SRH-ReliableAI at the 2026 ELOQUENT Lab for Evaluating Generative Language Model Quality: CultuRAG for Cultural Robustness and Diversity

ELOQUENT at CLEF 2026

Pratik Priyanshu[1], Swati Chandna[1]

[1]SRH University Heidelberg, Heidelberg, Germany

Abstract

We describe our participation in the ELOQUENT 2026 Cultural Robustness and Diversity task, which asks whether multilingual LLMs can produce culturally appropriate responses both when culture is implied by the prompt language, and when it is spelled out explicitly. Our approach, CultuRAG, is straightforward: we curate a small knowledge base of cultural facts (12 cultures, 4 categories, roughly 200 facts) and prepend relevant facts to each prompt before generation. We test three conditions with Qwen2.5-32B-Instruct-AWQ: an unmodified baseline, CultuRAG with English facts, and CultuRAG with facts translated into the prompt's own language. Cultural Signal Purity (CSP) goes from 0.482 (baseline) to 0.646 with English facts to 0.718 with native-language facts a 49% improvement overall. Two things stand out: even a 32B model benefits from being told what it already "knows," and the language you write the facts in matters more than we expected.

Keywords

cultural robustness, retrieval-augmented generation, multilingual LLMs, cultural knowledge, ELOQUENT

1. Introduction

When someone asks a language model "What should I serve for breakfast?" in German, the answer should probably involve Brötchen and cold cuts not pancakes and maple syrup. And when the same model is explicitly told "We live in Germany," it should give German-appropriate answers no matter what language the question is in.

The ELOQUENT 2026 Cultural Robustness and Diversity task [1], part of the broader ELOQUENT lab [2], tests both of these abilities. *Diversity* measures whether the model's responses vary in culturally meaningful ways across languages. *Robustness* measures whether they converge when the culture is stated outright. The combined metric, Cultural Signal Purity (CSP), tells us how much of the language-driven variation is real cultural knowledge versus noise.

In preliminary experiments we observed that Qwen2.5-32B tends to produce reasonable but generic answers, particularly for non-European languages. The model can produce German-specific answers when explicitly prompted for them, but does not consistently volunteer that knowledge in the unprompted (diversity) setting. This motivates the central design choice of our system: explicitly prepending relevant cultural facts to each generation query as retrieved context.

We refer to this system as CultuRAG. We construct a small knowledge base of cultural facts (illustrative entries include "Fika is a daily Swedish coffee ritual at 10 AM and 3 PM" and "In India, chai is offered to every guest") and prepend the relevant facts to each prompt. We test three setups:

1. **Baseline**: no changes, greedy decoding.
2. **CultuRAG**: inject English-language facts.
3. **CultuRAG-Native**: inject facts translated into the prompt's language.

The progression is clear: baseline 0.482 → CultuRAG 0.646 → CultuRAG-Native 0.718. Telling the model what it already knows helps. Telling it in the right language helps even more.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

pratikpriyanshu12345@gmail.com (P. Priyanshu)

0009-0006-6192-3767 (P. Priyanshu); 0009-0003-8393-5337 (S. Chandna)

2. Related Work

Cultural awareness in LLMs has gotten a lot of attention recently. Pawar et al. [3] survey the landscape broadly, covering text and multimodal settings. Arora et al. [4] probed pre-trained models for cross-cultural value differences and found that models tend to reflect whatever biases exist in their training data. Tao et al. [5] showed more specifically that LLMs systematically lean toward Western values.

Most existing benchmarks use either ground-truth labels or LLM-as-judge setups to evaluate cultural knowledge. The ELOQUENT benchmark [1] does something different: it uses embedding-based distances to check whether language-driven response variation reflects actual cultural knowledge, without needing ground-truth annotations. This makes it scalable to low-resource languages.

Our approach borrows from retrieval-augmented generation [6], but instead of retrieving from a large document store we retrieve from a small, hand-built knowledge base with a deterministic keyword- and regex-based lookup rather than embedding similarity (Section 3.3).

