

# UTokyo Tsuruoka Lab at SemEval-2026 Task 9: Efficient Single Forward Pass Inference for Multi-Label Polarization Classification

Howard Tangkulung and Yoshimasa Tsuruoka  
The University of Tokyo, Japan  
{howard,tsuruoka}@logos.t.u-tokyo.ac.jp

## Abstract

Detecting and interpreting polarized online content is increasingly crucial as online platforms become central to public information exchange. We present an efficient adaptation of large language models for multi-label polarization classification in SemEval-2026 Task 9: Detecting Multilingual, Multicultural and Multievent Online Polarization. Our single-forward-pass inference method outperforms baseline multi-step decoding approaches for multi-label classification by reducing error propagation while improving inference efficiency. Beyond performance and efficiency analysis, we investigate the cross-lingual transferability of the system, observing statistically significant generalization within language families, a result that offers a practical path for low-resource language adaptation. Our system ranked 1st in 8 languages for Subtask 1 and 6 languages for Subtask 2, and placed in the top 5 for 16 out of 22 languages across both subtasks. Overall, we provide a simple, effective, and efficient solution for multilingual polarization classification.<sup>1</sup>

## 1 Introduction

Online polarization refers to the widening ideological division between groups, which can lead to social friction and conflict (Naseem et al., 2026b). Fueled by inflammatory digital content, these divisions can form echo chambers that reinforce existing beliefs (Garimella, 2018) and, if unaddressed, may erode social cohesion and democratic institutions (Martínez-España et al., 2024). To support global research and mitigation efforts, SemEval-2026 Task 9 (Naseem et al., 2026a) introduced the POLAR benchmark (Naseem et al., 2026b), comprising over 110K instances across 22 languages. In this work, we address two subtasks from the

<sup>1</sup>Our code is available at <https://github.com/howarudo/semeval-2026-task-9-utokyo-tsuruoka-lab>

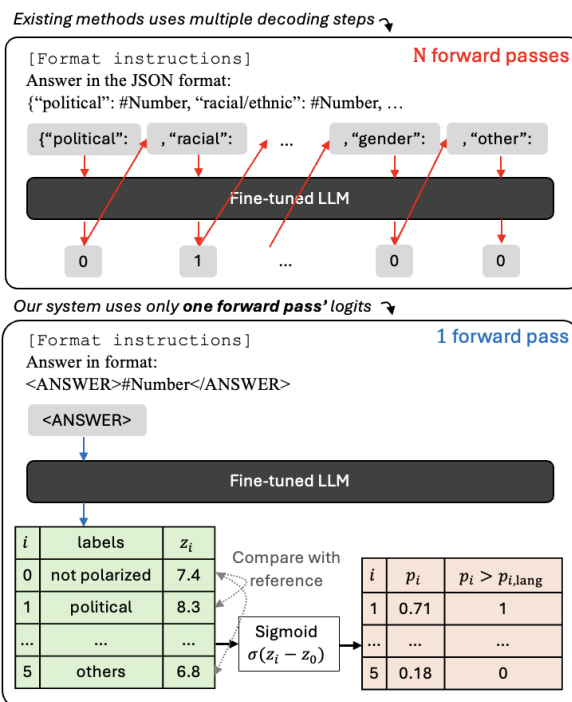


Figure 1: Overview of our system for Subtask 2. **SFP** performs simultaneous multi-label prediction in a single forward pass.

benchmark: (1) Polarization Detection (binary classification), and (2) Polarization Type Classification (multi-label classification).

The main challenge in POLAR is robust generalization across languages, cultural contexts, and events. To address this, our system leverages large language models (LLMs) for their strong multilingual and contextual understanding capabilities. However, many LLM-based multi-label systems rely on multi-step decoding (e.g., sequential label prediction), which compounds prediction errors and increases latency. These drawbacks make such systems costly in real-world, high-traffic environments. To circumvent these limitations, we present a **Single-Forward-Pass (SFP)** inference method that enables the simultaneous prediction of all la-

bels. Compared to multi-step decoding baselines, **SFP** reduces inference time by approximately  $4.5\times$  while improving predictive performance. Furthermore, we analyze cross-lingual transferability in training-free scenarios, demonstrating that performance gains can effectively transfer between languages within the same family or sub-branch.

Our system achieved top-tier performance across the board, ranking 1st in 8 languages for Subtask 1 and 6 languages for Subtask 2. Ultimately, we placed in the top 5 for 16 out of the 22 languages evaluated, highlighting the simplicity and efficiency of our approach for adapting LLMs to multilingual polarization classification.

## 2 Background

### 2.1 POLAR Dataset

POLAR (Naseem et al., 2026b) is a multilingual benchmark for online polarization spanning 22 languages (shown in Table 6), multiple platforms, and cultural events. SemEval-2026 Task 9 (Naseem et al., 2026a) includes three subtasks, and we focus on the following two subtasks:

**Subtask 1: Polarization Detection** Binary classification of whether an instance is polarized.

#### Subtask 2: Polarization Type Classification

Multi-label classification of polarization types (e.g., political, racial/ethnic, religious, gender/sexual, and other).

### 2.2 Related Work

**Online Polarization Classification** Prior work has largely relied on fine-tuned multilingual encoders, such as XLM-RoBERTa (Conneau et al., 2020) and its variants (Zia et al., 2022; Antypas and Camacho-Collados, 2023; Bruno et al., 2024). Evaluation on the POLAR benchmark (Naseem et al., 2026b) indicates that fine-tuned small language models (SLMs) outperform zero-shot LLMs for detection, whereas zero-shot LLMs are stronger for type classification. This suggests that LLMs are better equipped to capture nuanced polarization categories. However, despite the recognized cross-lingual capabilities of LLMs (Tanwar et al., 2023; Jaremko et al., 2025), performance on the POLAR benchmark remains inconsistent across languages within the same family (Naseem et al., 2026b). Such variability motivates our investigation into the cross-lingual transferability of our system within a training-free framework.

