MoMo at SemEval-2026 Task 9: Inference-Only Prompting vs. Fine-Tuning for Multilingual Polarization Detection

Sushant Kumar Ray
University of Delhi
skray1331@gmail.com

Rakshita Saksainaa
Delhi Skill and Entrepreneurship University
btech41823039@dseu.ac.in

Abstract

We describe our submission to SemEval-2026 Task 9 Subtask 1, which focuses on multilingual polarization detection over the POLAR dataset. We compare three adaptation paradigms: fully fine-tuned multilingual encoders, frozen encoders augmented with lightweight residual heads, and inference-only multilingual LLM prompting in zero-shot and few-shot settings. For few-shot prompting, we evaluate both random and similarity-based support example selection. Similarity-based few-shot prompting with a multilingual LLM competes with our fine-tuned encoder baselines while requiring no task-specific training. We further analyze energy usage, stability across prompt selections and per-language behavior to characterize trade-offs between architectural adaptation and prompt-based inference. While our submission uses a fully fine tuned XLM-RoBERTa Large, the results indicate that inference-only prompting can be a competitive and energy-efficient alternative to task-specific fine-tuning in multilingual classification.

1 Introduction

Multilingual and multicultural text classification is a core challenge in natural language processing. Polarization detection across diverse languages and cultures is a particularly difficult instance of this problem. This work describes our SemEval-2026 Task 9 submission for Subtask 1 (polarization detection) (Naseem et al., 2026a). The dominant paradigm for this setting has been task-specific fine-tuning of pretrained multilingual encoders (Conneau et al., 2020; He et al., 2021). In our submission, we used XLM-RoBERTa Large (Conneau et al., 2020) fully fine tuned on the shared task dataset. Although effective, fine-tuning requires model training for every new task and introduces additional computational, energy, and engineering cost.

Recent multilingual large language models (LLMs) challenge the necessity of this paradigm (Brown et al., 2020; Grattafiori et al., 2024; Yang et al., 2025). These models often exhibit strong cross-lingual transfer and can be applied through zero-shot or few-shot prompting without parameter updates. Prior work has compared fine-tuned encoders and prompted LLMs in terms of predictive performance, but the associated compute and energy trade-offs remain underexplored (Iftikhar et al., 2025; Maliakel et al., 2026). To address this gap, we present a systematic evaluation of fine-tuning against prompting paradigms. We evaluate the following:

1. Fully fine-tuned multilingual encoders.
2. Frozen encoders augmented with lightweight residual classification heads.
3. Multilingual LLMs operating under zero-shot and few-shot prompting conditions.

Within our few-shot experiments, we further contrast random support example selection against top-k cosine-similarity retrieval to analyze the impact of context selection.

Our findings indicate that similarity based few-shot prompting with a multilingual LLM performs on par with fully fine-tuned encoder baselines on the POLAR dataset. Crucially, inference-only LLM strategies avoid task-specific training while maintaining competitive accuracy. Our findings suggest that inference-only multilingual LLMs are a computationally efficient alternative to task-specific fine-tuning.

2 Related Work

2.1 Multilingual Encoder Fine-Tuning

Pretrained multilingual transformer encoders, such as XLM-RoBERTa and mDeBERTa (Conneau et al., 2020; He et al., 2021) are widely used for cross-lingual text classification. These models are trained on large-scale multilingual corpora and are

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 3591–3601

subsequently adapted to downstream tasks via supervised fine-tuning. This paradigm has demonstrated strong performance on various multilingual benchmarks (Hu et al., 2020; Liang et al., 2020; Ruder et al., 2021).

Fine-tuning enables task-specific adaptation of model parameters and often yields substantial performance improvements over base models (Devlin et al., 2019; Radford et al., 2018). However, this approach requires retraining for each new task, leading to repeated computations and energy consumption. This process typically involves multiple training runs and hyperparameter tuning, which amplifies the total cost.

Recent work has explored lightweight alternatives such as adapter modules, residual classification heads, and steering vectors to reduce the number of trainable parameters while preserving performance (Houlsby et al., 2019; Pfeiffer et al., 2021). These approaches aim to minimize the optimization cost while retaining the representational capacity of pretrained encoders. These approaches, however, still rely on task-specific training.

2.2 Large Language Models for Zero and Few-Shot Classification

Large language models (LLMs) have demonstrated strong in-context learning capabilities, enabling zero-shot and few-shot classification without parameter updates (Brown et al., 2020; Liu et al., 2022). Models such as Qwen3 and LLaMa-3 exhibit multilingual competence and can be applied directly to downstream classification tasks through prompting (Yang et al., 2025; Grattafiori et al., 2024).

Prior studies have shown that few-shot prompting can match or even surpass fully fine-tuned encoder baselines in certain tasks (Min et al., 2022; Liu et al., 2022). However, performance often depends on support example selection, prompt design, and sampling variability.

To improve stability, similarity-based retrieval methods have been proposed to select support examples that are semantically close to the input instance (Rubin et al., 2022; Liu et al., 2022). Such retrieval-augmented prompting strategies typically outperform random example selection, especially in multilingual or low-resource settings.

2.3 Energy and Sustainability in NLP

While the pretraining cost of large models is substantial, downstream fine-tuning also contributes to cumulative energy usage across tasks and deployments (Strubell et al., 2019). Studies in green AI argue for evaluating models not only by accuracy, but also by computational efficiency and carbon footprint (Schwartz et al., 2020).

In multilingual shared-task scenarios, repeated fine-tuning of encoders for each new dataset can lead to substantial task-specific energy expenditure. In contrast, inference-only LLM approaches eliminate retraining and adaptation, and potentially reduces per-task energy costs. However, prompting methods introduce longer input sequences and higher token-level inference costs, making the trade-off non-obvious.

3 Task Description and Dataset

SemEval-2026 Task 9 focuses on multilingual text classification in diverse languages and cultures (Naseem et al., 2026a). The objective is multi-label classification: for each predefined category, the model predicts a binary label indicating whether that category is present. The shared-task dataset, adapted from the POLAR dataset (Naseem et al., 2026b), spans multiple languages with varying resource availability, and provides a realistic cross-lingual evaluation scenario. We evaluate subtask-one, which involves polarization detection. The task presents several challenges:

- **Cross-lingual generalization:** Performance must be maintained across diverse languages and scripts.
- **Class imbalance:** Certain categories are underrepresented, increasing the difficulty of macro-level evaluation.
- **Domain variability:** Text instances may vary in style, length, and linguistic structure across languages.