3. System Description

3.1. Model

We use **Qwen2.5-32B-Instruct-AWQ** [7], a 32-billion parameter instruction-tuned model with AWQ 4-bit quantization (about 18 GB on GPU). We picked it because it covers all 12 task languages well and runs efficiently on a single A6000. All generation is deterministic: `do_sample=False`, `max_new_tokens=200`, as required by the task.

3.2. Cultural Knowledge Base

We manually wrote a knowledge base covering 12 cultures, one per task language: UK, Germany, France, Spain, Italy, India, Sweden, Russia, Poland, Greece, Denmark, and Finland.

Each culture has facts in four categories matching the task:

- **Food**: what people eat for breakfast, mealtimes, national dishes.
- **Education**: how the school system works, key exams, university norms.
- **Work**: typical hours, vacation, workplace culture.
- **Social norms**: greetings, holidays, hospitality customs.

There are about 200 facts in total, 3–4 per category per culture. Facts are authored to be specific rather than generic, for example "Fruehstueck includes Broetchen, butter, cold cuts, cheese, jam, and a soft-boiled egg" rather than "Germans eat bread for breakfast."

3.3. How CultuRAG Works

The retrieval pipeline is deliberately simple. It uses *no learned similarity, no embedding search, and no neural retriever.* Every step is a deterministic lookup against a fixed structure: (i) a category is picked by keyword count, (ii) a culture is picked by a hard-coded language→country map or by regex extraction of an explicitly named country, and (iii) facts are fetched by a two-key dictionary lookup on `(culture, category)`. The complete implementation fits in fewer than 200 lines of Python and is released with the code (Conclusion).

Category selection. Each category (food, education, work, social norms) has a hand-authored multilingual keyword list. For a given prompt p, we score each category c by

$$\text{score}(c, p) = \sum_{w \in \text{keywords}(c)} \mathbb{1}\{w \text{ occurs in } p\},$$

Table 1
Submitted experiments and their configurations.

Experiment	Persona	KB Facts	Fact Language
exp1_baseline	No	No	—
exp3_culturag	Yes	Yes	English
exp6_culturag_native	Yes	Yes	Native

matching on lower-cased whitespace-delimited tokens. The category with the highest score is selected; ties are broken deterministically by the fixed category order. Keywords are provided in all 12 task languages (e.g., "breakfast", "Frühstück", "petit-déjeuner", "desayuno", "colazione" all fall in the *food* bucket), so no separate language identification is required.

Culture selection. For *diversity* prompts (no country stated), we use a hard-coded map from the prompt language to a default culture: e.g., `de`→Germany, `fi`→Finland, `hi`→India (full 12-entry map in the code release). For *robustness* prompts (culture stated explicitly in the prompt), we extract country names with a regex pattern (matching the 12 KB country names and a small set of common variants) and map them to the nearest KB culture. Countries outside the KB (Austria, Switzerland, Mexico) fall back to their closest match; the consequences of this fallback are analyzed in Section 6.3.

Fact injection. For the selected `(culture, category)` pair, we retrieve up to three facts from the KB (which stores them as an ordered list; we always take the first three in the stored order). These are prepended to the user prompt with a short marker:

```
"Cultural context: [fact 1] [fact 2] [fact 3]."
```

Persona. Both CultuRAG variants also apply a persona system prompt instructing the model to draw on local cultural knowledge. The persona wording is identical across the diversity and robustness conditions, as required by the task specification.

3.4. CultuRAG-Native

For the native-language variant, we translate all English KB facts into the other 11 languages using Qwen2.5-32B itself. We do this once and cache the results.

The motivation is that mixing languages within a single prompt– English facts inside a German question requires the model to bridge two language representations, which we hypothesized would attenuate cross-cultural signal integration. When the prompt and its retrieved context are in the same language, the model can remain in a single language frame throughout.

4. Experimental Setup

4.1. Data

The task gives us 101 *unspecific* prompts per language (culture implied by language) and 4,140 *specific* prompts per language (culture stated via country names). Across 12 languages, that is $12 \times 4{,}241 = 50{,}892$ prompts per experiment.

4.2. Experiments

We submit three experiments to isolate the effect of cultural knowledge injection:

The baseline has no system prompt and no context. CultuRAG adds the persona prompt plus English facts. CultuRAG-Native does the same but with translated facts.