**Adapting LLMs for Classification Tasks** LLMs have been successfully adapted to multi-label classification (Muhammad et al., 2025; Xue et al., 2025; Wongso et al., 2025), but common approaches rely on auto-regressive decoding. This sequential approach is prone to error propagation, where an incorrect label prediction can negatively affect subsequent predictions. To address error propagation issues, Xue et al. (2025) utilized a *pairwise* decoding strategy that predicts each label independently. Similarly, Wongso et al. (2025) proposed a *sentence-label pairing* strategy where the model is asked if a label applies to the input sentence. While these methods allow independent label prediction, they introduce a computational bottleneck in which inference costs scale linearly with the number of labels. Building on this challenge, we draw inspiration from SALSA (Berdichevsky et al., 2025), which proposes a single-forward-pass scoring function for binary classification. We extend this method to the multi-label setting by estimating the probabilities of all target labels simultaneously in one forward pass, thereby achieving the independence of pairwise methods without the linear scaling of inference costs.

## 3 System Overview

Given an input text  $x$ , our system outputs either a binary label  $y \in \{0, 1\}$  indicating whether  $x$  is polarized (Subtask 1), or a multi-hot vector  $y \in \{0, 1\}^L$  indicating the presence of each polarization type (Subtask 2), where  $L$  is the number of types. For Subtask 1, we submit only the SALSA-trained model (12B or 27B), while for Subtask 2, we submit only the SFP-trained model (12B or 27B).

### 3.1 Subtask 1: Polarization Detection

**Standard fine-tuning** We embed  $x$  into an instruction prompt that requires the model to generate 1 (polarized) or  $\emptyset$  (not polarized). We fine-tune using causal language modeling and optimize the cross-entropy at the label position:

$$\mathcal{L}_{\text{CE}}(x, y) = -\log \frac{\exp(z_y)}{\sum_{v \in \mathcal{V}} \exp(z_v)}, \quad (1)$$

where  $z_v$  is the logit of vocabulary token  $v$  at the label position and  $\mathcal{V}$  is the vocabulary. At inference time, we predict the label token with the higher logit.

**SALSA** Using SALSA (Berdichevsky et al., 2025), we score the two label tokens  $\emptyset$  and 1 against

Polarization Type	Token
None	$\emptyset$
Political	1
Racial/ Ethnic	2
Religious	3
Gender/ Sexual	4
Other	5

Table 1: Tokens assigned to each polarization type for the SFP method.

each other at the label position. Let  $z_0$  and  $z_1$  denote the logits of  $\emptyset$  and 1 at the label position. We compute:

$$\mathcal{L}_{\text{SALSA}}(x, y) = -\log \frac{\exp(z_y)}{\exp(z_0) + \exp(z_1)}, \quad (2)$$

essentially a binary cross-entropy loss by ignoring non-label tokens. During inference, the probability of polarization is  $P(y = 1 | x) = \frac{\exp(z_1)}{\exp(z_0) + \exp(z_1)}$ , and we compare it to a language-specific threshold  $\tau_{\text{lang}}$  tuned on the validation set.

### 3.2 Subtask 2: Polarization Type Classification

**JSON fine-tuning (JSON FT)** As a multi-pass auto-regressive baseline, we prompt the model to generate a JSON object containing binary values ( $\emptyset/1$ ) for each of the  $L$  types. During training, we compute the cross-entropy loss only on the value tokens for each field and ignore other tokens:

$$\mathcal{L}_{\text{JSON}}(x, y) = -\frac{1}{L} \sum_{i=1}^L \log \frac{\exp(z_{y_i})}{\sum_{v \in \mathcal{V}} \exp(z_v)}, \quad (3)$$

where  $z_{y_i}$  is the logit of the correct value token for type  $i$  at its corresponding position in the output. At inference time, we perform greedy rule-based decoding to extract the predicted values while enforcing the JSON structure and token constraints.

**Single Forward Pass (SFP)** We adapt the SALSA single-forward-pass approach to multi-label classification by treating each label as a separate binary classification problem that shares a common None token for the negative class. We assign a dedicated single-token string to each type and a shared None token (Table 1). Given  $x$ , we obtain logits  $\{z_0, z_1, \dots, z_L\}$  at a fixed classification position, where  $z_0$  corresponds to None and  $z_i$  corresponds to type  $i$ . The model is trained to

score each type token against the None token independently, optimizing the following loss:

$$\mathcal{L}_{\text{SFP}}(x, y) = -\frac{1}{L} \sum_{i=1}^L \log \frac{\exp(z_{y_i})}{\exp(z_0) + \exp(z_i)}, \quad (4)$$

where  $y_i = 0$  if type  $i$  is absent and  $y_i = i$  if type  $i$  is present. At inference time, we predict type  $i$  as present if  $P(y_i = 1 | x) = \frac{\exp(z_i)}{\exp(z_0) + \exp(z_i)} \geq \tau_{\text{lang}, i}$ , where  $\tau_{\text{lang}, i}$  is tuned on the validation set.

## 4 Experimental Setup

**Dataset** We develop and tune our methods using the POLAR dataset (Naseem et al., 2026b). For preliminary experiments, we further split train into internal-train and internal-validation subsets (80/20, stratified by label, fixed seed) for hyperparameter selection and threshold tuning. We then use the official validation split for method comparison and analysis. For the final submission, we train on the full train split, tune thresholds on the validation split, and report results on the test split.

**Models** For method comparison and analysis, we use gemma-3-12b-it.<sup>2</sup> For submission, we train both gemma-3-12b-it and gemma-3-27b-it-bnb-4bit,<sup>3</sup> and select the best-performing model on the validation split (hyperparameters tuned using the internal-validation split) for each language and subtask.