Table 1 summarizes the dataset statistics across all 22 languages. Training set sizes range from 1,700 samples (pan) to 6,991 samples (swa), reflecting substantial variation in per-language resource availability. Class distributions are notably uneven: the proportion of polarized instances varies from 10.7% (hau) to 90.8% (khm), confirming significant class imbalance across languages. The development set maintains approximately 5% of training size per language. Such extreme imbalance in certain languages (e.g., khm, hau) presents a challenging evaluation setting, particularly for models sensitive to class priors.

Table 1: Dataset statistics for POLAR Subtask 1 across all 22 languages. %Pol = percentage of polarized instances in training set.

Lang	Train	Pol	Non	%Pol	Dev	Test
amh	3332	2518	814	75.6	166	1501
arb	3380	1512	1868	44.7	169	1521
ben	3333	1424	1909	42.7	166	1501
deu	3180	1512	1668	47.5	159	1432
eng	3222	1175	2047	36.5	160	1452
fas	3295	2440	855	74.1	164	1484
hau	3651	392	3259	10.7	182	1644
hin	2744	2346	398	85.5	137	1236
ita	3334	1368	1966	41.0	166	1538
khm	6640	6029	611	90.8	332	2988
mya	2889	1682	1207	58.2	144	1301
nep	2005	1008	997	50.3	100	903
ori	2368	683	1685	28.8	118	1066
pan	1700	840	860	49.4	100	809
pol	2391	1003	1388	41.9	119	1077
rus	3348	1023	2325	30.6	167	1508
spa	3305	1660	1645	50.2	165	1488
swa	6991	3504	3487	50.1	349	3147
tel	2366	1274	1092	53.8	118	1066
tur	2364	1155	1209	48.9	115	1093
urd	3563	2476	1087	69.5	177	1606
zho	4280	2121	2159	49.6	214	1927
Total	73,681	39,145	34,536	53.1	3,687	33,288

4 Methodology

We compare the following three paradigms for multilingual classification:

- Fine-tuning of multilingual encoders,
- Frozen encoders augmented with residual adapters, and
- Inference-only multilingual LLM prompting.

To quantify environmental impact, we measure energy usage with the `eco2ai` package (Budennyy et al., 2022). Inference with Qwen3-Next-80B is run on an NVIDIA H100 GPU due to VRAM requirements, whereas encoder fine-tuning and adapter training are run on an NVIDIA P100 GPU (Choquette, 2023; NVIDIA, 2016).

4.1 Encoder-Based Fine-Tuning

We evaluate two multilingual transformer encoders:

- XLM-RoBERTa Large (Conneau et al., 2020), which we used for our submission, and
- DeBERTaV3-base (He et al., 2021)

All model parameters are updated during training. We perform hyperparameter search over a predefined set of learning rates, selecting configurations based on development performance. However, hyperparameter grid search is not considered for our energy requirements. To account for optimization instability, each configuration is trained ten times. We report median macro-F1 and quartiles across runs.

4.2 Residual Adapters over Frozen Multilingual Encoders

To reduce task-specific training cost while preserving representational capacity, we use a lightweight residual adapter over frozen encoder backbones (Houlsby et al., 2019; Pfeiffer et al., 2021). Let $X \in \mathbb{R}^d$ denote the encoder representation of the input. The residual module is defined as:

$$R(X) = A(X) + B(X) \tag{1}$$

where A is a nonlinear transformation (linear function followed by ReLU) and B is a linear projection. We stack two such modules for improved representational capacity, and compute final logits from $R(R(X))$. Each application of R progressively reduces the vector dimensionality. This architecture enables task adaptation with a small number of additional parameters while keeping the backbone fixed. Only the residual modules and classification head are trained.

4.3 LLM-Based Inference

We evaluate multilingual prompting using two Qwen models: Qwen-3-8B and Qwen-3-Next-80B (Yang et al., 2025). We use Qwen because of its strong multilingual performance. Qwen-3-Next-80B is a mixture-of-experts (MoE) model with approximately 80B total parameters, of which about 3B are active per token. This sparse activation reduces effective inference compute. All experiments are conducted in an inference-only setting with no task-specific adaptation. We fix temperature to 0.7 to quantify variance under stochastic sampling.

Support Example Selection Strategies. We evaluate three prompting regimes:

1. **Zero-shot Classification:** The model receives only task instructions and the input instance.
2. **Random Few-Shot Prompting:** Support examples are sampled uniformly from the training set.
3. **Similarity-Based Few-Shot Retrieval:** For each test instance, we compute embeddings using QwenEmbeddings, EmbeddingGemma, and LaBSE (Zhang et al., 2025; Vera et al., 2025; Feng et al., 2022). Using cosine similarity, we retrieve the top three most similar training examples for each class (polarization and non-polarization) and include them as support examples in the prompt.

We constrain all LLM outputs to a fixed textual format and extract predictions through exact string

matching for deterministic evaluation. In practice, Qwen follows the required output format reliably. Prompt templates are provided in the Appendix A.

5 Results

We evaluate all systems using macro-averaged F1. In addition to overall performance, we analyze stability across runs, per-language behavior, and energy consumption. Full numerical results for all 22 languages are provided in the Appendix B. Here, we report results for seven representative high and low-resource languages.

5.1 Overall Performance

Tables 2 and 3 summarize the overall macro-F1 scores for encoder-based and LLM-based systems, respectively. Table 4 compares thinking and non-thinking inference for Qwen-3-Next-80B under similarity-based few-shot prompting. Zero-shot prompting with Qwen achieves competitive performance but remains below fully fine-tuned encoder baselines. Introducing few-shot support examples substantially improves performance. Random support example selection consistently improves performance across all languages over zero-shot prompting, while similarity-based retrieval provides further gains and consistently outperforms random selection across languages. Notably, similarity-based few-shot prompting with Qwen-3-Next-80B slightly surpasses fully fine-tuned XLM-R and mDeBERTa baselines on some languages. We additionally evaluate the "thinking" variant of Qwen-3-Next-80B for selected languages. Enabling extended reasoning produces moderate improvements for some low-resource languages. However, gains for high-resource languages such as English and Italian were marginal, while energy consumption increased by up to 11 times. This indicates that extended reasoning is not uniformly beneficial and should be weighed against its computational cost.These results highlight that retrieval quality and model capacity are key factors in multilingual polarization detection, particularly under class imbalance and low-resource conditions.

