MKJ at SemEval-2026 Task 9: A Comparative Study of Generalist, Specialist, and Ensemble Strategies for Multilingual Polarization

Maziar Kianimoghadam Jouneghani
University of Turin
maziar.kianimoghadam@edu.unito.it

Abstract

We present a systematic study of multilingual polarization detection across 22 languages for SemEval-2026 Task 9 (Subtask 1), contrasting multilingual generalists with language-specific specialists and hybrid ensembles. While a standard generalist like XLM-RoBERTa suffices when its tokenizer aligns with the target text, it may struggle with distinct scripts (e.g., Khmer, Odia) where monolingual specialists yield significant gains. Rather than enforcing a single universal architecture, we adopt a language-adaptive selection strategy that chooses among multilingual generalists, language-specific specialists, and hybrid ensembles based on development performance. Additionally, cross-lingual augmentation via NLLB-200 yielded mixed results, often underperforming native architecture selection and degrading morphologically rich tracks. Our final system achieves an overall macro-averaged F1 score of **0.796** and an average accuracy of **0.826** across all 22 tracks. Code and final test predictions are publicly available at: https://github.com/Maziarkiani/SemEval2026-Task9-Subtask1-Polarization.

1 Introduction

Polarization detection—identifying rhetoric that reinforces ideological division—transcends simple sentiment analysis, requiring models to parse complex socio-political nuances. SemEval-2026 Task 9 (Naseem et al., 2026a) introduces the POLAR benchmark (Naseem et al., 2026b) for this task across 22 typologically diverse languages. In this paper, we address **Subtask 1 only**, focusing on multilingual polarization detection across the 22 language tracks of the shared task. The central challenge is an architectural dilemma: balancing the semantic reasoning of massive multilingual generalists against the culturally grounded vocabularies of monolingual specialists.

In this work, we address three central research questions: (**RQ1**) Does XLM-RoBERTa provide a sufficient universal baseline across 22 diverse languages? (**RQ2**) When the baseline fails, under what linguistic conditions do monolingual specialists or alternative high-capacity generalists (e.g., mDeBERTa-v3) outperform it? (**RQ3**) Can cross-lingual data augmentation (via translation) compensate for resource scarcity?

To navigate these questions, we developed a language-adaptive selection strategy. Rather than forcing a universal architecture, we established a systematic empirical policy for architecture selection (detailed in Section 3). Table 1 details our final system configurations, highlighting where generalization succeeds and where specialization is mandatory.

2 Related Work

Polarization detection extends tasks like sentiment and stance analysis by requiring models to identify antagonistic rhetoric across diverse languages, scripts, and contexts. The POLAR benchmark and SemEval-2026 Task 9 formalize this challenge across 22 languages (Naseem et al., 2026a,b).

While multilingual encoders like XLM-R (Conneau et al., 2020) are standard baselines, downstream performance is not dictated by scale alone. Rust et al. (2021) show that monolingual tokenizers often outperform multilingual ones due to varied tokenization quality across languages. This directly motivates our fragmentation analysis and comparison of generalists versus language-specific specialists.

Ensembling is also a recurring strategy in competitive NLP, especially when different models exhibit complementary error profiles. The SemEval-2026 Task 9 overview reports that ensemble prediction, threshold tuning, and weighted fusion were among the most common strategies adopted by

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1398–1406

participating systems, particularly in strong submissions (Naseem et al., 2026a). Similar ensemble-based approaches have also been used in recent SemEval systems for multilingual classification, where combining diverse models improves robustness across heterogeneous inputs (Zhu, 2024).

Finally, while data augmentation addresses low-resource constraints, its efficacy is highly task-dependent. Several participating teams experimented with translation and paraphrasing (Naseem et al., 2026a). However, recent work suggests synthetic data quality is highly method-dependent (Pecher et al., 2026). Consequently, our translation ablation is positioned not as a rejection of augmentation in general, but as evidence that naive forward translation was less reliable than architecture selection in our setting.

3 Methodology

3.1 Phase 1: Architecture Selection Pipeline

We established XLM-RoBERTa-Base (Conneau et al., 2020) as a universal baseline across all 22 tracks. To address the subword fragmentation that standard generalists often suffer in distinct scripts (Appendix A.1), we maintained a comprehensive development ledger and tested between 4 and 13 candidate models or ensemble configurations per track, with deeper search in languages where more resources were available and early candidates failed to meaningfully surpass the baseline.