**Training** We fine-tune LLMs with LoRA (Hu et al., 2022) and use unsloth<sup>4</sup> for memory-efficient training. Hyperparameter and training details are in Appendix C.

**Evaluation Metric** We report macro F1 per language, the official metric for both subtasks (Naseem et al., 2026b,a).

## 5 Results

Table 2 shows the results for both subtasks.

### 5.1 Subtask 1: Polarization Detection

**Development Results** Fine-tuning improves performance over the base model, with the largest

<sup>2</sup><https://huggingface.co/unsloth/gemma-3-12b-it>

<sup>3</sup><https://huggingface.co/unsloth/gemma-3-27b-it-bnb-4bit>

<sup>4</sup><https://unsloth.ai/>

#	Lang.	Subtask 1					Subtask 2				
		validation				test	validation				test
		Base 12B	Std. FT 12B	SALSA 12B	SALSA 27B	UTokyo Tsuruoka Lab	Base 12B	JSON FT 12B	SFP 12B	SFP 27B	UTokyo Tsuruoka Lab
	eng	77.48	78.36	78.20	<b>82.55</b>	82.52	40.33	43.77	<b>49.57</b>	45.05	53.22
	deu	77.86	77.97	76.05	<b>80.39</b>	75.31	49.26	52.52	<b>57.39</b>	56.99	62.00
1	urd	74.03	77.14	<b>79.67</b>	79.09	81.96	35.73	<b>79.88</b>	79.46	78.63	77.56
	ben	82.88	84.00	84.69	<b>88.24</b>	86.25	23.56	27.50	30.71	<b>43.41</b>	33.59
	hin	70.06	<b>89.90</b>	86.16	86.16	81.72	44.48	77.74	<b>84.96</b>	80.76	78.25
	ori	59.46	<b>84.08</b>	82.28	82.85	82.55	36.27	49.67	65.07	<b>68.23</b>	60.27
	nep	79.47	<b>89.98</b>	87.98	87.98	91.03	64.38	74.49	<b>76.30</b>	76.16	79.74
	pan	71.43	<b>86.94</b>	85.95	83.68	82.57	29.17	45.38	47.79	<b>49.83</b>	46.77
	spa	67.80	<b>72.66</b>	72.63	70.78	80.30	49.07	62.38	<b>64.16</b>	62.00	67.35
	ita	45.80	61.98	<b>61.98</b>	60.76	61.33	25.40	39.97	<b>40.00</b>	39.48	55.05
	rus	61.61	<b>82.26</b>	80.84	81.78	83.03	44.27	56.85	<b>58.40</b>	56.22	61.66
	pol	67.55	81.03	81.03	<b>85.11</b>	82.59	53.13	60.91	<b>69.33</b>	67.42	64.97
	fas	52.38	83.80	<b>83.99</b>	82.49	80.58	38.97	54.60	<b>61.60</b>	53.53	54.64
2	hau	49.04	77.29	<b>81.33</b>	80.98	78.29	19.31	27.14	29.34	<b>34.33</b>	40.22
	arb	77.58	80.91	82.64	<b>86.88</b>	84.88	45.39	58.18	<b>60.82</b>	59.84	66.78
	amh	72.37	69.74	<b>77.17</b>	76.11	79.54	37.79	44.09	<b>47.96</b>	47.06	62.95
3	zho	82.09	91.58	92.52	<b>92.99</b>	92.89	69.36	80.96	<b>82.36</b>	82.13	83.50
	mya	71.13	87.14	85.03	<b>87.38</b>	88.36	28.81	54.25	56.17	<b>62.36</b>	70.79
4	khm	12.67	62.76	<b>69.77</b>	60.71	72.75	49.88	<b>77.25</b>	69.59	<b>70.18</b>	70.48
5	swa	71.84	<b>83.06</b>	81.92	82.44	79.25	30.52	45.16	48.49	<b>49.74</b>	53.97
6	tel	41.61	88.05	<b>91.52</b>	85.59	88.83	27.65	21.18	<b>42.47</b>	42.21	39.31
7	tur	79.09	85.20	<b>86.04</b>	84.32	83.03	52.67	<b>64.14</b>	63.44	62.37	65.24
	<b>Avg</b>	65.69	80.72	<b>81.34</b>	81.33	81.80	40.70	54.46	58.43	<b>58.56</b>	61.29

Table 2: Macro F1 scores for each method on the validation set, and the official test set results for the best performing model size trained with either SALSA or SFP. The best performing method is **bolded** and the model used for submission is highlighted in **blue**. Languages are grouped following Appendix A.

gains observed in lower-resource languages such as khm and tel. SALSA marginally improves standard fine-tuning in the overall average macro F1 score. We observe only marginal differences between the 12B and 27B variants under SALSA.

**Test Results** For submission, we select the best-performing model per language based on validation performance among SALSA-trained 12B and 27B models. Our system ranked 1st in 8 languages (arb, ben, eng, ori, pan, rus, spa, urd) and in the top 5 in 8 more (amh, deu, hin, mya, nep, pol, tur, zho).

## 5.2 Subtask 2: Polarization Type Classification

**Development Results** As in Subtask 1, fine-tuning improves over the base model. Our SFP method outperforms the JSON auto-regressive baseline in average macro F1 and in 19 out of 22 languages, indicating that simultaneous label scoring is more effective than sequential generation for this task.

**Test Results** For submission, we again select the best-performing model size per language based

on validation results using the SFP method. Our system ranks 1st in 6 languages (deu, eng, ita, khm, ori, pol) and in the top 5 in 10 more (arb, ben, hau, mya, nep, rus, spa, swa, tur, zho).