5.2 Adapter-based models

Adapter-based models consistently perform at or below the random baseline across all languages. We attribute this to three factors: (1) restricted parameter capacity may be insufficient for multilingual polarization detection; (2) severe class imbalance (10.7%-90.8% polarized across languages

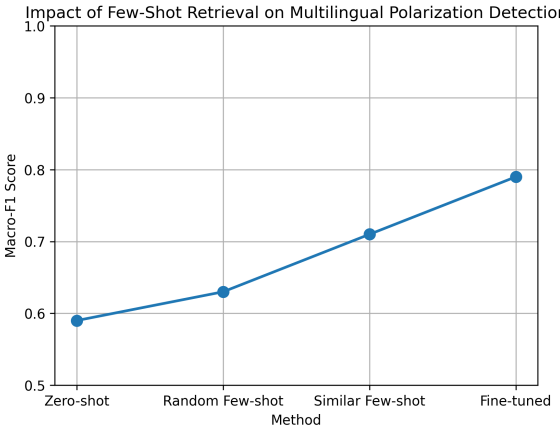

Figure 1: Performance comparison of prompting strategies. Similarity-based few-shot retrieval consistently outperforms zero-shot and random selection, approaching fine-tuned encoder performance.

(see Table 1)) likely biases adapter predictions more than fully fine-tuned models; and (3) detecting subtle cultural and contextual cues may require deeper representation updates than lightweight adapters provide. Our analysis is based on observed performance trends rather than exhaustive hyperparameter exploration (e.g., class-weight tuning or learning rate sweeps). These findings nevertheless highlight important limitations of parameter-efficient adaptation under severe class imbalance and motivate further investigation.

5.3 Stability of Few-Shot Prompting

Few-shot prompting exhibits sensitivity to support example selection. Random few-shot prompts show noticeable variance across runs, indicating instability arising from context sampling. In some cases, random support examples degrade performance relative to zero-shot prompting. In contrast, similarity-based retrieval significantly reduces performance variance and produces more consistent macro-F1 scores across runs. This suggests that semantically aligned support examples stabilize in-context learning in multilingual classification. These findings reinforce that retrieval quality plays a key role in determining both performance and robustness.

5.4 Per-Language Analysis

Per-language macro-F1 results are reported in the Appendix B. Across high-resource languages, encoder-based models and similarity-based LLM prompting perform comparably. Fine-tuned encoders maintain a slight advantage in languages with substantial training representation. In lower-

Table 2: Median macro-F1 scores for encoder-based models over seven representative languages with quartiles, followed by training carbon consumption and per-sample inference carbon consumption (Wh). All systems are trained over 10 runs. Best score per language is highlighted in **bold**.

	Models	eng	urd	ita	rus	fas	arb	zho	avg
Fine-Tuned	XLM-R Large (Our Submission)	**0.79**	**0.78**	0.67	0.79	**0.78**	0.83	**0.90**	**0.79**
	XLM-R Large (Ours)	**0.79** [0.75 - 0.82] 595.64 0.0071	0.77 [0.73 - 0.81] 601.28 0.0073	0.66 [0.62 - 0.70] 589.77 0.0069	0.80 [0.75 - 0.83] 598.14 0.0075	0.77 [0.73 - 0.82] 592.03 0.0067	**0.84** [0.81 - 0.87] 604.51 0.0072	0.89 [0.86 - 0.92] 587.96 0.0076	**0.79** [0.75 - 0.82] 595.62 0.0072
	mDeBERTa	0.78 [0.75 - 0.81] 416.95 0.0062	0.76 [0.71 - 0.80] 419.22 0.0064	**0.69** [0.65 - 0.73] 414.37 0.0059	**0.81** [0.76 - 0.84] 421.08 0.0066	0.76 [0.71 - 0.80] 412.94 0.0057	0.82 [0.79 - 0.85] 418.56 0.0063	0.88 [0.84 - 0.91] 423.11 0.0067	**0.79** [0.74 - 0.82] 418.03 0.0063
Adapter-Models	XLM-R Large	0.36 [0.27 - 0.39] 20.43 0.0085	0.41 [0.23 - 0.41] 20.71 0.0087	0.34 [0.33 - 0.34] 20.18 0.0082	0.33 [0.23 - 0.41] 20.94 0.0089	0.43 [0.21 - 0.43] 19.97 0.0080	0.36 [0.31 - 0.36] 20.56 0.0086	0.34 [0.33 - 0.34] 21.02 0.0091	0.37 [0.27 - 0.38] 20.54 0.0086
	mDeBERTa	0.39 [0.30 - 0.39] 16.55 0.0070	0.41 [0.23 - 0.41] 16.81 0.0072	0.32 [0.32 - 0.34] 16.47 0.0068	0.41 [0.27 - 0.41] 16.93 0.0074	0.21 [0.21 - 0.43] 16.22 0.0066	0.36 [0.36 - 0.36] 16.68 0.0071	0.34 [0.33 - 0.34] 16.99 0.0075	0.35 [0.29 - 0.38] 16.66 0.0071
	RemBERT	0.39 [0.29 - 0.39] 15.23 0.0065	0.25 [0.24 - 0.41] 15.47 0.0067	0.33 [0.32 - 0.34] 15.11 0.0063	0.41 [0.41 - 0.41] 15.62 0.0069	0.39 [0.24 - 0.43] 14.98 0.0061	0.32 [0.31 - 0.36] 15.36 0.0066	0.38 [0.34 - 0.42] 15.74 0.0070	0.35 [0.31 - 0.39] 15.36 0.0066
	LaBSE	0.41 [0.39 - 0.46] 13.24 0.0060	0.38 [0.32 - 0.46] 13.41 0.0062	0.38 [0.35 - 0.44] 13.06 0.0057	0.40 [0.30 - 0.44] 13.57 0.0064	0.32 [0.30 - 0.42] 12.95 0.0055	0.40 [0.37 - 0.48] 13.33 0.0061	0.35 [0.33 - 0.38] 13.62 0.0065	0.38 [0.34 - 0.44] 13.31 0.0061
	Embedding Gemma	0.33 [0.27 - 0.39] 24.02 0.0071	0.41 [0.28 - 0.41] 24.31 0.0073	0.34 [0.33 - 0.34] 23.88 0.0068	0.32 [0.23 - 0.41] 24.46 0.0075	0.43 [0.21 - 0.43] 23.71 0.0066	0.36 [0.33 - 0.36] 24.15 0.0072	0.33 [0.33 - 0.34] 24.57 0.0076	0.36 [0.28 - 0.38] 24.16 0.0072
	Qwen-3-Embeddings-8B	0.46 [0.44 - 0.49] 26.32 0.0095	0.45 [0.33 - 0.51] 26.61 0.0097	0.48 [0.45 - 0.54] 26.14 0.0092	0.46 [0.43 - 0.49] 26.48 0.0099	0.42 [0.33 - 0.49] 25.97 0.0090	0.44 [0.43 - 0.47] 26.35 0.0096	0.42 [0.38 - 0.43] 26.72 0.01	0.45 [0.40 - 0.49] 26.37 0.0096
	Random Baseline	0.50	0.48	0.50	0.48	0.47	0.51	0.51	0.49

Table 3: Median macro-F1 scores for LLM inference over seven representative languages. Each cell reports median, quartiles (Q1-Q3), and mean energy usage (10^{-2} Wh). Each LLM configuration is evaluated over 10 runs with temperature fixed at 0.7. Best score per language is highlighted in **bold**.