This policy yielded improvements in 18 of 22 languages (e.g., Odia $+10.6\%$, Khmer $+8.1\%$). However, some tracks required tactical pivots: in **Burmese** and **Punjabi**, standalone specialists underperformed, likely due to domain mismatch between formal pretraining corpora and noisy tweets; in **Spanish**, the specialist (*RoBERTuito*) yielded only a marginal $+0.6\%$ gain, prompting a pivot to mDeBERTa-v3 ($+4.7\%$); and in **Urdu**, specialists failed to separate meaningfully from the baseline, necessitating ensembles. Notably, mDeBERTa-v3's consistent cross-lingual performance also made it a core component in diverse tracks such as **Hindi** and **Persian** (Farsi).

Through this systematic evaluation of monolingual specialists, high-capacity generalists (mDeBERTa-v3), and hybrid ensembles, a non-baseline architecture was adopted only if it improved development Macro-F1 by $\geq 2\%$ or showed a more balanced Precision/Recall profile than the XLM-R baseline. In low-margin cases, this criterion was applied conservatively: small development-set differences were not treated as decisive unless accompanied by a more stable precision–recall trade-off. A representative sample of these transitions is detailed in Appendix A.2.

3.2 Phase 2: Hybrid Ensembles & Calibration

For languages exhibiting complementary error profiles, we constructed weighted soft-voting ensembles to leverage complementary strengths. For instance, in **English**, we combined a domain-specific social media model (BERTweet) with a general-purpose encoder (*DeBERTa-v3-Large* (He et al., 2021)). The ensemble probability is calculated as $P(x) = \alpha P_{Spec}(x) + (1 - \alpha)P_{Gen}(x)$, where $\alpha \in [0, 1]$ denotes the specialist weight in the soft-voting mixture (generalized to an unweighted mean for architectures with > 2 models such as **Punjabi**).

In practice, ensemble configurations were explored as part of the broader development-phase search (Section 3.1). The reported weights therefore correspond to the best-performing tried configurations on the development set rather than to a dedicated local search over α alone. No additional held-out subset was created for α tuning; candidate ensemble variants were compared on the full development set, as the already limited validation sizes made further partitioning impractical.

We used this empirical ensemble selection strategy primarily in languages where different candidate systems exhibited complementary error profiles. Thresholds τ were additionally adjusted in a small number of cases to counteract class bias and restore more balanced decision boundaries in overly aggressive models. Despite the overfitting risks inherent to small validation samples ($N \approx 160$), such targeted calibration was often practically useful. Specifically, the standalone **Hausa** specialist and the ensembles for **Hindi**, **Persian**, **Polish**, and **English** utilized $\tau \in \{0.35, 0.60, 0.60, 0.45, 0.45\}$ respectively for precision optimization or recall support (Table 1). We did not perform a separate systematic sensitivity analysis of α under fixed-model $\pm 10\%$ perturbations.

3.3 Phase 3: Cross-Lingual Augmentation

To address RQ3, we explored cross-lingual data augmentation across 11 language tracks. Using **NLLB-200-Distilled-600M** (Costa-jussà et al., 2022), we translated the complete English training

Lang	Code	Strategy	Model Details	Weights & τ
			Group A: Generalist Sufficiency	
Burmese	mya	Generalist	mDeBERTa-v3 (Base) (He et al., 2021)	-
Nepali	nep	Generalist	mDeBERTa-v3 (Base)	-
Spanish	spa	Generalist	mDeBERTa-v3 (Base)	-
Swahili	swa	Generalist	mDeBERTa-v3 (Base)	-
			Group B: Specialist Superiority	
Arabic	arb	Specialist	AraBERTv02-Twitter (Large) (Antoun et al., 2020)	-
Bengali	ben	Specialist	BanglaBERT (Base) (Bhattacharjee et al., 2022)	-
Chinese	zho	Specialist	MacBERT (Base) (Cui et al., 2020)	-
German	deu	Specialist	GBERT (Base) (Chan et al., 2020)	-
Hausa	hau	Specialist	Davlan/Hausa-XLMR (Alabi et al., 2022)	$\tau = 0.35$
Italian	ita	Specialist	GilBERTo (idb-ita, 2020)	-
Khmer	khm	Specialist	Metythorn/Khmer-XLMR (Base) (Pen, 2025)	-
Odia	ori	Specialist	L3Cube-Odia (Joshi, 2023)	-
Telugu	tel	Specialist	L3Cube-Telugu (Joshi, 2023)	-
			Group C: Hybrid Ensembles	
Amharic	amh	Ensemble	Afro-XLMR (Alabi et al., 2022) + mDeBERTa-v3	40/60
English	eng	Ensemble	DeBERTa-v3 (Large) + BERTweet (Nguyen et al., 2020)	65/35, $\tau = 0.45$
Hindi	hin	Ensemble	mDeBERTa-v3 + L3Cube-Hindi-Hate (Velankar et al., 2021)	50/50, $\tau = 0.60$
Persian	fas	Ensemble	ParsBERT (Farahani et al., 2021) + mDeBERTa-v3	60/40, $\tau = 0.60$
Polish	pol	Ensemble	XLM-R (Base) + HerBERT (Mroczkowski et al., 2021)	50/50, $\tau = 0.45$
Punjabi	pan	Ensemble	mDeBERTa-v3 + XLM-R + MuRIL (Khanuja et al., 2021)	Average
Russian	rus	Ensemble	DeepPavlov (Kuratov and Arkhipov, 2019) + mDeBERTa-v3	50/50
Turkish	tur	Ensemble	Savasy (Yildirim, 2024) + dbmdz (Schweter, 2020)	50/50
Urdu	urd	Ensemble	MuRIL (Base) (Khanuja et al., 2021) + XLM-R (Base)	50/50