## 6 Analysis

### 6.1 Threshold Tuning

We tuned classification thresholds  $\tau$  within  $\{0.1, 0.15, \dots, 0.9\}$  to optimize performance. For Subtask 1, we calibrated a language-specific threshold  $\tau_l$ . For Subtask 2, we utilized language- and label-specific thresholds  $\tau_{l,c}$ . Where positive examples were missing for a class  $c$  in language  $l$ , we used the mean threshold from languages  $L'$  that contained positive samples,  $\tau_{l,c} = \frac{1}{|L'|} \sum_{l' \in L'} \tau_{l',c}$ .

Threshold tuning improved the average macro F1 score from 80.81% to 81.34% for Subtask 1 (SALSA 12B) and from 56.34% to 58.43% for Subtask 2 (SFP 12B). As shown in Figure 2, Subtask 2 saw more consistent gains (18/22 languages) compared to Subtask 1 (10/22 languages). One possible explanation for these differing gains is the higher class imbalance in positive samples in Subtask 2 (see Table 7). When class imbalance is more pro-

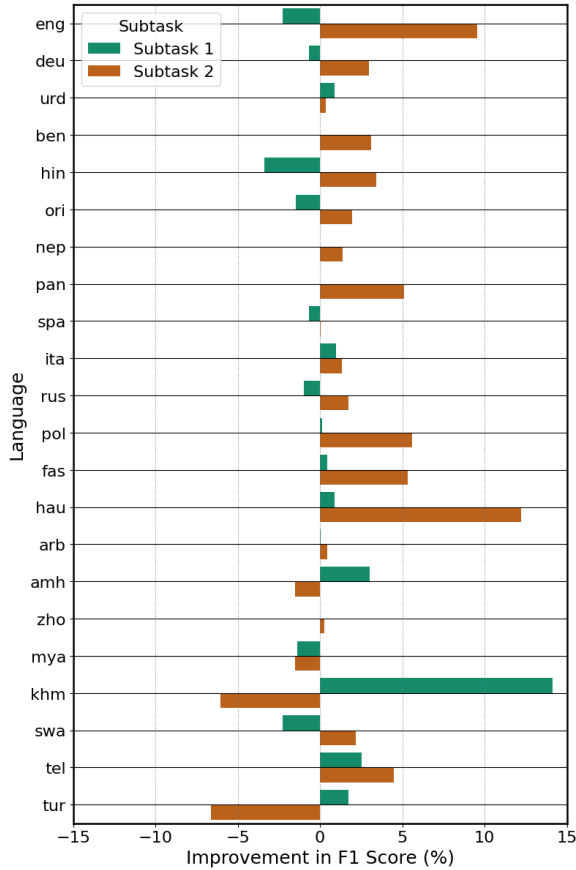


Figure 2: Improvements in macro F1 scores for each language with and without threshold tuning for Subtask 1 and Subtask 2.

nounced, threshold tuning sets a more appropriate decision boundary to maximize the macro F1 score. However, we note that in some languages, threshold tuning did not improve performance, suggesting that its effectiveness may depend on specific dataset characteristics.

## 6.2 Single Forward Pass Inference Speedup

We measure the inference speed of our SFP method relative to the JSON FT baseline on 3,687 instances from the validation split of Subtask 2. With a batch size of 8, JSON FT requires  $2.36 \pm 0.07$  seconds per batch, while SFP requires only  $0.53 \pm 0.08$  seconds per batch. Table 3 shows that SFP improves inference throughput by approximately  $4.5\times$  over JSON decoding. Such a gain is critical for modern polarization detection, where systems must process high volumes of content under strict latency budgets. SFP reduces serving costs and improves deployability in high-traffic settings without compromising performance.

Method	No. of Forward Passes	Throughput (instances/s)
JSON FT	5	3.39
SFP	1	15.24

Table 3: Average inference throughput and number of forward passes for the JSON fine-tuning method and the SFP method, measured on the same hardware and configuration.

## 6.3 Error Analysis for Multi-label Classification

To better understand the differences between JSON FT and SFP, we conducted an error analysis comparing the outputs without threshold tuning (i.e., using a fixed threshold of 0.5) for SFP. Across 14,737 instances in our internal-validation set, the two approaches disagreed on at least one label for 2,174 instances.

We specifically examined whether sequential decoding is more susceptible to error propagation. JSON FT predicts labels in a fixed order (political  $\rightarrow$  racial/ethnic  $\rightarrow$  religious  $\rightarrow$  gender/sexual  $\rightarrow$  other). As shown in Table 4, JSON FT exhibits a higher error rate when at least one preceding label is predicted incorrectly. In contrast, when all preceding labels are correct, the two methods achieve comparable error rates (Table 5).

Method	P(Wrong   At least 1 previous mistake)			
	racial/ethnic	religious	gender/sexual	other
JSON FT	0.22	0.17	0.13	0.27
SFP	0.17 ( $\downarrow 0.05$ )	0.13 ( $\downarrow 0.04$ )	0.10 ( $\downarrow 0.03$ )	0.22 ( $\downarrow 0.05$ )

Table 4: Conditional error rates for each polarization type given that there was at least one previous mistake in the predicted labels. The political label is excluded as it is predicted first for JSON FT.

Method	P(Wrong   All previous correct)			
	racial/ethnic	religious	gender/sexual	other
JSON FT	0.08	0.03	0.03	0.07
SFP	0.08 (0.00)	0.03 (0.00)	0.03 (0.00)	0.08 ( $\uparrow 0.01$ )

Table 5: Conditional error rates for each polarization type given that previous labels were all predicted correctly.

These results suggest that multi-step decoding for multi-label prediction is inherently vulnerable to cascading errors, whereas SFP could mitigate this by predicting all labels simultaneously.