Models	Method	eng	urd	ita	rus	fas	arb	zho	mean
Qwen-3-8B	Zero-Shot	0.62 [0.60 - 0.64] 1.68	0.58 [0.56 - 0.60] 1.70	0.60 [0.57 - 0.63] 1.67	0.61 [0.58 - 0.63] 1.69	0.48 [0.44 - 0.51] 1.73	0.57 [0.54 - 0.60] 1.68	0.66 [0.63 - 0.69] 1.75	0.59 [0.56 - 0.62] 1.70
	Random Few-Shot	0.66 [0.63 - 0.69] 1.88	0.62 [0.59 - 0.65] 1.90	0.64 [0.61 - 0.67] 1.87	0.65 [0.62 - 0.68] 1.89	0.52 [0.48 - 0.56] 1.93	0.61 [0.58 - 0.64] 1.88	0.70 [0.67 - 0.73] 1.96	0.63 [0.60 - 0.66] 1.90
	Similar Few-Shot Qwen-Embeddings-8B	0.73 [0.69 - 0.77] 2.20	0.71 [0.67 - 0.75] 2.22	0.68 [0.64 - 0.71] 2.19	0.72 [0.67 - 0.77] 2.21	0.55 [0.49 - 0.59] 2.26	0.73 [0.69 - 0.76] 2.20	0.82 [0.77 - 0.85] 2.28	0.71 [0.66 - 0.74] 2.22
	Similar Few-Shot Embedding Gemma	0.68 [0.63 - 0.72] 2.04	0.65 [0.61 - 0.68] 2.06	0.62 [0.56 - 0.66] 2.03	0.67 [0.63 - 0.72] 2.05	0.49 [0.42 - 0.55] 2.10	0.67 [0.62 - 0.72] 2.04	0.75 [0.71 - 0.78] 2.12	0.65 [0.60 - 0.69] 2.06
	Similar Few-Shot LaBSE	0.64 [0.59 - 0.69] 1.92	0.61 [0.55 - 0.65] 1.94	0.55 [0.49 - 0.59] 1.91	0.61 [0.56 - 0.66] 1.93	0.45 [0.38 - 0.50] 1.98	0.61 [0.55 - 0.64] 1.92	0.72 [0.67 - 0.75] 2.00	0.60 [0.54 - 0.64] 1.94
Qwen-3-Next-80B	Zero-Shot	0.68 [0.65 - 0.71] 1.62	0.64 [0.61 - 0.67] 1.64	0.66 [0.63 - 0.69] 1.61	0.67 [0.64 - 0.70] 1.63	0.54 [0.50 - 0.58] 1.67	0.63 [0.60 - 0.66] 1.62	0.73 [0.70 - 0.76] 1.69	0.65 [0.62 - 0.68] 1.64
	Random Few-Shot	0.72 [0.69 - 0.75] 1.82	0.68 [0.65 - 0.71] 1.84	0.70 [0.67 - 0.73] 1.81	0.71 [0.68 - 0.74] 1.83	0.59 [0.55 - 0.63] 1.87	0.67 [0.64 - 0.70] 1.82	0.77 [0.74 - 0.80] 1.90	0.69 [0.66 - 0.72] 1.84
	Similar Few-Shot Qwen-Embeddings-8B	**0.79** [0.76 - 0.82] 2.12	**0.75** [0.71 - 0.79] 2.14	**0.73** [0.70 - 0.76] 2.11	**0.75** [0.72 - 0.78] 2.13	**0.64** [0.58 - 0.68] 2.18	**0.82** [0.79 - 0.85] 2.12	**0.86** [0.83 - 0.89] 2.20	**0.76** [0.73 - 0.80] 2.14
	Similar Few-Shot Embedding Gemma	0.73 [0.68 - 0.78] 2.00	0.68 [0.64 - 0.71] 2.02	0.69 [0.64 - 0.72] 1.99	0.68 [0.63 - 0.71] 2.01	0.61 [0.56 - 0.66] 2.06	0.78 [0.75 - 0.82] 2.00	0.80 [0.76 - 0.84] 2.08	0.71 [0.67 - 0.75] 2.02
	Similar Few-Shot LaBSE	0.67 [0.62 - 0.71] 1.90	0.64 [0.59 - 0.68] 1.92	0.65 [0.60 - 0.69] 1.89	0.61 [0.56 - 0.65] 1.91	0.56 [0.50 - 0.62] 1.96	0.73 [0.68 - 0.77] 1.90	0.75 [0.70 - 0.78] 1.98	0.66 [0.61 - 0.70] 1.92

Table 4: Qwen-3-Next-80B: thinking vs. non-thinking inference under similarity-based few-shot prompting.

Variant	eng	urd	ita	rus	fas	arb	zho	mean	Energy (10^{-2} Wh)
Non-Thinking	0.79	0.75	0.73	0.75	0.64	0.82	0.86	0.76	**1.98**
Thinking	**0.81**	**0.82**	**0.85**	**0.77**	**0.79**	**0.90**	**0.92**	**0.82**	21.05

resource languages, the performance gap narrows considerably. In several cases, similarity-based few-shot prompting matches or exceeds fine-tuned encoder performance. Fine-tuned encoder systems exhibit stronger sensitivity to language-specific data volume. In contrast, LLM-based systems demonstrate more uniform behavior across languages, suggesting stronger cross-lingual generalization under inference-only adaptation. This pattern indicates that inference-only LLM approaches may be particularly advantageous in multilingual scenarios where per-language labeled data is limited.

