ, ƙ, ɣ. Do not provide full sentences. Just the entity name or short phrase.
```

Prompt 2: Gemini 1.5 Flash prompt for ha-NG (Stage 2)
```
Tambaya: {question}. Bada amsa a takaice cikin harshen Hausa (Boko script). Lambobi kawai idan an tambaya.
```

The GPT-4o prompt uses English instructions to control output length, while the Gemini repair prompt is written in Hausa. We kept this asymmetry because it worked better in development and reduced some Stage 1 safety-trigger failures.

3.4 Reliability and post-processing

We use a fixed inter-request delay of 4.1 seconds during Stage 2 to respect Gemini API rate limits; Stage 1 has no fixed delay beyond what the OpenAI client imposes. Transient exceptions in either stage are caught and recorded as placeholder strings rather than being retried. We do not implement exponential backoff. Persistent failures from Stage 2 are left as "error_still_failed" in the final submission; this preserves TSV row alignment for scoring at the cost of a guaranteed-wrong row.

Post-processing on every non-placeholder prediction applies, in order: (i) `str.strip()` to remove surrounding whitespace; (ii) removal of leading and trailing characters in the set { . , " , ' } for GPT-4o outputs and { . , " , * } for Gemini outputs (the additional * covers Markdown em-

phasis that Gemini occasionally produces). We do not lowercase, normalize Unicode, normalize Boko-script characters, or truncate to a fixed token count. This is intentionally conservative: we found that aggressive normalization risks altering Hausa surface forms in ways that change meaning.

In practice, our output-format control comes from four mechanisms working together: the prompt instruction to return only an entity name or short phrase, low-temperature decoding, the Stage 1 token cap of `20`, and the lightweight boundary-character trimming described above. We do not use grammar-constrained decoding, stopword filtering, or dictionary-based canonicalization.

3.5 Data and Experimental Setup

We used the Track 1 SAQ input file released on CodaBench (questions only). After loading, we filtered rows whose locale matched `ha` or `NG`, which selected the ha-NG (native Hausa) subset of about 500 questions. Non-target rows were filled with `"not applicable"` and submitted as such. No BLEnD labels were available during development, so model and prompt selection relied on official leaderboard feedback and qualitative inspection of our own outputs.

Our model choice was pragmatic rather than exhaustive. We used GPT-4o as the primary model because pilot runs showed better adherence to short-answer instructions, and we used Gemini 1.5 Flash as the fallback because it provided an independent API stack that recovered many rows that had failed upstream for infrastructure reasons. Qualitative inspection consisted of manually reviewing the full set of 500 ha-NG predictions from the final run and annotating recurring error types such as placeholders, truncation, orthographic substitution, and sentence-length violations. We did not run a controlled multi-run variance study or a formal ablation over alternative prompts and decoding settings, so we avoid strong causal claims beyond the descriptive patterns reported here.

We did not fine-tune on BLEnD and do not use BLEnD instances as few-shot in-context examples, following task policy on hidden evaluation data (SemEval Organizing Committee, 2026a,b).

4 Results and Error Analysis

4.1 Official result

Our best development-leaderboard run reached **36.4 accuracy**. We used submissions primarily to validate end-to-end pipeline behavior under the shared evaluation setup rather than to run a controlled ablation. Accordingly, we treat differences across submissions as descriptive only and do not interpret them as statistically meaningful evidence for one prompt variant over another.

Statistic	Value
Mean word count	3.04
Median word count	3
Maximum word count	6
≤ 1 word	92 (18.4%)
≤ 3 words	294 (58.8%)
≤ 6 words	500 (100.0%)
Apparent truncation	81 (16.2%)

Table 1: Length-related statistics over the 500 ha-NG predictions in the final submission. Apparent truncation is defined in the text.