## 6.4 Cross-lingual Transferability within Family Branches / Sub-branches

Naseem et al. (2026b) report that zero-shot LLM performance is not strongly correlated among languages within the same family branch or sub-branch. Here, we explore whether *fine-tuning on a single language* yields transferable gains for Subtask 2 in other languages within the same family branch/sub-branch. For each source language  $\ell_s$ , we fine-tune the model on its internal-train split. For every target language  $\ell_t$ , we tune thresholds on the target language’s internal-validation split and evaluate on the official validation split. We report improvements relative to the base model with threshold tuning but *without* fine-tuning. Figure 7 in Appendix E fully visualizes the resulting improvement matrix.

To quantify the role of linguistic relatedness, we collect all off-diagonal language pairs  $(\ell_s, \ell_t)$  and compare improvement distributions for pairs in the same family branch/sub-branch versus different branches/sub-branches (Figure 3). Language pairs within the same family branch show larger average improvements than pairs from different branches (2.97 vs. 0.79 macro-F1 points). Similarly, pairs within the same family sub-branch show larger improvements than pairs from different sub-branches (4.05 vs. 1.35 macro-F1 points). These differences are statistically significant under both the one-sided Mann–Whitney  $U$  test ( $p < 0.05$ ), and the one-sided Welch’s  $t$ -test ( $p < 0.05$ ), indicating that single-language fine-tuning transfers more effectively to related languages.

This finding could be helpful for adapting our system to new languages with limited training data, but that belong to a family branch/sub-branch with an existing fine-tuned model.

## 7 Conclusion

We introduced an efficient LLM-based system for SemEval-2026 Task 9, centered on **SFP** inference for multi-label polarization type classification. By avoiding sequential auto-regressive decoding, **SFP** mitigates error propagation, substantially reduces inference cost, and improves macro F1 across languages. In addition to performance and efficiency analysis, our cross-lingual analysis suggests that fine-tuning on a single language can transfer to related languages within the same family branch/sub-branch, providing a practical path toward efficient adaptation to low-resource languages.

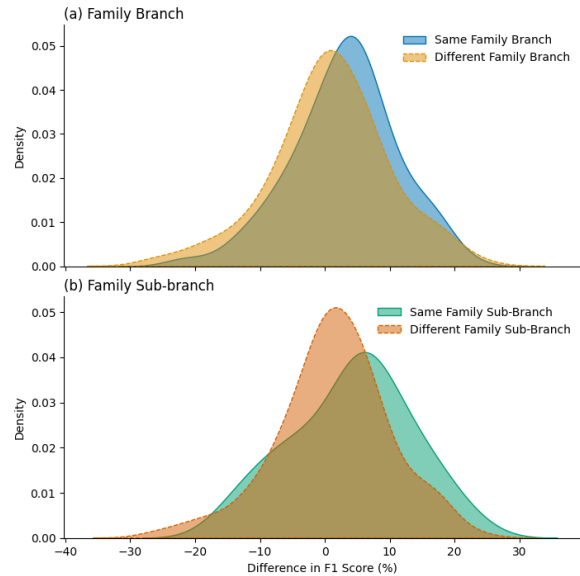


Figure 3: Distribution of macro F1 improvements for off-diagonal language pairs, grouped by whether the pair belongs to the same family branch/sub-branch (top/bottom) or different branches/sub-branches.

## Limitations

Our SFP method is evaluated primarily on Subtask 2 of SemEval-2026 Task 9, and its effectiveness may not directly generalize to multi-label tasks with strong label dependencies where sequential prediction can be beneficial. In addition, our cross-lingual transfer experiments rely on target-language threshold tuning on validation (or internal validation) data, closer to a few-shot setting than a fully zero-shot scenario. We hope these results motivate future work on improving zero-shot cross-lingual transfer, and more broadly on efficient polarization classification with LLMs.

## Ethical Considerations

Polarization classification may support research and moderation, but can also be misused for suppression of legitimate expression or to unfairly target specific groups. We encourage careful evaluation and responsible deployment, especially in high-stakes contexts.

## Acknowledgements

We thank the SemEval-2026 Task 9 organizers for coordinating the shared task and introducing the POLAR dataset. We also thank the anonymous reviewers for their valuable feedback and suggestions.