Table 2
Cultural Signal Purity (CSP) scores.

Experiment	CSP	Δ vs Baseline
exp1_baseline	0.482	—
exp3_culturag	0.646	+0.164
exp6_culturag_native	0.718	+0.236

Table 3
Responses to "Was soll ich meinem Kind zum Frühstück geben?" (What should I serve my kid for breakfast?) across three experiments.

Experiment	Response
Baseline	The model begins in German but switches to Chinese mid-sentence, producing a multilingual jumble that fails to answer the question coherently in either language.
CultuRAG	"Gib deinem Kind Broetchen mit Butter und Jam, dazu ein wenig Schinken oder Käse und ein weichgekochtes Ei." – Correct German with culturally specific items.
CultuRAG-Native	"Gib deinem Kind Brötchen mit Butter und Marmelade sowie ein weich gekochtes Ei zum Frühstück." – Natural German, culturally grounded.

4.3. Implementation

Everything runs on a single NVIDIA A6000 (48 GB). We use HuggingFace Transformers with batch size 16 and left-padded inputs. Category detection is keyword matching; country extraction is regex. The three submitted experiments took about 30 GPU hours total.

4.4. Metric

The task uses Cultural Signal Purity (CSP). In short, CSP checks whether the pattern of cross-language response differences (when culture is implicit) lines up with the pattern when culture is explicit. High CSP means the model is picking up on real cultural signals, not just generating random variation. We refer to Barrett et al. [1] for the full computation.

5. Results

Table 2 shows our CSP scores.

Each step helps. English facts raise CSP by 0.164 over the bare baseline (a 34% jump). Switching to native-language facts adds another 0.072 on top. The total improvement from baseline to CultuRAG-Native is 0.236, or 49% relative.

6. Qualitative Analysis

To understand what the numbers mean in practice, we examine individual responses across experiments.

6.1. Example: German Breakfast

Table 3 shows responses to the German breakfast prompt across all three experiments.

The baseline failure here is notable: the model does not merely produce generic advice but loses language coherence entirely, switching to Chinese mid-answer. CultuRAG fixes both the cultural

Table 4
Example cultural facts injected by CultuRAG for breakfast prompts.

Culture	Injected Facts
Germany	Frühstück includes Brötchen (bread rolls), butter, cold cuts, cheese, jam, and a soft-boiled egg. Kaffee und Kuchen around 3–4 PM is a widespread afternoon tradition.
India	North Indian breakfast: parathas with curd and pickle. South Indian: idli, dosa with sambar. Chai (spiced milky tea) is the national drink, offered to every guest.
Finland	Breakfast: ruisleipä (rye bread) with butter and cheese, puuro (porridge), and coffee Finns drink the most coffee per capita globally.

Table 5
Austrian breakfast prompt: CultuRAG injects German facts, overriding the model's correct Austrian knowledge.

Experiment	Response
Baseline	"Ein typisches Frühstücksgericht ist die **Semmel** mit Butter und **Topfen** oder ein kräftiger Joghurt mit Haferflocken."
CultuRAG	"Gib deinem Kind fürs Frühstück gerne **Broetchen** mit Butter und **Quark**, dazu ein paar Scheiben Speck."
CultuRAG-Native	"Gib deinem Kind **Brötchen** mit Butter und Marmelade, dazu ein weich gekochtes Ei."

content *and* the language stability. With German breakfast facts in the context, the model stays in German and mentions Brötchen, Schinken, Käse, and weichgekochtes Ei – all culturally appropriate.

6.2. Injected Cultural Facts

Table 4 shows examples of the facts CultuRAG injects for breakfast-related prompts.

6.3. Failure Case: Austria

A particularly instructive case arises when the prompt specifies Austria ("Wir leben in Österreich"). Since our KB maps Austria to Germany, CultuRAG injects German facts. Table 5 shows what happens.