5.5 Energy Consumption

Our energy measurement compares task-specific computational cost across paradigms. Fine-tuning requires multiple epochs of training along with optional hyperparameter tuning, resulting in a significant energy consumption. In contrast, LLM-based approaches requires no task-specific training. Although few-shot prompting increases token length and per-instance inference time relative to fine-tuned encoders, total task-specific energy consumption remains substantially lower due to the elimination of training. Per-instance energy consumption is higher for LLM inference than for encoder models. However, this cost is incurred only at deployment and does not compound through repeated training cycles. Moreover, since language and culture undergo continuous shifts over time, avoiding the resource intensive repeated fine-tuning makes LLM pipelines significantly more viable for long-term deployment.

6 Conclusion

In this work, we presented a systematic comparison between fine-tuning of multilingual encoders and inference-only prompting with multilingual large language models for SemEval-2026 Task 9. We evaluated fully fine-tuned and residual-augmented configurations of various SLM backbones alongside various prompting strategies using Qwen. Our results show that similarity-based few-shot prompting achieves performance on par with fully fine-tuned encoders. Although LLM inference incurs higher per-instance energy cost, the overall task-specific energy consumption is substantially reduced due to the absence of task specific retraining. These findings suggest that multilingual LLM prompting is a competitive and computationally efficient alternative to traditional fine-tuning pipelines in task-specific settings. Our study highlights the importance of evaluating adaptation paradigms through predictive performance, computational and environmental considerations.

7 Limitations

Despite the promising results, several limitations must be acknowledged.

First, the performance difference between prompting and fine-tuning is relatively small. While consistent, the margin does not imply a universal superiority of LLM-based methods. Different tasks, label granularities, or domain distributions may yield different outcomes.

Second, our energy analysis focuses on task-specific cost and does not account for the substantial pretraining energy of large language models. Although pretraining cost is amortized across downstream tasks, it remains environmentally significant. Our comparison therefore evaluates downstream adaptation paradigms rather than total lifecycle energy.

Third, inference energy usage and latency for LLM-based methods is higher than that for encoder models. In high-throughput or real-time applications, this latency difference may outweigh the training energy savings.

Fourth, prompting performance is sensitive to prompt design, support example formatting, and retrieval configuration. While we control for prompt variability through multiple runs and similarity-based selection, alternative prompt formulations may yield different results.

Finally, our study evaluates a limited set of encoder and LLM backbones. Broader conclusions would require additional multilingual models across different architectures and parameter scales.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Semen Andreevich Budennyy, Vladimir Dmitrievich Lazarev, Nikita Nikolaevich Zakharenko, Aleksei N Korovin, Olga A Plosskaya, Denis Valer'evich Dimitrov, Vladimir S Akhripkin, Ivan V Pavlov, Ivan Valer'evich Oseledets, Ivan Segundovich Barsola, and 1 others. 2022. Eco2ai: carbon emissions tracking of machine learning models as the first step towards sustainable ai. In *Doklady mathematics*, volume 106, pages S118–S128. Springer.

Jack Choquette. 2023. Nvidia hopper h100 gpu: Scaling performance. *IEEE Micro*, 43(3):9–17.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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 (long and short papers)*, pages 4171–4186.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 878–891.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International conference on machine learning*, pages 4411–4421. PMLR.

Sunbal Iftikhar, Saeed Hamood Alsamhi, and Steven Davy. 2025. Enhancing Sustainability in LLM Training: Leveraging Federated Learning and Parameter-Efficient Fine-Tuning . *IEEE Transactions on Sustainable Computing*, 10(06):1158–1172.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and 1 others. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pages 100–114.

Paul Joe Maliakel, Shashikant Ilager, and Ivona Brandic. 2026. Characterizing llm inference energy-performance tradeoffs across workloads and gpu scaling. *Preprint*, arXiv:2501.08219.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 11048–11064.

Usman Naseem, Robert Geislinger, Juan Ren, Sarah Kohail, Rudy Garrido Veliz, P Sam Sahil, Yiran Zhang, Marco Antonio Stranisci, Idris Abdulmumin, "Ozge Alacam, Cengiz Acar"urk, Aisha Jabr, Saba Anwar, Abinew Ali Ayele, Elena Tutubalina, Aung Kyaw Htet, Xintong Wang, Surendrabikram Thapa, Tanmoy Chakraborty, and 15 others. 2026a. SemEval-2026 task 9: Detecting multilingual, multicultural and multievent online polarization. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*. Association for Computational Linguistics.

Usman Naseem, Robert Geislinger, Juan Ren, Sarah Kohail, Rudy Garrido Veliz, P Sam Sahil, Yiran Zhang, Marco Antonio Stranisci, Idris Abdulmumin, Özge Alacam, Cengiz Acartürk, Aisha Jabr, Saba Anwar, Abinew Ali Ayele, Simona Frenda, Alessandra Teresa Cignarella, Elena Tutubalina, Oleg Rogov, Aung Kyaw Htet, and 24 others. 2026b. Polar: A benchmark for multilingual, multicultural, and multi-event online polarization. *Preprint*, arXiv:2505.20624.

NVIDIA. 2016. NVIDIA Tesla P100 GPU Accelerator Datasheet. https://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-PCIe-datasheet.pdf.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pages 487–503.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2655–2671.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and 1 others. 2021. Xtreme-r: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Communications of the ACM*, 63(12):54–63.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3645–3650.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 70 others. 2025. Embeddinggemma: Powerful and lightweight text representations. *Preprint*, arXiv:2509.20354.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

A Prompts

A.1 Zero-Shot Prompt

```
Polarization refers to increasingly extreme,
↪  divided beliefs or behaviors between
↪  opposing groups. Attitude polarization
↪  includes:
- Negative attitudes toward out-groups
- Blind support for in-groups
- Stereotyping, vilification, dehumanization,
↪  or intolerance

Texts should be labeled:
- Yes: if the message clearly reflects attitude
↪  polarization
- No: if it does not show any polarization
↪  indicators
Note: Always consider the overall context and
↪  meaning, not just individual words.

Tell me whether the following text is
↪  polarizing or not:
"{input_text}"

Answer in the following format:
↪  <answer></answer>
Make sure to include the tag
```