Table 1: Final System Configuration. Ensemble weights reflect the best-performing tried development configurations, and thresholds (τ) were adjusted only where beneficial. Unless explicitly stated in the table, we used the default decision threshold of $\tau = 0.5$.

set ($\approx$ 3,200 samples) into each target language and merged these synthetic samples with the native training data. This augmentation strategy was intended to test whether English supervision could provide useful additional signal for lower-resource tracks without requiring language-specific annotation expansion. However, as detailed in Appendix A.3, the results were highly inconsistent, leading us to abandon translation for the final pipeline and prioritize native architecture tuning instead.

4 Experimental Setup

To maximize data, we employed a **Merge-and-Retrain** strategy: after finalizing hyperparameters during the development phase, we concatenated the Train and Dev sets. A 5% internal split was reserved only for epoch-level evaluation logging, allowing us to monitor validation loss and basic convergence behavior during training. This split was not used for early stopping, best-checkpoint reloading, or further hyperparameter tuning; instead, each model was trained for its pre-specified number of epochs, and final predictions were generated from the resulting final model state.

All training utilized mixed precision (`fp16`), a maximum sequence length of 128, and weight decay of 0.01. Per-device batch sizes (2–16) and gradient accumulation steps were adjusted to target an effective batch size of 16 (ranging from 4 to 64). Learning rates ($5e^{-6}$ to $3e^{-5}$) and epochs (4–6) were tuned per model family, and all reported final runs used a single fixed random seed (42) per track. We did not perform a systematic multi-seed evaluation across languages; while this choice improved reproducibility and computational feasibility in a 22-language setting, it also means that seed-level variance—especially on very small development sets ($N \approx 160$)—remains a limitation. Accordingly, we interpret low-margin development differences cautiously, as some architecture selections near the decision boundary may be seed-sensitive.

5 Results and Discussion

5.1 Overall Performance

Our final results on the official test set are presented in Table 2. The system achieved a macro-averaged F1 of **0.796** and an average accuracy of **0.826**.

For context, Appendix A.4 compares the organizer baseline (Naseem et al., 2026a) with our XLM-R development baseline, which anchored Phase 1.

Lang	Acc	F1(B)	F1(M)	Lang	Acc	F1(B)	F1(M)
amh	0.824	0.880	0.774	nep	0.882	0.882	0.882
arb	0.832	0.807	0.829	ori	0.838	0.691	0.790
ben	0.847	0.824	0.845	pan	0.769	0.783	0.768
mya	0.889	0.902	0.887	fas	0.871	0.913	0.831
zho	0.891	0.889	0.891	pol	0.812	0.774	0.806
eng	0.810	0.742	0.796	rus	0.820	0.697	0.784
deu	0.711	0.690	0.710	spa	0.763	0.755	0.763
hau	0.931	0.672	0.817	swa	0.783	0.795	0.782
hin	0.896	0.938	0.803	tel	0.889	0.892	0.889
ita	0.630	0.539	0.615	tur	0.784	0.785	0.784
khm	0.927	0.961	0.711	urd	0.783	0.840	0.751

Table 2: Official Test Phase Results. F1(B) = Binary-F1, F1(M) = Macro-F1, Acc = Accuracy.