4.2 Constraint compliance and pipeline reliability

A meaningful fraction of incorrect rows are not generation errors but placeholder strings written by the pipeline itself when both stages failed to return a usable response. Out of the 500 instances in the ha-NG development set, Stage 1 (GPT-4o) produced failure markers or empty strings for 42 cases (8.4%). Stage 2 (Gemini 1.5 Flash) returned non-empty Hausa predictions for 31 of these 42 cases, showing that a substantial portion of primary-stage failures were recoverable through a second provider rather than reflecting unrecoverable knowledge gaps. At the surface level, the brevity constraint is respected strongly: predictions average 3.04 words, the median is 3, the maximum is 6, and all 500 predictions remain within six words (Table 1). Short-answer behavior is therefore not the main bottleneck. The more important failure mode is introduced most likely by our `max_tokens = 20` ceiling: **81 of 500 predictions (16.2%) appear to be truncated mid-word or mid-list**, ending in a partial token (e.g., `"Tuwo da miya ko sh"`, `"Wasan kwallon kafa ko ts"`) or a trailing comma (e.g., `"Doya, gero, dawa,"`). In other words, the same formatting controls that suppress verbosity can also induce unrecoverable string mismatches under the task's string-level scoring protocol.

4.3 Observed error patterns

Manual inspection of the 500 ha-NG predictions reveals three recurring patterns, in addition to the

Error category	Share	Example / reason
Orthographic issue	Moderate	Boko-hook substitutions such as k for ƙ create exact-match failures despite near-equivalent wording.
Cultural hallucination	High	Some wrong answers appear culturally plausible but name a different food, festival, or practice than the reference.
Constraint violation	High	Truncation, trailing punctuation, and occasional full-sentence outputs break the canonical short-answer format.
Persistent failure	Low	A small set of rows remain as placeholders after both stages, reflecting unresolved API or safety failures.

Table 2: Approximate manual breakdown of major error categories observed in the 500 ha-NG predictions. The labels summarize recurring patterns rather than exact adjudicated counts.

quantified truncation and Boko-script issues above:

1. **Mid-word / mid-list truncation.** As discussed above, 16.2% of predictions are cut off by the `max_tokens = 20` ceiling. A representative example is `"Tuwo da miya ko sh"`, which appears to begin enumerating "shinkafa" (rice) but stops at `"sh"`.

2. **Latin substitution for Boko characters.** The Boko character ƙ is sometimes replaced by an unhooked Latin k, e.g., `"Kwallon kafa"` (predicted) versus the Boko-orthography form `"Kwallon ƙafa"` (which other rows produce, indicating the model is capable of the form). The resulting strings differ at character level even though the underlying word is the same.

3. **Full-sentence answers.** A small number of predictions are grammatically complete sentences ending in a period (e.g., `"Ba a yin murnar Halloween."`) rather than the canonical short noun phrase. These survive post-processing because trailing-period stripping leaves a multi-word sentence intact.

These patterns are consistent with the pipeline capturing the relevant cultural concept but failing to produce the canonical surface form expected by the reference. Table 2 shows that incorrect cultural content is the largest error source, but orthographic and formatting-related failures also account for a substantial share of misses under string-level scoring.

4.4 Qualitative examples

Table 3 shows representative incorrect predictions observed during development. Each row contains the prediction ID, the raw model output, the observable defect, and a brief failure-pattern label. The examples illustrate that semantically plausible outputs can fail string-level scoring when they differ from the expected surface form.

5 Discussion and Future Work

The pipeline is simple and reproducible, but performance is limited by model knowledge and by how closely outputs match the canonical reference form. Section 4.2 also shows that the second provider improved robustness by recovering many primary-stage failures.

A direct next step is few-shot prompting with policy-compliant Hausa exemplars that demonstrate brevity and Boko orthography without using BLEnD instances. We also plan to trigger Stage 2 on suspicious outputs, not only hard failures, and to test conservative canonicalization such as punctuation and Unicode normalization.

6 Limitations

Our analysis is limited to ha-NG and to the development phase of the task. We did not evaluate transfer to other language culture pairs, and we did not implement retrieval augmentation or fine-tuning. The string-level scoring used by the task can undercount semantically correct but non-canonical answers, so 36.4 accuracy may not fully reflect the underlying cultural knowledge the pipeline elicits.

7 Ethical Considerations

Cultural QA systems can propagate stereotypes, overgeneralize regional practices, or produce misleading answers for minority communities, and this risk is amplified when answers are presented without context as short canonical phrases. Hausa cultural practice varies across regions (urban Kano, rural Sokoto, diaspora communities) and across generations; a single canonical reference may not represent any of these populations faithfully. We therefore recommend that downstream applications of cultural QA include human review for sensitive or high-stakes use, transparent confidence reporting, and explicit documentation of which language–culture pair and which annotator pool an answer is grounded in. We also note that our use of GPT-4o and Gemini 1.5 Flash inherits whatever cultural biases are present in those models (Naous et al., 2024; Durmus et al., 2023).