## References

- Dimosthenis Antypas and Jose Camacho-Collados. 2023. [Robust hate speech detection in social media: A cross-dataset empirical evaluation](#). In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 231–242, Toronto, Canada. Association for Computational Linguistics.
- Ruslan Berdichevsky, Shai Nahum-Gefen, and Elad Ben Zaken. 2025. [Salsa: Single-pass autoregressive llm structured classification](#). *Preprint*, arXiv:2510.22691.
- Mauro Bruno, Elena Catanese, and Francesco Ortame. 2024. [Towards a hate speech index with attention-based LSTMs and XLM-RoBERTa](#). In *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 106–113, Pisa, Italy. CEUR Workshop Proceedings.
- 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.
- Kiran Garimella. 2018. *Polarization on Social Media*. Ph.D. thesis, Aalto University, Finland.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Julia Jaremko, Dagmar Gromann, and Michael Wiegand. 2025. [Revisiting implicitly abusive language detection: Evaluating LLMs in zero-shot and few-shot settings](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3879–3898, Abu Dhabi, UAE. Association for Computational Linguistics.
- Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. [mmbert: A modern multilingual encoder with annealed language learning](#). *Preprint*, arXiv:2509.06888.
- Raquel Martínez-España, Julio Fernández-Pedauey, José Giner-Pérez de Lucía, Jose Miguel Rojo-Martínez, Kaoutar Bakdid-Albane, and Juan José García-Escribano. 2024. [Methodology for Measuring Individual Affective Polarization Using Sentiment Analysis in Social Networks](#). *IEEE Access*.
- Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Seid Muhie Yimam, Jan Philip Wahle, Terry Lima Ruas, Meriem Beloucif, Christine De Kock, Tadesse Destaw Belay, Ibrahim Said Ahmad, Nirmal Surange, Daniela Teodorescu, David Ifeoluwa Adelani, Alham Fikri Aji, Felermينو Dario Mario Ali, Vladimir Araujo, Abinew Ali Ayele, Oana Ignat, Alexander Panchenko, and 2 others. 2025. [SemEval-2025 task 11: Bridging the gap in text-based emotion detection](#). In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 2558–2569, Vienna, Austria. 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 Acarürk, 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)*, San Diego, CA, USA. 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.
- Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. [Multilingual LLMs are better cross-lingual in-context learners with alignment](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6292–6307, Toronto, Canada. Association for Computational Linguistics.
- Wilson Wongso, David Setiawan, Ananto Joyoadikusumo, and Steven Limcorn. 2025. [Lazarus NLP at SemEval-2025 task 11: Fine-tuning large language models for multi-label emotion classification via sentence-label pairing](#). In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 763–772, Vienna, Austria. Association for Computational Linguistics.
- Jieying Xue, Phuong Nguyen, Minh Nguyen, and Xin Liu. 2025. [JNLP at SemEval-2025 task 11: Cross-lingual multi-label emotion detection using generative models](#). In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 20–27, Vienna, Austria. Association for Computational Linguistics.
- Haris Bin Zia, Ignacio Castro, Arkaitz Zubiaga, and Gareth Tyson. 2022. [Improving zero-shot cross-lingual hate speech detection with pseudo-label fine-tuning of transformer language models](#). *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):1435–1439.

## A Language Family and Sub-branches

We follow the language family and sub-branches defined in Naseem et al. (2026b).

#	Lang.	ISO-639	Lang. Family	Sub-branch
1	English	eng	Indo-European	Germanic
	German	deu	Indo-European	Germanic
	Urdu	urd	Indo-European	Indo-Aryan
	Bengali	ben	Indo-European	Indo-Aryan
	Hindi	hin	Indo-European	Indo-Aryan
	Odia	ori	Indo-European	Indo-Aryan
	Nepali	nep	Indo-European	Indo-Aryan
	Punjabi	pan	Indo-European	Indo-Aryan
	Spanish	spa	Indo-European	Romance
	Italian	ita	Indo-European	Romance
	Russian	rus	Indo-European	Slavic
	Polish	pol	Indo-European	Slavic
	Persian	fas	Indo-European	Iranian
	2	Hausa	hau	Afro-Asiatic
Arabic		arb	Afro-Asiatic	Semitic
3	Amharic	amh	Afro-Asiatic	Semitic
	Chinese	zho	Sino-Tibetan	Sinitic
4	Burmese	mya	Sino-Tibetan	Tibeto-Burman
	Khmer	khm	Austroasiatic	Mon-Khmer
5	Swahili	swa	Niger-Congo	Bantu
6	Telugu	tel	Dravidian	Dravidian
7	Turkish	tur	Turkic	Turkic

Table 6: Language family and sub-branch for each language, table adapted from Naseem et al. (2026b).

## B Dataset Label Distribution

Table 7 shows the proportion of positive samples for each label in Subtask 1 and Subtask 2 for each language.

## C Hyperparameters and Training Details

**LoRA Parameters** We fixed the LoRA hyperparameters as in Table 8.

Hyperparameter	Value
Rank ( $r$ )	32
Alpha ( $\alpha$ )	16
Dropout	0.0

Table 8: LoRA hyperparameters tested for LLM fine-tuning.

**Training Hyperparameters** We fixed the hyperparameters in Table 9.

Lang.	Size	Sub. 1	Subtask 2				
		Pol.	Polit.	Race	Relig.	Gen/Sex	Other
eng	4834	0.37	0.36	0.09	0.03	0.02	0.04
deu	4771	0.48	0.41	0.19	0.11	0.06	0.14
urd	5346	0.69	0.67	0.54	0.55	0.51	0.51
ben	5000	0.43	0.34	0.01	0.02	0.01	0.10
hin	4117	0.85	0.74	0.12	0.59	0.11	0.13
ori	3552	0.29	0.21	0.05	0.06	0.03	0.04
nep	3008	0.50	0.17	0.14	0.08	0.05	0.12
pan	2609	0.49	0.31	0.06	0.08	0.11	0.09
spa	4958	0.50	0.27	0.19	0.16	0.13	0.13
ita	5038	0.43	0.08	0.18	0.07	0.09	0.04
rus	5023	0.30	0.14	0.10	0.04	0.06	0.02
pol	3587	0.42	0.37	0.09	0.04	0.05	0.06
fas	4943	0.74	0.44	0.02	0.10	0.06	0.24
hau	5477	0.11	0.05	0.03	0.03	0.01	0.00
arb	5070	0.45	0.24	0.17	0.08	0.11	0.17
amh	4999	0.75	0.67	0.26	0.02	0.01	0.25
zho	6421	0.50	0.06	0.23	0.02	0.17	0.09
mya	4334	0.58	0.25	0.05	0.03	0.11	0.45
khm	9960	0.91	0.18	0.01	0.03	0.02	0.66
swa	10487	0.50	0.03	0.35	0.04	0.02	0.08
tel	3550	0.53	0.22	0.17	0.09	0.13	0.24
tur	3572	0.50	0.44	0.16	0.16	0.06	0.05

Table 7: Proportion of positive samples for each label per language. Labels are abbreviated as follows: Pol. = Polarized, Polit. = Political, Race = Racial/Ethnic, Relig. = Religious, Gen/Sex = Gender/Sexual.

Hyperparameter	Value
Epochs	1
Batch Size	16
Warmup Ratio	0.03
Max Length	1024

Table 9: Fixed hyperparameters used for fine-tuning.