5.2 Leaderboard Dynamics and Performance Consistency

To evaluate framework robustness, we compared our results against the best *participant* score on the public leaderboard snapshot (SOTA), defining the performance delta as $\Delta_{SOTA} = S_{Ours} - S_{SOTA}$.[1]

Given the subjectivity of polarization annotation and small development sets (Uma et al., 2021; Mostafazadeh Davani et al., 2022), we interpret minor leaderboard gaps cautiously. We report a descriptive 4-point Macro-F1 window ($\Delta_{\text{SOTA}} \geq -0.04$) to contextualize proximity to the public leaderboard best, rather than claiming definitive architectural superiority. Under this heuristic, our system is within 4 points of the top score on 13 tracks (Table 3), although cases near the cutoff should be interpreted cautiously.

Language	Rank	Our Score	SOTA	Δ_{SOTA}
Persian (fas)	**3**	0.8308	0.8348	**-0.0040**
Burmese (mya)	**2**	0.8874	0.8913	**-0.0039**
Telugu (tel)	7	0.8892	0.9053	-0.0161
Hausa (hau)	6	0.8168	0.8336	-0.0168
Bengali (ben)	7	0.8446	0.8625	-0.0179
Arabic (arb)	12	0.8294	0.8488	-0.0194
Hindi (hin)	17	0.8032	0.8236	-0.0204
Amharic (amh)	10	0.7744	0.8002	-0.0258
Swahili (swa)	15	0.7823	0.8113	-0.0290
English (eng)	18	0.7958	0.8252	-0.0294
Odia (ori)	14	0.7903	0.8255	-0.0352
Polish (pol)	13	0.8061	0.8431	-0.0370
Spanish (spa)	22	0.7632	0.8030	-0.0398

Table 3: Tracks within 4 Macro-F1 points of the public leaderboard best participant submission ($\Delta_{\text{SOTA}} \geq -0.04$). Δ_{SOTA} denotes the raw Macro-F1 difference.

This pattern suggests relatively consistent cross-lingual competitiveness under the above descriptive heuristic, including an official **2nd-place** finish in Burmese and a shared **2nd-place** standing in Persian reported in the official task overview (Naseem et al., 2026a), while the public leaderboard snapshot ranks the raw scores numerically.

Cross-Lingual Stability & Leaderboard Density. Analyzing the public leaderboard snapshot reveals substantial cross-lingual variance among competing systems, which dominate in higher-resource tracks but degrade in distinct-script tracks like Khmer. By switching between generalists and specialists, our approach mitigated cross-lingual overfitting, remaining competitive in 59% of tracks. Resource availability clearly impacts leaderboard density: higher-resource tracks (e.g., English, Spanish) are highly compressed, suggesting standard architectures have reached saturation. Conversely, in distinct-script tracks like **Persian** and **Burmese**, pivoting to a targeted ensemble and high-capacity generalist provided a measurable, highly competitive advantage. Finally, **Italian** proved uniquely challenging; a dense, low-scoring cluster (0.60–0.65) suggests distribution shifts hindered nearly all architectures, with only one isolated system reaching 0.7303.

5.3 Failure Modes and Error Analysis

While our approach proved robust for the majority of languages, we observed distinct failure modes in five tracks where our system underperformed significantly relative to the current public best participant submission ($\Delta_{SOTA} < -0.05$), as detailed in Table 4. Notably, Khmer also illustrates that a positive dev-test shift does not necessarily imply genuine generalization quality.

Language	Macro-F1	SOTA	Δ_{SOTA}
Italian (ita)	0.6149	0.7303	-0.1154
German (deu)	0.7096	0.7608	-0.0512
Punjabi (pan)	0.7679	0.8257	-0.0578
Khmer (khm)	0.7113	0.7744	-0.0631
Urdu (urd)	0.7505	0.8196	-0.0691

Table 4: The "Challenge Tracks": languages where our system underperformed by more than 5 Macro-F1 points relative to the public best participant submission ($\Delta_{SOTA} < -0.05$).

The Recall Trap (Khmer & Urdu): Both models struggled to recover the *neutral* class. Khmer predicted the polarized class for ∼95% of test samples, indicating majority-class collapse. Although

[1]Public SemEval-2026 Task 9 leaderboard snapshot: https://github.com/Polar-SemEval/Leaderboards. Here, SOTA refers to the best non-baseline participant submission in a given language.

exceeding the majority baseline, we treat this as a failure mode. We did not add remedies like focal loss or negative sampling in the final submission to maintain a consistent cross-lingual framework, as they would have required additional track-specific retraining under a limited submission budget, and such adjustments could not be reliably validated on the limited development set.

