Detecting Multilingual, Multicultural and Multievent Online Polarization: Task 9 POLAR @ SemEval-2026

Aatman Vaidya
University of Tübingen
aatman-vrundavan.vaidya@student.uni-tuebingen.de

Abstract

some abstract

1 Introduction

Online polarization leads to hostility between social, political, or identity groups and has also been linked to real world violence (Zelalem and Guest, 2021). Online polarized text often includes hate speech, toxicity, sarcasm, misogyny, profanity, and other forms of harm. The **POLAR @ SemEval-2026 Task 9** focuses on detecting multilingual, multicultural and multi-event online polarization[1].

The task covers 22 languages and contains the following 3 subtasks.

- **Subtask 1 Polarization Detection:** It is a binary classification task to determine whether a post contains polarized content (Polarized or Not Polarized).
- **Subtask 2 Polarization Type Classification:** The task is to classify the type or target polarization for a given text. It is a multi-label classification task to identify the target of polarization as one of the following categories: Political, Racial/Ethnic, Religious, Gender/Sexual or Other.
- **Subtask 3 Manifestation Identification:** It is a multi-label classification task to classify how polarization is expressed, with multiple possible labels including Stereotype, Vilification, Dehumanization, Extreme Language, Lack of Empathy", or Invalidation.

In this report, I summarize the progress made so far. My work to date has focused on Subtask 1. I reviewed relevant literature on training and fine-tuning classification models, with particular attention to low-resource languages. I also conducted an exploratory analysis of the dataset provided by the organizers and fine-tuned several existing pre-trained encoder-only models on this data.

[1] https://polar-semeval.github.io/

2 Dataset

The dataset is constructed by aggregating multiple resources related to hate speech, toxicity, polarization, and other forms of online harm across different languages. It consists of textual data collected from social media and online platforms, including news websites, Reddit, blogs, Bluesky, and regional forums. The content spans a range of topics such as elections, conflicts, gender rights, and migration. Each language includes approximately 3,000–5,000 annotated instances. Overall, the task covers 22 languages representing diverse cultural and geographical contexts: Amharic, Arabic, Bengali, Burmese, Chinese, English, German, Hausa, Hindi, Italian, Khmer, Nepali, Odia, Persian, Polish, Punjabi, Russian, Spanish, Swahili, Telugu, Turkish, and Urdu. The dataset statistics for Subtask 1 are reported in Table 1.

Statistic	Value
Training data	73,681
Dev data	3,687
Positive class samples (label = 1)	39,145
Negative class samples (label = 0)	34,536
Positive class ratio	0.5313
Negative class ratio	0.4687

Table 1: Dataset Statistics

For model training, I partitioned the training dataset provided by the organizers into training, validation, and test sets using a 90%–5%–5% split. This choice was motivated by the fact that the official development set constitutes $\approx 5\%$ of the training data; accordingly, I adopted a similar proportion for both the validation and test splits. The data were stratified to preserve overall class bal-

ance across the splits. However, maintaining an equal distribution of languages within each class was not feasible due to the inherent imbalance in the original dataset.

3 Methodology and Experiments

3.1 Experimental Setup

This section outlines the methodology adopted for subtask 1. Looking at prior work (Ragab et al., 2025), encoder-only language models (LMs) have been widely used for natural language understanding (NLU) tasks, including classification, and have achieved state-of-the-art performance (Marone et al., 2025). Multilingual extensions of these models, such as mBERT (Devlin et al., 2019) and XLM-RoBERTa (XLM-R) (Conneau et al., 2020), further enable cross-lingual modelling. Recent analyses indicates that encoder-only models are significantly more effective for classification tasks than decoder-only models (e.g., large language models) at comparable parameter sizes, and can outperform decoder models that are an order of magnitude larger (Weller et al., 2025; Gisserot-Boukhlef et al., 2025). Hence I decided to fine-tune encoder-only models on the given dataset.

Parameter	Value
Rank (r)	4
LoRA α	16
Target modules	query, key, value
LoRA dropout	0.05
Bias	none
Task type	SEQ_CLS

Table 2: LoRA PEFT hyperparameters used for fine-tuning encoder models.

The models were selected based on prior work (Plaza-del Arco et al., 2023; Aluru et al., 2020). Specifically, six well-performing multilingual encoder models were chosen: XLM-RoBERTa-base (Conneau et al., 2020), mBERT-base (Devlin et al., 2019), mDeBERTa-base (He et al., 2021), GTE-multilingual-MLM-base (Zhang et al., 2024), MMBERT-base (Marone et al., 2025), and Glot500-base (Imani et al., 2023). For fine-tuning the selected encoder models, I used Low-Rank Adaptation (LoRA) as a parameter-efficient fine-tuning (PEFT) technique (Hu et al., 2022). Table 2 lists the LoRA-specific hyperparameters used. The training process was configured with standard training arguments, including batch sizes, learning rate, number of epochs, evaluation strategy, and early stopping, as detailed in Table 3. Model performance was evaluated using the accuracy metric, and early stopping was applied to prevent overfitting.