Then, for each method and model size, we test a range of learning rates,  $\{5 \times 10^{-5}, 1 \times 10^{-4}, 3 \times 10^{-4}\}$ , and select the learning rate that achieved the best average macro F1 score across languages on the validation set (Table 10).

Method	Best Learning Rate
Std. FT (12B)	$1 \times 10^{-4}$
SALSA (12B)	$1 \times 10^{-4}$
SALSA (27B)	$5 \times 10^{-5}$
JSON FT (12B)	$1 \times 10^{-4}$
SFP (12B)	$1 \times 10^{-4}$
SFP (27B)	$5 \times 10^{-5}$

Table 10: Best learning rates selected for each method and model size.

**Experiment Environment** All experiments were performed on a single NVIDIA A100-PCI-E-40GB GPU. Experiments with the 12B model were

performed with fp16 precision, and experiments with the 27B model were done with bnb-4bit quantization. All training, quantization, and inference were performed with the unsloth==2025.11.3<sup>5</sup> library. Training and dataset split for internal-validation were performed with a fixed seed of 0 for reproducibility. More details are in the codebase<sup>6</sup>.

## D Prompt Templates

### D.1 Subtask 1

The prompt template used for the base model, standard fine-tuning, and SALSA is shown in Figure 4.

```

Does the following text contain polarization?
A text is polarized if it incites division, hatred, or
stereotyping towards other groups.

Answer in format:
<ANSWER>#Number</ANSWER>
where the number is one of the following:
0 - No
1 - Yes

The text:
<TEXT>
{{text}}
</TEXT>

```

Figure 4: The prompt template used for Subtask 1.

### D.2 Subtask 2

**JSON Fine-tuning** The prompt template used for the JSON fine-tuning method is shown in Figure 5.

**Single Forward Pass** The prompt template used for the SFP pass method is shown in Figure 6.

## E Cross-lingual Transferability Heatmap for Subtask 2

The complete heatmap showing the macro F1 changes for each language when the model is trained on only one language is shown in Figure 7.

## F SLM and LLM Comparison on Subtask 1

During the training phase, when some training data was not yet available, we compared the perfor-

<sup>5</sup><https://pypi.org/project/unsloth/2025.11.3/>

<sup>6</sup><https://github.com/howarudo/semEval-2026-task-9-utokyo-tsuruoka-lab>

```

What type of polarization does the text have?
Political: Extreme focus on division between political
groups.
Racial/ethnic: Focuses on ethnic identity of racial
origin.
Religious: Intolerance between religious groups.
Gender/sexual: Exclusion and marginalization based
on gender.
Other: Hate speech targeting other groups such as
socio-economic status, occupation, location, social
media platform, etc.

```

```

Answer in the JSON format:
{"political": #Number, "racial/ethnic": #Number,
"religious": #Number, "gender/sexual": #Number,
"other": #Number}
Where #Number is 1 if the text contains that type of
polarization, and 0 otherwise.

```

```

The text:
{{text}}

```

Figure 5: The prompt template used for the JSON fine-tuning method in Subtask 2.

mance of an SLM, mmBERT-base (Marone et al., 2025) to the LLM we used, gemma-3-12b-it, trained with the SFP method. For the SLM training, given an input text  $x$ , we extract the pooled representation of the [CLS] token  $\mathbf{h}_{CLS} \in \mathbb{R}^d$  from the last hidden layer of the SLM.

$$\mathbf{x}_{pooled} = \tanh(\mathbf{W}_p \mathbf{h}_{CLS} + \mathbf{b}_p), \quad (5)$$

where  $\mathbf{W}_p \in \mathbb{R}^{d \times d}$  and  $\mathbf{b}_p \in \mathbb{R}^d$ . This pooled output is passed through a final linear classification head with  $\mathbf{W}_c \in \mathbb{R}^{2 \times d}$  and  $\mathbf{b}_c \in \mathbb{R}^2$  to compute the class logits:

$$\mathbf{z} = \mathbf{W}_c \mathbf{x}_{pooled} + \mathbf{b}_c. \quad (6)$$

The SLM model was trained using cross-entropy loss, and the predicted label  $\hat{y}$  is obtained by taking the arg max of  $\mathbf{z}$ . We trained the SLM for {2, 4} epochs, with a learning rate of  $\{1 \times 10^{-5}, 3 \times 10^{-5}\}$ , and a batch size of 32. As shown in Table 11, we found that the SLM performed similarly or worse than the LLM on all languages, and decided to focus on the LLM for submission.

## G Qualitative Output Comparison: JSON FT vs. SFP

To complement the error analysis in Section 6.3, we present example predictions from the JSON fine-tuning method and the SFP method for Subtask 2 in Table 12. Examples #1–#2 show cases where SFP performed better than JSON FT. Example #3 shows a case where JSON FT performed better than SFP.

What type of polarization does the text have?  
 Political: Extreme focus on division between political groups.  
 Racial/ethnic: Focuses on ethnic identity of racial origin.  
 Religious: Intolerance between religious groups.  
 Gender/sexual: Exclusion and marginalization based on gender.  
 Other: Hate speech targeting other groups such as socio-economic status, occupation, location, social media platform, etc.

Answer in format:  
 <ANSWER>#Number</ANSWER>  
 where the number is one of the following:  
 0 - Not polarized  
 1 - Political  
 2 - Racial/ethnic  
 3 - Religious  
 4 - Gender/sexual  
 5 - Other (socioeconomic status, technology, social media preferences, etc.)

The text:  
 <TEXT>  
 {{text}}  
 </TEXT>

Figure 6: The prompt template used for the SFP method in Subtask 2.