ID	Prediction (raw)	Observable defect	Failure pattern
ha-NG_004	`Tuwo da miya ko sh`	Final token `sh` is a partial word; appears to begin "shinkafa".	Mid-word truncation
ha-NG_032	`Doya, gero, dawa,`	Trailing comma; list cut off before next item.	Mid-list truncation
ha-NG_114	`Kwallon kafa da kokawa`	Uses unhooked Latin `k` where Hausa orthography expects ƙ (cf. ha-NG_241 `"Kwallon ƙafa..."`).	Latin substitution for Boko
ha-NG_013	`Ba a yin murnar Halloween.`	Full sentence with terminal period rather than short noun phrase.	Full-sentence answer

Table 3: Representative ha-NG predictions illustrating the three quantified failure patterns. All predictions are taken verbatim from the final submission TSV.

8 Reproducibility Statement

We will release the code, prompts, and processing scripts used in this submission. The release will include: (i) preprocessing scripts for locale filtering of the Track 1 input file; (ii) the Stage 1 GPT-4o inference script with the ha-NG system prompt; (iii) the Stage 2 Gemini 1.5 Flash repair script with the corresponding ha-NG prompt; (iv) post-processing utilities used to construct the final submission files; and (v) a schema-validation step that checks row count, delimiter consistency, and non-empty predictions before submission. API keys and raw provider credentials are excluded; users must supply their own.

9 Conclusion

We presented a training-free, locale-conditional, multi-model prompting pipeline for SemEval-2026 Task 7 Track 1, targeting ha-NG. The pipeline combines GPT-4o as a primary model with Gemini 1.5 Flash as a fallback for failed rows, and it uses distinct prompts to encourage canonical short-form output and Boko-script orthography in Hausa. On the official development leaderboard, the pipeline reached 36.4 accuracy. Our error analysis separates infrastructure-bound placeholder failures from genuine generation errors. It also identifies surface-form mismatches as the largest remaining category of fixable errors and motivates few-shot prompting as a direct next step.

Acknowledgements

We thank the task organisers for releasing BLEnD and the shared evaluation framework, and the broader African NLP community for prior work that made this evaluation possible.

References

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, et al. 2021. MasakhaNER: Named entity recognition for African languages. In *Transactions of the Association for Computational Linguistics*, volume 9, pages 1116–1131.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pages 4336–4349.

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. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*.

Esin Durmus, Karina Nguyen, Thomas I. Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, et al. 2023. Towards measuring the representation of subjective global opinions in language models. *arXiv preprint arXiv:2306.16388*.

Gemini Team, Google. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6282–6293.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4288–4299.

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, Nedjma Ousidhoum, David Ifeoluwa Adelani, Seid Muhie Yimam, Ibrahim Sa'id Ahmad, Meriem Beloucif, Saif M. Mohammad, Sebastian Ruder, et al. 2023. AfriSenti: A Twitter sentiment analysis benchmark for African languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 13968–13981.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, Víctor Gutiérrez-Basulto, Yazmín Ibáñez-García, Hwaran Lee, Shamsuddeen Hassan Muhammad, Kiwoong Park, Anar Sabuhi Rzayev, Nina White, Seid Muhie Yimam, Mohammad Taher Pilehvar, et al. 2024. BLEnD: A benchmark for LLMs on everyday knowledge in diverse cultures and languages. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*.

Tarek Naous, Michael J. Ryan, Alan Ritter, and Wei Xu. 2024. Having beer after prayer? Measuring cultural bias in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 16366–16393.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? No problem! Exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning (MRL)*, pages 116–126.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

SemEval Organizing Committee. 2026a. SemEval system paper requirements. https://semeval.github.io/paper-requirements.html.

SemEval Organizing Committee. 2026b. SemEval system paper template and guidelines. https://semeval.github.io/system-paper-template.html.

Task 7 Organizers. 2026. SemEval-2026 Task 7: Everyday knowledge across diverse languages and cultures. In *Proceedings of SemEval-2026*.