The Specialist Trap (German): Although the specialist (*GBERT*) clearly outperformed the XLM-R baseline during development, its -0.0512 SOTA gap on the test set exposes the danger of relying on small validation splits ($N \approx 160$). This sharp reversal suggests the model likely overfit to the development data. Because GBERT showed such strong initial results with no obvious warning signs, its +6.95% development gain ultimately proved deceptive. Furthermore, since the German track did not display the extreme class bias seen elsewhere, basic threshold calibration likely would not have resolved this fundamental failure to generalize.

Semantic Overfitting (Punjabi): In **Punjabi**, our Super-Ensemble strategy resulted in a gap of -0.0578. Given the massive parameter count of ensembling three models on an extremely constrained validation split ($N = 100$), we hypothesize the system suffered from semantic overfitting. Rather than learning robust, generalizable decision boundaries, the combined architectures appear to have overfit to development-specific lexical cues, limiting their ability to transfer to the broader test distribution.

A further limitation is that we did not conduct a dedicated weight-sensitivity ablation around the final ensemble ratios; the reported mixtures reflect the selected development-phase configurations rather than a systematic local search over nearby α values.

The Low-Ceiling Track (Italian): Italian showed a different failure mode: the leaderboard was compressed in a relatively low scoring band, with most systems clustering below 0.68 and only one clear outlier at 0.7303. This suggests that the main bottleneck in Italian may lie less in architecture choice alone than in dataset-specific distributional difficulty.

5.4 Dev-Test Shift Analysis

To evaluate system robustness beyond absolute leaderboard position, we compared the best development-phase Macro-F1 scores against the official test results. This comparison helps distinguish genuine cross-split generalization from development-set overfitting, especially in a setting with small validation samples and substantial cross-lingual variation. We analyzed the top three gains, five largest losses, six most stable tracks, and one anomalous gain (detailed in Appendix A.5).

Group 1: Generalization Winners (Gains). Telugu (+3.3%), Burmese (+2.8%), and Polish (+2.3%) exhibited meaningful improvements from development to test. In all three cases, the gains were accompanied by comparatively balanced prediction ratios (41%–56%), suggesting that the selected architectures generalized with robust decision boundaries rather than benefiting from trivial label bias.

Special Case: Anomalous Positive Delta (Khmer). Although Khmer recorded the largest numerical dev-test gain ($+4.1\%$), we do not interpret this as a genuine generalization success. The model predicted the polarized class for 95.6% of test samples, indicating majority-class collapse rather than robust semantic discrimination. We therefore analyze Khmer as a pathological gain case: mathematically, it belongs among the largest positive deltas, but behaviorally it is better understood alongside the Challenge Tracks in Table 4.

The Khmer test set is highly skewed (90.8% polarized). While our 0.7113 score meaningfully exceeds a trivial majority-class baseline (0.4758), the extreme prediction skew confirms weak recovery of the neutral class rather than robust semantic generalization.

Group 2: Stable Performers. Spanish, Arabic, Bengali, Nepali, Amharic, and Chinese showed remarkable stability ($|\Delta| \leq 2.0\%$). For Arabic, Bengali, and Chinese, high-quality monolingual pretraining (e.g., AraBERT) acted as a strong regularizer. For Nepali, stability (+1.2%) highlighted the sample efficiency of mDeBERTa-v3's replaced token detection objective. Finally, Amharic (-0.9%) demonstrated that weighted ensembling effectively mitigates individual model variance.

Group 3: Overfitting Regressions (Losses). The largest drops occurred in Hindi (-8.5%), Italian (-7.8%), Urdu (-7.7%), Persian (-6.6%), and Punjabi (-6.2%). In Italian and Punjabi, models maintained class balance but performance collapsed, suggesting *semantic overfitting* to dev-set lexical artifacts. Conversely, Hindi, Persian, and Urdu suffered from excessive positive predictions (67%–83%), indicating that while threshold cali-

bration ($\tau \geq 0.60$) was necessary, it was insufficient to fully counteract inherent "trigger-happy" biases on these test sets.

Taken together, these patterns suggest that multilingual polarization detection is shaped less by model size alone than by the interaction among tokenizer fit, domain alignment, and calibration under limited supervision.