A.2 Few-Shot Prompt

```
Polarization refers to increasingly extreme,
↪  divided beliefs or behaviors between
↪  opposing groups. Attitude polarization
↪  includes:
- Negative attitudes toward out-groups
- Blind support for in-groups
- Stereotyping, vilification, dehumanization,
↪  or intolerance

Texts should be labeled:
- Yes: if the message clearly reflects attitude
↪  polarization
- No: if it does not show any polarization
↪  indicators
Note: Always consider the overall context and
↪  meaning, not just individual words.

You are given the following examples:
Texts exhibiting polarization:
{positive_support_samples}

Text NOT exhibiting polarization:
{negative_support_samples}

Given the above example(s), tell me whether the
↪  following text is polarizing or not:
"{input_text}"

Answer in the following format:
↪  <answer></answer>
Make sure to include the tag
```

B Per Language Performance

	Fine-Tuned		Adapter Module						Baseline
Language	XLM-R	mDeBERTa	XLM-R	mDeBERTa	RemBERT	LaBSE	E-Gemma	QwenEmb	Random
amh	**0.76** **[0.72 - 0.80]**	0.74 [0.69 - 0.79]	0.42 [0.21 - 0.42]	0.21 [0.21 - 0.42]	0.21 [0.21 - 0.23]	0.35 [0.22 - 0.42]	0.42 [0.25 - 0.42]	0.41 [0.37 - 0.43]	0.47 [0.47 - 0.47]
arb	**0.84** **[0.81 - 0.87]**	0.82 [0.79 - 0.85]	0.36 [0.31 - 0.36]	0.36 [0.36 - 0.36]	0.32 [0.31 - 0.36]	0.40 [0.37 - 0.48]	0.36 [0.33 - 0.36]	0.44 [0.43 - 0.47]	0.51 [0.50 - 0.51]
ben	**0.83** **[0.79 - 0.88]**	0.82 [0.78 - 0.86]	0.33 [0.30 - 0.37]	0.30 [0.30 - 0.37]	0.37 [0.32 - 0.37]	0.42 [0.32 - 0.48]	0.33 [0.30 - 0.37]	0.48 [0.43 - 0.52]	0.50 [0.50 - 0.51]
deu	0.71 [0.68 - 0.75]	**0.73** **[0.69 - 0.77]**	0.33 [0.32 - 0.34]	0.33 [0.32 - 0.34]	0.32 [0.32 - 0.34]	0.38 [0.36 - 0.42]	0.33 [0.32 - 0.34]	0.46 [0.41 - 0.48]	0.49 [0.49 - 0.50]
eng	**0.79** **[0.75 - 0.82]**	0.78 [0.75 - 0.81]	0.36 [0.27 - 0.39]	0.39 [0.30 - 0.39]	0.39 [0.29 - 0.39]	0.41 [0.39 - 0.46]	0.33 [0.27 - 0.39]	0.46 [0.44 - 0.49]	0.50 [0.49 - 0.50]
fas	**0.77** **[0.73 - 0.82]**	0.76 [0.71 - 0.80]	0.43 [0.21 - 0.43]	0.21 [0.21 - 0.43]	0.39 [0.24 - 0.43]	0.32 [0.30 - 0.42]	0.43 [0.21 - 0.43]	0.42 [0.33 - 0.49]	0.47 [0.46 - 0.48]
hau	**0.75** **[0.70 - 0.79]**	0.72 [0.67 - 0.77]	0.47 [0.47 - 0.47]	0.44 [0.10 - 0.47]	0.45 [0.31 - 0.47]	0.26 [0.13 - 0.47]	0.10 [0.10 - 0.47]	0.43 [0.26 - 0.48]	0.41 [0.40 - 0.41]
hin	**0.80** **[0.76 - 0.84]**	0.79 [0.75 - 0.83]	0.13 [0.13 - 0.46]	0.46 [0.13 - 0.46]	0.23 [0.13 - 0.46]	0.35 [0.23 - 0.47]	0.46 [0.46 - 0.46]	0.36 [0.29 - 0.46]	0.42 [0.41 - 0.43]
ita	0.66 [0.62 - 0.70]	**0.69** **[0.65 - 0.73]**	0.34 [0.33 - 0.34]	0.32 [0.32 - 0.34]	0.33 [0.32 - 0.34]	0.38 [0.35 - 0.44]	0.34 [0.33 - 0.34]	0.48 [0.45 - 0.54]	0.50 [0.49 - 0.51]
khm	**0.61** **[0.52 - 0.69]**	0.59 [0.50 - 0.66]	0.08 [0.08 - 0.48]	0.42 [0.08 - 0.48]	0.48 [0.08 - 0.48]	0.39 [0.30 - 0.47]	0.28 [0.08 - 0.48]	0.41 [0.37 - 0.49]	0.40 [0.39 - 0.40]
mya	**0.88** **[0.84 - 0.92]**	0.86 [0.82 - 0.90]	0.33 [0.30 - 0.36]	0.36 [0.30 - 0.36]	0.36 [0.34 - 0.38]	0.31 [0.30 - 0.45]	0.36 [0.36 - 0.36]	0.44 [0.41 - 0.46]	0.50 [0.49 - 0.50]
nep	**0.92** **[0.88 - 0.94]**	0.89 [0.85 - 0.92]	0.33 [0.33 - 0.33]	0.33 [0.33 - 0.33]	0.34 [0.34 - 0.36]	0.42 [0.36 - 0.49]	0.33 [0.33 - 0.33]	0.46 [0.40 - 0.48]	0.51 [0.49 - 0.52]
ori	**0.77** **[0.73 - 0.82]**	0.75 [0.70 - 0.80]	0.42 [0.25 - 0.42]	0.42 [0.22 - 0.42]	0.22 [0.22 - 0.22]	0.42 [0.37 - 0.45]	0.42 [0.42 - 0.42]	0.42 [0.38 - 0.48]	0.47 [0.46 - 0.48]
pan	**0.78** **[0.72 - 0.81]**	0.75 [0.70 - 0.79]	0.34 [0.33 - 0.34]	0.34 [0.33 - 0.34]	0.34 [0.33 - 0.37]	0.41 [0.34 - 0.46]	0.33 [0.33 - 0.34]	0.47 [0.40 - 0.51]	0.50 [0.49 - 0.52]
pol	**0.81** **[0.77 - 0.85]**	0.79 [0.75 - 0.83]	0.30 [0.30 - 0.35]	0.30 [0.30 - 0.37]	0.37 [0.30 - 0.37]	0.37 [0.36 - 0.44]	0.30 [0.30 - 0.35]	0.38 [0.35 - 0.47]	0.51 [0.48 - 0.52]
rus	0.80 [0.75 - 0.83]	**0.81** **[0.76 - 0.84]**	0.33 [0.23 - 0.41]	0.41 [0.27 - 0.41]	0.41 [0.41 - 0.41]	0.40 [0.30 - 0.44]	0.32 [0.23 - 0.41]	0.46 [0.43 - 0.49]	0.48 [0.47 - 0.49]
spa	**0.78** **[0.76 - 0.83]**	0.77 [0.74 - 0.80]	0.33 [0.33 - 0.34]	0.34 [0.33 - 0.37]	0.34 [0.33 - 0.37]	0.41 [0.35 - 0.47]	0.34 [0.33 - 0.34]	0.44 [0.38 - 0.48]	0.51 [0.50 - 0.51]
swa	**0.76** **[0.71 - 0.80]**	0.73 [0.67 - 0.78]	0.33 [0.33 - 0.33]	0.33 [0.33 - 0.33]	0.37 [0.35 - 0.45]	0.39 [0.34 - 0.47]	0.33 [0.33 - 0.33]	0.45 [0.44 - 0.49]	0.50 [0.49 - 0.50]
tel	**0.86** **[0.83 - 0.89]**	0.85 [0.82 - 0.88]	0.34 [0.33 - 0.34]	0.34 [0.33 - 0.34]	0.34 [0.33 - 0.34]	0.36 [0.35 - 0.41]	0.33 [0.33 - 0.34]	0.44 [0.40 - 0.49]	0.50 [0.49 - 0.52]
tur	0.80 [0.74 - 0.83]	**0.81** **[0.76 - 0.85]**	0.34 [0.32 - 0.34]	0.34 [0.34 - 0.34]	0.34 [0.33 - 0.36]	0.34 [0.34 - 0.41]	0.32 [0.32 - 0.34]	0.45 [0.41 - 0.50]	0.50 [0.49 - 0.51]
urd	**0.77** **[0.73 - 0.81]**	0.76 [0.71 - 0.80]	0.41 [0.23 - 0.41]	0.41 [0.23 - 0.41]	0.25 [0.24 - 0.41]	0.38 [0.32 - 0.46]	0.41 [0.28 - 0.41]	0.45 [0.33 - 0.51]	0.48 [0.48 - 0.48]
zho	**0.89** **[0.86 - 0.92]**	0.88 [0.84 - 0.91]	0.34 [0.33 - 0.34]	0.34 [0.33 - 0.34]	0.38 [0.34 - 0.42]	0.35 [0.33 - 0.38]	0.33 [0.33 - 0.34]	0.42 [0.38 - 0.43]	0.51 [0.50 - 0.51]
mean	**0.79** [0.75 - 0.83]	0.78 [0.73 - 0.82]	0.34 [0.28 - 0.38]	0.35 [0.27 - 0.38]	0.34 [0.29 - 0.38]	0.37 [0.32 - 0.45]	0.34 [0.30 - 0.38]	0.44 [0.39 - 0.48]	0.48 [0.47 - 0.49]