Parameter	Value
Number of epochs	30
Train batch size	16
Evaluation batch size	16
Learning rate	2e-5
Optimizer	AdamW_Torch
Gradient accumulation steps	2
Evaluation strategy	Steps
Evaluation steps	500
Early stopping patience	3 evaluations

Table 3: Training hyperparameters used for fine-tuning the encoder models.

3.2 Results

Table 4 presents the evaluation results of the six encoder models on both the custom test set and the official development set. **F1-macro** is the official error metric used by the competition organisers, hence I report both accurary and f1-macro scores for all models. I have also listed error metric scores for the 5% test set I create for training and the main development set provided my the organisers.

Across models, MMBERT-base achieved the highest performance on the main development set, with an F1-macro of 0.724 and mDeBERTa-base 0.7834 on the class-balanced test set. In general, all models achieved decent accuracy scores, ranging from 0.7395 to 0.7834 on the test set. The performance trends were largely consistent between the two evaluation sets, with mDeBERTa-base and GTE-multilingual-MLM-base also showing competitive results. I have reported the language specific scores for all modes in Table 5 in the Appendix.

The F1-macro scores differ between the class-balanced test set and the main development set. This difference arises because the official development set reflects the natural class distribution of the dataset, which is imbalanced, whereas the test set was stratified to maintain equal representation of each class. Consequently, the F1-macro scores on the balanced test set are generally higher, highlighting the effect of class imbalance on the metric. Overall, these results indicate that the chosen encoder models perform well across languages, with

Model Name	My Test Set		Main Dev Set	
	Accuracy	F1 Macro	Accuracy	F1 Macro
XLM-RoBERTa-base	0.759	0.756	0.756	0.693
mBERT-base	0.76	0.758	0.735	0.660
mDeBERTa-base	0.77	0.768	0.78	0.724
GTE-multilingual-MLM-base	0.765	0.7624	0.745	0.664
MMBERT-base	0.7834	0.7815	0.768	0.708
Glot500-base	0.7395	0.735	0.738	0.67

Table 4: Average Scores across all languages

MMBERT-base and mDeBERTa-base emerging as the most consistent models for Subtask 1.

4 Conclusion and Future Work

In this report, I presented the progress made on Subtask 1 of the POLAR @ SemEval-2026 challenge. I fine-tuned several multilingual encoder-only models using LoRA-based parameter-efficient adaptation and evaluated them on both a class-balanced test set and the official development set. The results indicate that MMBERT-base and mDeBERTa-base are the most consistent performers across languages, achieving competitive F1-macro scores.

Building on this, I want to take the following approaches in my **Future Work**.

Approach 1: Exploring zero-shot and few-shot classification using state-of-the-art large language models (LLMs), including Qwen, Mistral, and LLaMA (Kumar et al., 2024; Melis et al., 2025; Ranjan et al., 2025). For few-shot experiments, examples from the training data will be provided for each language. I also want to explore advanced prompting techniques such as chain-of-thought (CoT) prompting to encourage step-by-step reasoning in LLMs (Kumar et al., 2024) (although, with recent reasoning models this might not be needed).

Approach 2: Leveraging strong embedding models from recent LLMs (e.g., Qwen and Gemma) to create custom MLP classifiers trained on top of these embeddings. Such custom approaches would also given us a chance to address class imbalance by incorporating class-weighted loss functions, such as weighted cross-entropy, to penalize errors on minority classes more heavily during training and fine-tuning.

Approach 3: I want to try different model ensemble approaches, like majority or weighted voting among fine-tuned encoder models, or combining predictions from encoders via stacking or averaging logits, and incorporating confidence calibration techniques such as Platt scaling to handle varying model uncertainties.

Approach 4: I want to use synthetic data generation methods to generate data for low-resource languages (Nguyen, 2024), especially the under performing ones. And use this to expand the training data for encoder-only models. Or use good quality synthetic generated data as examples for few-shot prompting classification.

References

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

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.

Hippolyte Gisserot-Boukhlef, Nicolas Boizard, Manuel Faysse, Duarte M Alves, Emmanuel Malherbe, André FT Martins, Céline Hudelot, and Pierre Colombo. 2025. Should we still pretrain encoders

with masked language modeling? *arXiv preprint arXiv:2507.00994*.

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*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André FT Martins, François Yvon, and 1 others. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. *arXiv preprint arXiv:2305.12182*.

Deepak Kumar, Yousef Anees AbuHashem, and Zakir Durumeric. 2024. Watch your language: Investigating content moderation with large language models. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 18, pages 865–878.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*.

Matteo Melis, Gabriella Lapesa, and Dennis Assenmacher. 2025. A modular taxonomy for hate speech definitions and its impact on zero-shot llm classification performance. *arXiv preprint arXiv:2506.18576*.

Luan Thanh Nguyen. 2024. How good is synthetic data for social media texts? a study on fine-tuning low-resource language models for vietnamese. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, pages 871–884.