5.5 Discussion of Research Questions

RQ1: Does XLM-RoBERTa provide a sufficient universal baseline across 22 diverse languages? No. While XLM-RoBERTa offers a convenient starting point, it struggles significantly with distinct or underrepresented scripts. As established in Phase 1, monolingual specialists outperformed the initial XLM-RoBERTa baseline in the vast majority of our 22 tracks (e.g., Odia $+10.6\%$, Khmer $+8.1\%$). Furthermore, our fragmentation analysis (Appendix A.1) suggests XLM-R fragments text up to 38% more than native specialists. This lexical bottleneck suggests that the architectural capacity of a standard generalist often cannot compensate for a lack of specialized vocabulary. Consequently, achieving competitive performance required abandoning the universal baseline in favor of culturally grounded monolingual specialists or alternative high-capacity generalists (e.g., mDeBERTa-v3).

RQ2: Under what linguistic conditions do specialists or alternative generalists outperform the baseline? Our findings suggest specialists excel under two primary conditions: tokenizer efficiency and domain alignment. In **Arabic** (*AraBERT*) and **Khmer**, specialists resolved subword fragmentation bottlenecks *and* aligned with informal social media text. However, localized vocabulary alone is insufficient. In **Burmese** and **Nepali**, native specialists trained on formal literature suffered severe domain mismatch on tweets, losing to mDeBERTa-v3. Furthermore, high-capacity generalists occasionally outperformed even domain-adapted specialists (e.g., *RoBERTuito* in **Spanish**), indicating that architectural depth can sometimes eclipse localized pretraining.

RQ3: Can cross-lingual data augmentation compensate for resource scarcity? In our specific pipeline, exploratory cross-lingual transfer via NLLB-200 yielded mixed results (Appendix A.3). While translating the $\approx 3,200$ English source samples marginally lifted weak XLM-R baselines in **German, Spanish, and Turkish**, these minor gains were ultimately superseded by simply selecting stronger unaugmented architectures. Moreover, morphologically rich languages suffered severe degradation (**Russian** dropped $0.743 \rightarrow 0.684$, **Polish** $0.760 \rightarrow 0.660$), suggesting the translation model struggled to generate the precise inflections required by native tokenizers. While more advanced, large-scale augmentation paradigms may succeed, this naive forward translation proved too unstable for our pipeline, leading us to prioritize native architecture selection.

6 Conclusion

In this study, we addressed multilingual polarization detection across 22 languages through a language-adaptive selection strategy. Our results show that this task resists a one-size-fits-all solution: while XLM-RoBERTa provides a useful universal starting point, strong performance often required language-specific specialists, alternative high-capacity generalists such as mDeBERTa-v3, or weighted hybrid ensembles with targeted threshold calibration. Cross-lingual translation-based augmentation was comparatively unstable, especially in morphologically rich languages. Overall, the findings support language-specific architecture selection as a more reliable strategy than universal model scaling alone or unfiltered synthetic augmentation. This highlights the ongoing necessity for culturally and linguistically grounded NLP solutions.

Limitations

Our study has several methodological and computational limitations. First, architecture selection and threshold calibration relied on small validation splits ($N \approx 160$), so some decisions may be split- or seed-sensitive, and we did not perform multi-seed evaluation. Second, due to submission and computational constraints, we did not conduct local ensemble-weight sensitivity analyses or apply class-imbalance remedies (e.g., focal loss, negative sampling) for heavily skewed tracks such as Khmer or Urdu. Finally, our augmentation experiments relied on a 600M-parameter distilled NLLB-200 model in a simple forward-translation setup. The degradation observed in morphologically rich languages should therefore be interpreted as a limitation of this pipeline, not as a general rejection of multilingual data augmentation.

Acknowledgments

I thank the organizers of SemEval-2026 Task 9. I dedicate the effort and late nights behind this work to my people in Iran, whose resilience and courage continue to inspire me every day.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

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*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resources Association.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2023. Spanish pre-trained bert model and evaluation data. *arXiv preprint arXiv:2308.02976*.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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, Online. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, and 19 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2021. Parsbert: Transformer-based model for persian language understanding. *Neural Processing Letters*, 53(6):3831–3847.

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*.

idb-ita. 2020. Gilberto: An italian pretrained language model based on roberta. GitHub repository.

Raviraj Joshi. 2023. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *Preprint*, arXiv:2211.11418.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha P. Talukdar. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Yuri Kuratov and Mikhail Y. Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.