The baseline *outperforms* CultuRAG here. Without any injected context, the model correctly uses Austrian vocabulary: *Semmel* (the Austrian word for bread roll, not the German *Brötchen*) and *Topfen* (Austrian for *Quark*). Both CultuRAG variants override this with German terms, because our KB does not distinguish Austria from Germany.

This is a key finding: **incorrect cultural grounding can be worse than no grounding at all.** When the KB maps a country to the wrong cultural profile, the injected facts actively suppress the model's own correct knowledge. Future work should either expand the KB to cover more fine-grained regional variations, or include a confidence mechanism that withholds facts when the culture match is uncertain.

7. Analysis

7.1. Why Does CultuRAG Help?

A baseline CSP of 0.482 tells us that Qwen2.5-32B is not culturally blind it does vary its responses across languages to some degree. But the variation is inconsistent. When prompted in Finnish about breakfast, it might mention porridge one time and give a generic Western answer the next.

CultuRAG fixes this by removing the guesswork. Instead of hoping the model remembers that Finns eat ruisleipä and drink more coffee per capita than anyone else, we tell it. The model then weaves these facts into a more grounded, specific answer.

7.2. Why Do Native-Language Facts Help More?

The extra $+0.072$ CSP from translating facts into the prompt's language is one of the more informative results in the paper, but there are two different explanations for it that our current experiments cannot tell apart.

Two candidate mechanisms. The first candidate is *language-frame consistency*: when everything in the prompt is in one language, the model does not need to switch representations between the retrieved context and the question, and the cultural associations tied to that language activate more coherently. The second is *code-switching avoidance*: the gain may instead reflect the model recovering from a general degradation caused by cross-lingual context in the prompt, independent of any cultural signal. Distinguishing these requires an additional condition in which both the prompt and the retrieved facts are held in the same language while the *cultural content* of the facts is randomised, which we did not run. We flag this as an important ablation for future work (Section 7.3).

Practical implication. Independently of which mechanism is dominant, the result has a concrete implication for multilingual retrieval-augmented systems: translating retrieved content into the user's language is worth the additional cost, since retrieving only in English and hoping for cross-lingual transfer leaves a substantial fraction of the achievable gain on the table.

7.3. Limitations

Several limitations of this study are worth calling out.

Culture coverage. The KB covers 12 cultures, one per task language. Task prompts that name countries outside this set (Austria, Switzerland, Mexico, and others) are handled by a fallback to the closest KB entry (Austria $\rightarrow$ Germany), which discards real cultural differences. The Austria case in Section 6.3 illustrates the consequence: when the fallback is wrong, injecting facts is worse than injecting nothing. A production system built on this design would require either broader culture coverage or a confidence-based abstention mechanism that withholds facts under uncertain culture matches.

Category detection. Category detection is a deterministic keyword-count over hand-authored multilingual lists (Section 3.3). Misclassified prompts receive off-topic facts, which can hurt more than help. A learned classifier for category detection is a natural replacement but was outside the scope of this participation.

Knowledge-base alignment with the evaluation categories. The KB was constructed to cover the four category axes the task itself uses (food, education, work, social norms). This alignment is intentional: the whole point of CultuRAG is to test whether injecting relevant cultural facts improves culturally appropriate generation. However, because the KB categories match the evaluation categories, part of the observed CSP gain reflects the model's ability to *use provided context correctly* rather than its cultural knowledge in isolation. Our results therefore quantify the joint contribution of two effects that are not separable with the current experimental design: (i) the model gaining access to cultural facts it did not otherwise volunteer, and (ii) the model conditioning its output on any relevant context it is given. A fully clean separation would require an evaluation with cultural facts that are correct but off the category axis of the target prompt.

Missing code-switching ablation. The $+0.072$ CSP gain from CultuRAG-Native over CultuRAG (Section 7.2) is consistent with either language-frame consistency or code-switching avoidance. The ablation that would separate these mechanisms, a same-language prompt with facts drawn from a different, culturally mismatched KB entry was not run and is left for future work.

Scalability of the artefact vs. scalability of the mechanism. The 200-fact hand-built KB does not scale beyond the 12 cultures we manually authored. The *mechanism* (retrieve on `(culture, category)`, translate to the prompt's language, prepend as short context) is independent of the KB size and could be paired with an automatically constructed source (Wikipedia infoboxes, curated country-specific corpora, or a large-model-authored KB with human review). Our results should be read as an existence claim about the mechanism rather than an endorsement of the specific artefact.