Lang.	mmBERT	gemma-3-12b-it (SFP)
eng	79.33	80.61
deu	71.69	77.35
urd	72.30	72.59
hin	74.74	84.32
nep	86.99	86.99
spa	67.24	73.05
ita	62.66	63.43
fas	82.51	84.63
hau	73.56	77.44
arb	77.91	82.11
amh	67.61	77.61
zho	86.44	93.92
tur	77.38	84.34
Avg	75.41	79.88

Table 11: Comparison of macro F1 scores (in %) on the validation set for Subtask 1 between the SLM (mmBERT) and the LLM (gemma-3-12b-it).

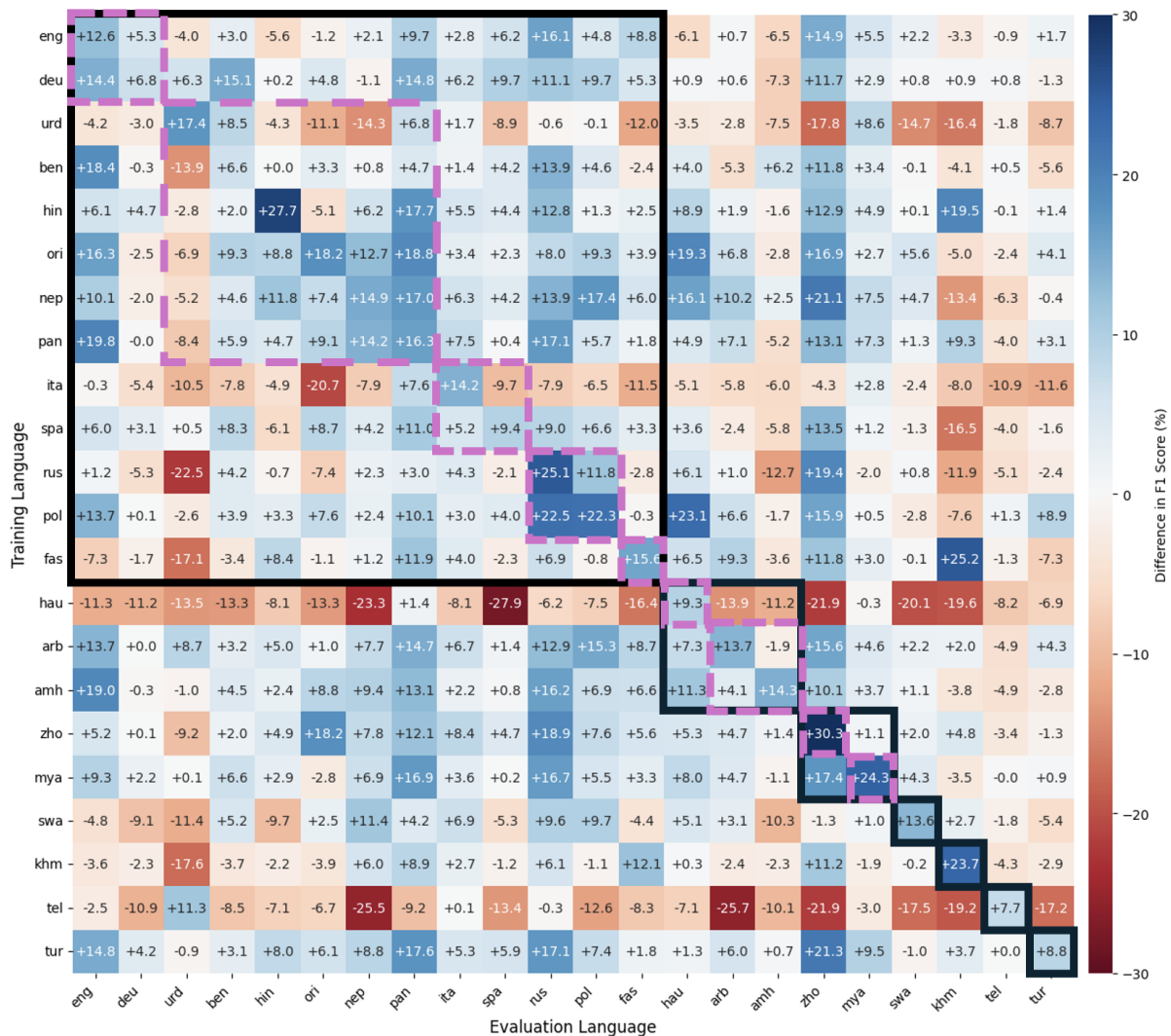


Figure 7: Heatmap showing the macro F1 improvements over the base model for each language when the model is trained on only one language. **Bolded frames**: same family, **purple dashed frames**: same sub-branch.

#	Text ID	Gold	JSON FT	SFP
1	pan_521032e35bec0f1f389152175610565d	[1, 1, 1, 0, 0]	[0, 0, 0, 0, 0] [0.25, 0.29, 0.32, 0.03, 0.05]	[0, 1, 1, 0, 0] [0.29, 0.75, 0.50, 0.05, 0.08]
2	rus_259608de63e3b821f043a094ac79d6d2	[0, 0, 0, 0, 0]	[1, 1, 0, 0, 0] [0.71, 0.56, 0.00, 0.00, 0.02]	[1, 0, 0, 0, 0] [0.75, 0.41, 0.00, 0.00, 0.03]
3	amh_e1e50a90f7b0331004a54f93393da560	[1, 0, 0, 0, 1]	[1, 0, 0, 0, 1] [0.93, 0.09, 0.00, 0.01, 0.62]	[1, 0, 0, 0, 0] [0.96, 0.09, 0.00, 0.00, 0.35]

Table 12: Example predictions (top row) and  $P(y_i=1 | x)$  (bottom row) for JSON fine-tuning vs. SFP. The labels represent the 5 polarization types in order: political, racial/ethnic, religious, gender/sexual, and other.