Table 5: Per model-language macro-F1 for all encoder methods. Each cell reports median (top line) and [first quartile, third quartile] (bottom line). Best scores per language is highlighted in **bold**.

	Qwen-3-8B					Qwen-3-Next-80B				
Language	ZS	RFS	Sim QEmb	Sim EGem	Sim LaBSE	ZS	RFS	Sim QEmb	Sim EGem	Sim LaBSE
amh	0.50 [0.45 - 0.56]	0.49 [0.43 - 0.55]	0.65 [0.59 - 0.68]	0.60 [0.56 - 0.63]	0.54 [0.48 - 0.59]	0.54 [0.47 - 0.60]	0.50 [0.43 - 0.56]	**0.70 [0.64 - 0.74]**	0.67 [0.63 - 0.70]	0.62 [0.57 - 0.65]
arb	0.72 [0.67 - 0.75]	0.71 [0.67 - 0.75]	0.73 [0.69 - 0.76]	0.67 [0.62 - 0.72]	0.61 [0.55 - 0.64]	0.78 [0.74 - 0.82]	0.72 [0.67 - 0.76]	**0.82 [0.79 - 0.85]**	0.78 [0.75 - 0.82]	0.73 [0.68 - 0.77]
ben	0.74 [0.69 - 0.79]	0.73 [0.67 - 0.76]	0.75 [0.70 - 0.79]	0.69 [0.63 - 0.72]	0.65 [0.60 - 0.69]	0.77 [0.72 - 0.81]	0.71 [0.66 - 0.75]	**0.81 [0.77 - 0.84]**	0.77 [0.73 - 0.81]	0.73 [0.69 - 0.77]
deu	0.63 [0.58 - 0.67]	0.62 [0.58 - 0.66]	0.64 [0.59 - 0.67]	0.58 [0.51 - 0.62]	0.53 [0.46 - 0.57]	0.67 [0.63 - 0.71]	0.64 [0.59 - 0.68]	**0.69 [0.65 - 0.72]**	0.66 [0.61 - 0.71]	0.60 [0.54 - 0.64]
eng	0.70 [0.65 - 0.74]	0.71 [0.65 - 0.74]	0.73 [0.69 - 0.77]	0.68 [0.63 - 0.72]	0.64 [0.59 - 0.69]	0.75 [0.72 - 0.78]	0.78 [0.74 - 0.81]	**0.79 [0.76 - 0.82]**	0.73 [0.68 - 0.78]	0.67 [0.62 - 0.71]
fas	0.44 [0.37 - 0.49]	0.43 [0.37 - 0.48]	0.55 [0.49 - 0.59]	0.49 [0.42 - 0.55]	0.45 [0.38 - 0.50]	0.51 [0.45 - 0.56]	0.49 [0.42 - 0.55]	**0.64 [0.58 - 0.68]**	0.61 [0.56 - 0.66]	0.56 [0.50 - 0.62]
hau	0.47 [0.41 - 0.53]	0.46 [0.40 - 0.52]	0.53 [0.46 - 0.57]	0.46 [0.40 - 0.51]	0.39 [0.32 - 0.46]	0.55 [0.49 - 0.61]	0.53 [0.46 - 0.59]	**0.57 [0.50 - 0.62]**	0.52 [0.45 - 0.56]	0.46 [0.40 - 0.51]
hin	0.57 [0.51 - 0.62]	0.56 [0.50 - 0.61]	0.65 [0.61 - 0.69]	0.61 [0.56 - 0.64]	0.56 [0.50 - 0.61]	0.62 [0.56 - 0.67]	0.60 [0.54 - 0.65]	**0.73 [0.69 - 0.77]**	0.70 [0.65 - 0.73]	0.63 [0.58 - 0.68]
ita	0.67 [0.63 - 0.71]	0.66 [0.62 - 0.70]	0.68 [0.64 - 0.71]	0.62 [0.56 - 0.66]	0.55 [0.49 - 0.59]	**0.74 [0.71 - 0.77]**	0.72 [0.68 - 0.75]	0.73 [0.70 - 0.76]	0.69 [0.64 - 0.72]	0.65 [0.60 - 0.69]
khm	0.08 [0.00 - 0.15]	0.11 [0.02 - 0.18]	0.22 [0.14 - 0.28]	0.17 [0.08 - 0.24]	0.11 [0.03 - 0.17]	0.12 [0.05 - 0.19]	0.15 [0.07 - 0.22]	**0.29 [0.18 - 0.36]**	0.23 [0.15 - 0.29]	0.19 [0.12 - 0.27]
mya	0.48 [0.41 - 0.54]	0.55 [0.48 - 0.60]	0.79 [0.75 - 0.82]	0.75 [0.71 - 0.78]	0.70 [0.66 - 0.74]	0.56 [0.50 - 0.61]	0.58 [0.52 - 0.63]	**0.83 [0.79 - 0.86]**	0.76 [0.72 - 0.80]	0.70 [0.64 - 0.73]