Dariusz Kłeczek. 2020. Polbert: Attacking polish nlp tasks with transformers. In *Proceedings of the PolEval 2020 Workshop*. Institute of Computer Science, Polish Academy of Sciences. P. 79.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. 2021. Herbert: Efficiently pretrained transformer-based language model for polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kyiv, Ukraine. 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, 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.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Loreto Parisi, Simone Francia, and Paolo Magnani. 2020. Umberto: an italian language model trained with whole word masking. `https://github.com/musixmatchresearch/umberto`. GitHub repository.

Branislav Pecher, Jan Cegin, Robert Belanec, Ivan Srba, Jakub Simko, and Maria Bielikova. 2026. Better as generators than classifiers: Leveraging LLMs and synthetic data for low-resource multilingual classification. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 2840–2857, Rabat, Morocco. Association for Computational Linguistics.

Metythorn Pen. 2025. Xlm-roberta khmer masked language model.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Stefan Schweter. 2020. Berturk – bert models for turkish.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Abhishek Velankar, Hrushikesh Patil, Amol Gore, Shubham Salunke, and Raviraj Joshi. 2021. Hate and offensive speech detection in hindi and marathi. *CoRR*, abs/2110.12200.

Savas Yildirim. 2024. Fine-tuning transformer-based encoder for turkish language understanding tasks. *arXiv preprint arXiv:2401.17396*.

Yuhang Zhu. 2024. Rdproj at SemEval-2024 task 4: An ensemble learning approach for multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 181–187, Mexico City, Mexico. Association for Computational Linguistics.

Dmitry Zmitrovich, Aleksandr Abramov, Andrey Kalmykov, Vitaly Kadulin, Maria Tikhonova, Ekaterina Taktasheva, Danil Astafurov, Mark Baushenko, Artem Snegirev, Tatiana Shavrina, Sergei S. Markov, Vladislav Mikhailov, and Alena Fenogenova. 2024. A family of pretrained transformer language models for Russian. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 507–524, Torino, Italia. ELRA and ICCL.

A Appendix

A.1 Tokenization Fragmentation Analysis

The *Fragmentation Ratio* (average subwords per word) measures tokenizer efficiency. High fragmentation wastes attention on lexical reconstruction over semantic modeling. Table 5 suggests that native specialists reduce fragmentation by up to 38.0% for non-Latin and morphologically rich languages.

Lang	XLM-R	Specialist	Reduction
German	1.60	1.46 (*GBERT*)	8.8%
Polish	1.90	1.58 (*HerBERT*)	16.8%
Persian	1.53	1.17 (*ParsBERT*)	23.5%
Arabic	1.85	1.37 (*AraBERT*)	25.9%
Bengali	1.84	1.14 (*BanglaBERT*)	38.0%

Table 5: Fragmentation Ratio (lower is better). This subset represents complex morphology (German, Polish) and distinct scripts (Persian, Arabic, Bengali).

A.2 Development Architecture Selection

As detailed in Section 3.1, our strategy required a $\geq 2\%$ Macro-F1 improvement to justify pivoting from the XLM-R baseline to an alternative architecture. Table 6 provides a representative snapshot of this development ledger, illustrating where specialist models, stronger generalists, or ensembles cleared that selection threshold.

Lang	Selected Architecture	XLM-R	Selected	Δ_{Dev}
Baseline → Monolingual Specialist				
Odia	L3Cube-Odia	0.7257	0.8317	**+10.60%**
Khmer	Metythorn/Khmer-XLMR	0.5888	0.6696	**+8.08%**
German	GBERT (Base)	0.6700	0.7395	**+6.95%**
Hausa	Davlan/Hausa	0.7834	0.8488	**+6.54%**
Arabic	AraBERTv02 (Large)	0.7972	0.8272	**+3.00%**
Baseline → High-Capacity Generalist				
Spanish	mDeBERTa-v3 (Base)	0.6964	0.7439	**+4.75%**
Nepali	mDeBERTa-v3 (Base)	0.8500	0.8700	**+2.00%**
Baseline → Hybrid Ensemble				
Hindi	mDeBERTa-v3 + L3Cube-Hindi	0.7750	0.8882	**+11.32%**
Turkish	Super-Ensemble	0.7390	0.8075	**+6.85%**
Persian	ParsBERT + mDeBERTa-v3	0.8150	0.8969	**+8.19%**
Amharic	Afro-XLMR + mDeBERTa-v3	0.7162	0.7831	**+6.69%**
Punjabi	XLM-R + MuRIL + mDeBERTa-v3	0.7799	0.8296	**+4.97%**

Table 6: Architecture Selection: Development Macro-F1 comparisons between the initial XLM-R baseline and the adopted architecture.