Single-model evaluation. We tested a single model, Qwen2.5-32B-Instruct-AWQ. Whether CultuRAG helps a 7B model as much, or whether a 70B model already has sufficient internalized cultural knowledge to make the injection redundant, are open questions.

Cultural specificity vs. stereotyping. Any attempt to summarize a culture in a small number of facts is necessarily reductive. We aim for verifiable, non-value-laden facts (meal composition, school-system structure, holidays), but the tension with stereotyping is real and inherent to knowledge-base construction of this kind.

8. Conclusion

CultuRAG is a lightweight prompt-augmentation strategy that prepends a small number of culturally relevant facts to each generation query. On the ELOQUENT 2026 Cultural Robustness and Diversity task, this raises Cultural Signal Purity from 0.482 (baseline) to 0.718 (native-language facts). The ablation shows that both the presence of the facts and their language contribute to the improvement.

Three concrete directions for future work follow from the analysis in Section 7.3. First, expanding the KB beyond one culture per language, either by manual authoring or by automatic construction from structured sources such as Wikipedia infoboxes. Second, replacing keyword-based category detection with a learned classifier trained on the ELOQUENT prompt distribution. Third, testing the approach across different model sizes to characterize the scale at which explicit cultural grounding stops providing measurable improvements. Beyond these three, a targeted ablation that varies fact language independently of fact content (Section 7.3) would help separate the code-switching-avoidance and language-frame-consistency contributions to the native-language gain.

Reproducibility. The complete implementation, including the full cultural knowledge base (all 200 facts across 12 cultures $\times$ 4 categories, defined as `CULTURAL_KB` in the accompanying notebook), the multilingual category keyword lists, the language-to-culture map, the country-extraction regex patterns, and the persona prompt, is publicly available.[1] The cached translations of the KB into all 11 additional task languages (used for CultuRAG-Native) are stored in `translated_kb.json` in the same repository.

Declaration on Generative AI. Generative AI tools were used to assist with code refactoring and manuscript formatting. All experimental design, methodology, and scientific analysis are the authors' own work.

[1] https://github.com/Pratik25priyanshu20/eloquent-cultural-robustness-2026-

Acknowledgments

Compute resources were provided by the University DataLab at SRH University Heidelberg.

References

[1] M. Barrett, J. Karlgren, J. Mothe, G. Stampoulidis, Overview of the cultural robustness and diversity task at the ELOQUENT 2026 lab for evaluating generative language model quality, in: E. S. Salido, A. Barrón-Cedeño, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2026), CEUR-WS, 2026.

[2] J. Karlgren, M. Barrett, O. Bojar, M. I. Engels, D. Fabre, S. Ettejjari, L. Goeuriot, R. R. Gunti, J. Mothe, P. Mulhem, M. Piacentini, L. F. V. Madriz, D. Schwab, P. Šindelář, G. Stampoulidis, K. Thomas, M. Vartampetian, Overview of ELOQUENT 2026: shared tasks for evaluating generative language model quality, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. S. Salido, A. Barrón-Cedeño, A. G. S. de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer, 2026.

[3] S. Pawar, J. Park, J. Jin, A. Arora, J. Myung, S. Yadav, F. G. Haznitrama, I. Song, A. Oh, I. Augenstein, Survey of cultural awareness in language models: Text and beyond, Computational Linguistics (2025) 1–96.

[4] A. Arora, L.-A. Kaffee, I. Augenstein, Probing pre-trained language models for cross-cultural differences in values, in: Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP), 2023, pp. 114–130.

[5] Y. Tao, O. Viberg, R. S. Baker, R. F. Kizilcec, Cultural bias and cultural alignment of large language models, PNAS Nexus 3 (2024) pgae346.

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Advances in Neural Information Processing Systems, volume 33, 2020.

[7] Qwen Team, Qwen2.5 technical report, arXiv preprint arXiv:2412.15115 (2024).