nep	0.76 [0.71 - 0.78]	0.75 [0.69 - 0.80]	0.77 [0.73 - 0.80]	0.72 [0.67 - 0.77]	0.68 [0.64 - 0.73]	0.79 [0.75 - 0.83]	0.76 [0.72 - 0.80]	**0.86 [0.83 - 0.89]**	0.80 [0.76 - 0.84]	0.76 [0.71 - 0.78]
ori	0.64 [0.58 - 0.67]	0.63 [0.59 - 0.66]	0.68 [0.63 - 0.72]	0.62 [0.57 - 0.66]	0.58 [0.51 - 0.64]	0.68 [0.62 - 0.72]	0.63 [0.57 - 0.68]	**0.78 [0.73 - 0.82]**	0.71 [0.67 - 0.75]	0.66 [0.61 - 0.70]
pan	0.61 [0.56 - 0.65]	0.60 [0.53 - 0.64]	0.72 [0.68 - 0.76]	0.68 [0.62 - 0.72]	0.61 [0.56 - 0.65]	0.65 [0.60 - 0.70]	0.62 [0.56 - 0.67]	**0.77 [0.72 - 0.81]**	0.71 [0.65 - 0.75]	0.67 [0.63 - 0.72]
pol	0.70 [0.65 - 0.74]	0.69 [0.64 - 0.74]	0.71 [0.67 - 0.76]	0.66 [0.62 - 0.69]	0.60 [0.54 - 0.64]	0.74 [0.71 - 0.77]	0.73 [0.69 - 0.76]	**0.76 [0.73 - 0.79]**	0.69 [0.64 - 0.73]	0.62 [0.58 - 0.65]
rus	0.70 [0.66 - 0.74]	0.70 [0.65 - 0.73]	0.72 [0.67 - 0.77]	0.67 [0.63 - 0.72]	0.61 [0.56 - 0.66]	0.75 [0.71 - 0.79]	**0.77 [0.73 - 0.80]**	0.75 [0.72 - 0.78]	0.68 [0.63 - 0.71]	0.61 [0.56 - 0.65]
spa	0.64 [0.58 - 0.68]	0.65 [0.61 - 0.69]	0.70 [0.64 - 0.75]	0.64 [0.59 - 0.68]	0.60 [0.53 - 0.65]	0.71 [0.67 - 0.75]	**0.75 [0.71 - 0.79]**	**0.75 [0.72 - 0.78]**	0.71 [0.65 - 0.75]	0.64 [0.59 - 0.67]
swa	0.37 [0.29 - 0.44]	0.61 [0.57 - 0.65]	0.63 [0.58 - 0.68]	0.56 [0.50 - 0.61]	0.50 [0.44 - 0.55]	0.44 [0.37 - 0.51]	0.65 [0.59 - 0.70]	**0.68 [0.61 - 0.73]**	0.63 [0.58 - 0.68]	0.56 [0.50 - 0.60]
tel	0.43 [0.37 - 0.47]	0.58 [0.52 - 0.63]	0.64 [0.59 - 0.69]	0.59 [0.54 - 0.64]	0.53 [0.48 - 0.58]	0.52 [0.46 - 0.57]	0.62 [0.56 - 0.67]	**0.69 [0.63 - 0.74]**	0.64 [0.58 - 0.67]	0.60 [0.56 - 0.64]
tur	0.72 [0.67 - 0.77]	0.71 [0.66 - 0.75]	0.73 [0.69 - 0.77]	0.67 [0.61 - 0.70]	0.63 [0.58 - 0.68]	0.76 [0.72 - 0.80]	**0.78 [0.74 - 0.81]**	0.75 [0.72 - 0.78]	0.70 [0.64 - 0.73]	0.64 [0.60 - 0.68]
urd	0.61 [0.55 - 0.65]	0.66 [0.61 - 0.71]	0.71 [0.67 - 0.75]	0.65 [0.61 - 0.68]	0.61 [0.55 - 0.65]	0.69 [0.64 - 0.73]	0.73 [0.68 - 0.77]	**0.75 [0.71 - 0.79]**	0.68 [0.64 - 0.71]	0.64 [0.59 - 0.68]
zho	0.78 [0.73 - 0.81]	0.77 [0.72 - 0.80]	0.82 [0.77 - 0.85]	0.75 [0.71 - 0.78]	0.72 [0.67 - 0.75]	0.81 [0.77 - 0.84]	0.84 [0.80 - 0.87]	**0.86 [0.83 - 0.89]**	0.80 [0.76 - 0.84]	0.75 [0.70 - 0.78]
mean	0.59 [0.53 - 0.63]	0.61 [0.55 - 0.65]	0.67 [0.62 - 0.71]	0.62 [0.56 - 0.66]	0.56 [0.51 - 0.61]	0.64 [0.59 - 0.69]	0.65 [0.60 - 0.69]	**0.73 [0.68 - 0.76]**	0.68 [0.63 - 0.72]	0.62 [0.57 - 0.66]

Table 6: Per-language Qwen results for model size and prompting mode. Abbreviations: ZS = zero-shot, RFS = random few-shot, QEmb = Qwen embeddings, EGem = EmbeddingGemma. Best scores per language is highlighted in **bold**.