Flor Miriam Plaza-del Arco, Debora Nozza, and Dirk Hovy. 2023. Respectful or toxic? using zero-shot learning with language models to detect hate speech. In *The 7th workshop on online abuse and harms (woah)*, pages 60–68.

Mohamed Ibrahim Ragab, Ensaf Hussein Mohamed, and Walaa Medhat. 2025. Multilingual propaganda detection: Exploring transformer-based models mbert, xlm-roberta, and mt5. In *Proceedings of the first International Workshop on Nakba Narratives as Language Resources*, pages 75–82.

Rishabh Ranjan, Likhith Ayinala, Mayank Vatsa, and Richa Singh. 2025. Multimodal zero-shot framework for deepfake hate speech detection in low-resource languages. *arXiv preprint arXiv:2506.08372*.

Orion Weller, Kathryn Ricci, Marc Marone, Antoine Chaffin, Dawn Lawrie, and Benjamin Van Durme. 2025. Seq vs seq: An open suite of paired encoders and decoders. *arXiv preprint arXiv:2507.11412*.

Zecharias Zelalem and Peter Guest. 2021. Why facebook keeps failing in ethiopia. *Rest of World*.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *arXiv preprint arXiv:2407.19669*.

A Language Specific Performance

Language	xlm-roberta-base		m-bert-base		m-deberta-base		m-gte-multilingual-mlm-base		glot-500 -base		mm-bert -base	
	Acc	F1 Macro	Acc	F1 Macro	Acc	F1 Macro	Acc	F1 Macro	Acc	F1 Macro	Acc	F1 Macro
Amharic	0.747	0.5356	0.7229	0.4196	0.7952	0.6856	0.7349	0.4448	0.7349	0.482	0.7771	0.6082
Arabic	0.7751	0.7751	0.7574	0.7565	0.7633	0.7626	0.7692	0.7681	0.7574	0.7569	0.7811	0.78
Bengali	0.8193	0.818	0.741	0.7303	0.8072	0.8044	0.8012	0.798	0.7651	0.7636	0.8494	0.8459
German	0.717	0.7168	0.6918	0.6918	0.7233	0.723	0.6604	0.6603	0.6792	0.679	0.7044	0.7037
English	0.7562	0.7208	0.7812	0.7543	0.7875	0.7602	0.7625	0.7101	0.7562	0.7208	0.775	0.7506
Persian	0.7805	0.7073	0.8232	0.7494	0.7683	0.6746	0.7622	0.6342	0.7744	0.674	0.8476	0.7916
Hausa	0.8681	0.6301	0.8956	0.665	0.8736	0.6697	0.9011	0.708	0.8516	0.6123	0.8901	0.6131
Hindi	0.8759	0.7562	0.8467	0.6861	0.8759	0.7562	0.8394	0.6482	0.8686	0.7122	0.8248	0.6162
Italian	0.5723	0.5607	0.5783	0.562	0.5843	0.5809	0.6024	0.5804	0.5904	0.5843	0.5602	0.5531
Khmer	0.9036	0.4747	0.9096	0.5075	0.9006	0.5023	0.9006	0.4739	0.9036	0.4747	0.9036	0.504
Burmese	0.75	0.7402	0.7431	0.7363	0.8056	0.8009	0.7778	0.7677	0.75	0.737	0.8056	0.8009
Nepali	0.81	0.8084	0.68	0.6768	0.83	0.8292	0.79	0.7874	0.79	0.7864	0.83	0.8286
Odia	0.7797	0.6878	0.6864	0.5373	0.7712	0.6623	0.6949	0.41	0.7797	0.7156	0.7203	0.6775
Punjabi	0.71	0.6991	0.72	0.7172	0.75	0.7498	0.68	0.6604	0.66	0.6511	0.72	0.7196
Polish	0.7311	0.7224	0.6639	0.6531	0.8235	0.8147	0.6555	0.6454	0.6723	0.6558	0.6807	0.6779
Russian	0.7485	0.7068	0.7605	0.6767	0.7425	0.6982	0.7305	0.6554	0.7365	0.686	0.7844	0.7459
Spanish	0.6606	0.6596	0.6727	0.6718	0.703	0.7029	0.6667	0.6662	0.6242	0.6202	0.6364	0.636
Swahili	0.6676	0.6625	0.7163	0.7156	0.7335	0.7333	0.7135	0.7115	0.6562	0.6509	0.7622	0.7612
Telugu	0.6949	0.6946	0.6441	0.644	0.7542	0.7502	0.6525	0.6525	0.661	0.6606	0.6949	0.6941
Turkish	0.7043	0.7038	0.6522	0.6515	0.8	0.7998	0.7217	0.7212	0.713	0.7116	0.7391	0.7391
Urdu	0.7119	0.6173	0.6949	0.5192	0.7175	0.6098	0.7458	0.6672	0.7062	0.5873	0.7571	0.6723
Chinese	0.8458	0.8458	0.7944	0.7944	0.8598	0.8596	0.8364	0.836	0.8084	0.8081	0.8505	0.8505

Table 5: Language Performance for all 22 languages