A.3 Translation Ablation Results

Table 7 compares the XLM-R baseline, the translation-augmented variant, and our final submitted architectures. It provides a compact view of whether synthetic translated data offered meaningful gains over the baseline and how those results compared with the stronger architectures ultimately selected for submission.

Lang	Aug. Model	Baseline	Augmented	Final
Arabic	MARBERT (Abdul-Mageed et al., 2021)	0.797	0.797	**0.829**
German	GBERT-Base	0.670	0.699	**0.710**
Italian	UmBERTo (Parisi et al., 2020)	0.646	0.613	**0.615**
Odia	MuRIL-Base	0.726	0.731	**0.790**
Punjabi	MuRIL-Base	0.780	0.700	**0.768**
Polish	PolBERT (Kłeczek, 2020)	0.760	0.660	**0.806**
Russian	RuBERT (Zmitrovich et al., 2024)	0.743	0.684	**0.784**
Spanish	BETO (Cañete et al., 2023)	0.696	0.702	**0.763**
Swahili	Afro-XLMR	0.779	**0.791**	0.782
Turkish	dbmdz	0.739	0.756	**0.784**
Urdu	MuRIL-Base	0.722	0.705	**0.751**

Table 7: Translation Ablation (Macro-F1). Augmentation destabilized highly inflected languages and underperformed our final architectures in 10 of 11 tracks.

A.4 Organizer Baseline vs. XLM-R Baseline

Table 8 compares the official Subtask 1 LaBSE baseline (Naseem et al., 2026a) with our in-house XLM-R dev baseline, which served as the reference point for our early architecture decisions.

Language	Organizer Baseline	XLM-R (Dev)
Amharic (amh)	0.764	0.716
Arabic (arb)	0.812	0.797
Bengali (ben)	0.825	0.839
Burmese (mya)	0.861	0.857
Chinese (zho)	0.864	0.888
English (eng)	0.773	0.784
German (deu)	0.686	0.670
Hausa (hau)	0.821	0.783
Hindi (hin)	0.782	0.775
Italian (ita)	0.564	0.646
Khmer (khm)	0.737	0.588
Nepali (nep)	0.883	0.850
Odia (ori)	0.776	0.725
Punjabi (pan)	0.749	0.779
Persian (fas)	0.835	0.815
Polish (pol)	0.773	0.759
Russian (rus)	0.748	0.742
Spanish (spa)	0.750	0.696
Swahili (swa)	0.790	0.779
Telugu (tel)	0.889	0.855
Turkish (tur)	0.750	0.739
Urdu (urd)	0.742	0.722

Table 8: Comparison of the organizer-provided baseline and our in-house XLM-R development baseline.

A.5 Dev-Test Shift Analysis

Table 9 reports the best Macro-F1 delta between development and test. To evaluate system robustness, the 15 analyzed tracks are grouped to illustrate the effects of distribution shift, architectural robustness, and threshold calibration.

Language	Dev F1	Test F1	Δ
Special Case: Anomalous Gain			
Khmer (khm)	0.670	0.711	**+4.1%**
Group 1: Top Gains			
Telugu (tel)	0.856	0.889	**+3.3%**
Burmese (mya)	0.859	0.887	**+2.8%**
Polish (pol)	0.783	0.806	**+2.3%**
Group 2: Stable Performers			
Spanish (spa)	0.744	0.763	+1.9%
Nepali (nep)	0.870	0.882	+1.2%
Arabic (arb)	0.827	0.829	+0.2%
Bengali (ben)	0.846	0.845	-0.1%
Amharic (amh)	0.783	0.774	-0.9%
Chinese (zho)	0.911	0.891	-2.0%
Group 3: Top Losses			
Punjabi (pan)	0.830	0.768	**-6.2%**
Persian (fas)	0.897	0.831	**-6.6%**
Urdu (urd)	0.828	0.751	**-7.7%**
Italian (ita)	0.693	0.615	**-7.8%**
Hindi (hin)	0.888	0.803	**-8.5%**

Table 9: Performance Shift: Best Development Phase Macro-F1 vs. Official Test Phase Macro-F1.