UIT-Polar at SemEval-2026 Task 9: Detecting Multilingual, Multicultural and Multievent Online Polarization

Doan Nguyen Tran Hoan, Nguyen Trong Chinh
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
hoandnt.20@grad.uit.edu.vn, chinhnt@uit.edu.vn

Abstract

We present a two-stage hybrid system for SemEval-2026 Task 9 on multilingual and multievent online polarization detection. The first stage employs DeBERTa for high-recall binary filtering to mitigate severe class imbalance. The second stage leverages Mistral for fine-grained polarization classification, enabling improved semantic reasoning over candidate instances. This coarse-to-fine design enhances robustness and efficiency while preserving minority-class performance. UIT-Polar system achieves Top-5 results on the English test set, demonstrating the effectiveness of integrating encoder-based screening with LLM-based refinement.

1 Introduction

Online polarization is increasingly common on social media, especially around political, cultural, and event-related topics. SemEval-2026 Task 9 (Naseem et al., 2026b) focuses on detecting polarized content across different languages and events. This is challenging due to linguistic variation, cultural differences, and context. In addition, most data is non-polarized, creating a strong class imbalance. Detecting polarization is important for content moderation and understanding online discussions.

The UIT-Polar system uses a two-stage approach. First, DeBERTa (He et al., 2023) is used as a binary classifier to filter out non-polarized posts, helping reduce noise and handle imbalance. Then, Mistral (Jiang et al., 2023) performs fine-grained classification on the filtered data. This setup combines efficiency with deeper contextual understanding, improving detection of minority polarized cases.

Results show that this two-stage method is more stable under imbalanced and multilingual data. The system ranked **Top-5** on the English test set. It improves minority-class recall compared to single-stage models. However, it performs best on clear polarization and struggles with subtle, sarcastic, or culturally nuanced expressions.

2 Background

In SemEval-2026 Task 9, we participated in Subtask 1 (Naseem et al., 2026a): *Polarization Detection (True/False)*, formulated as a binary classification problem. Given a social media post, the system must determine whether it contains polarized content (True = 1) or not (False = 0). A post is labeled as polarized only if it clearly reflects attitude polarization in its overall meaning and context, rather than based on isolated keywords. For example, strongly antagonistic or divisive statements toward a group are considered polarized, whereas neutral reporting or factual descriptions are labeled as non-polarized.

The shared task covers multiple languages and events, reflecting multilingual and multicultural online discourse. In this work, we focus exclusively on the English track. The organizers provide annotated training data for this subtask, and the dataset exhibits substantial class imbalance, with non-polarized instances forming the majority class. System performance is evaluated using Macro F1-score, which emphasizes balanced performance across both classes.

3 UIT-Polar: A Two-Stage Polarization Detection System

3.1 Overall Architecture

The **UIT-Polar** system adopts a two-stage coarse-to-fine architecture for binary polarization detection. Given an input post x, the system predicts a binary label $y \in \{0, 1\}$, where 1 denotes polarized content and 0 denotes non-polarized content.

Rather than relying on a single classifier, UIT-Polar decomposes the task into two complementary modules:

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 127–132

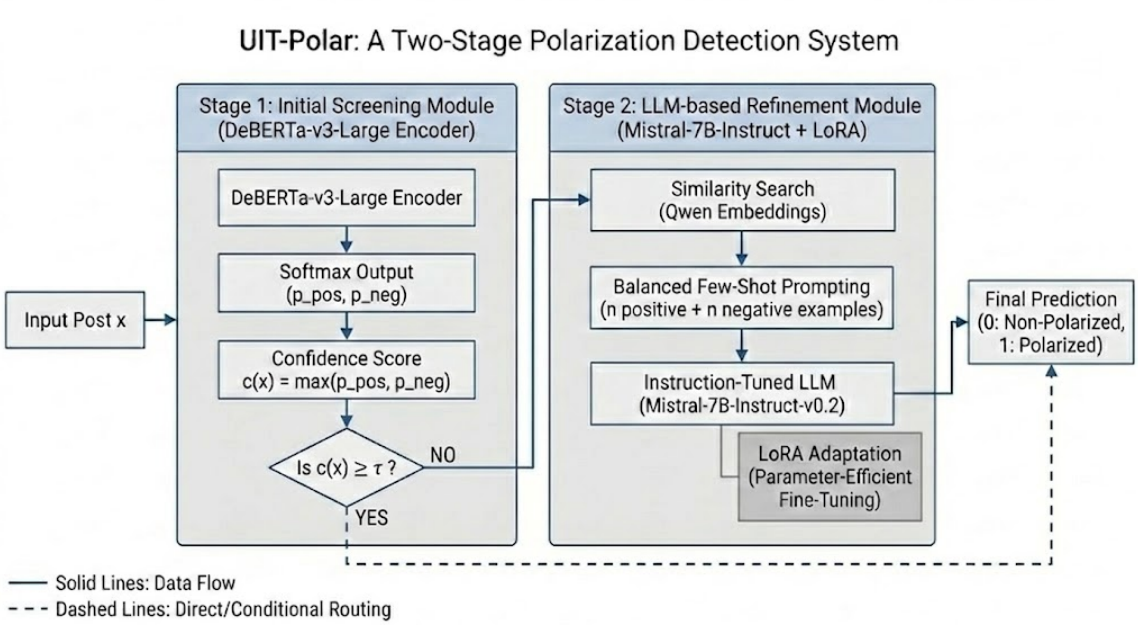

Figure 1: Overview of the UIT-Polar two-stage architecture. The Initial Screening Module directly classifies high-confidence instances, while low-confidence inputs are routed to the LLM-based Refinement Module. The final prediction is a binary label (0: Non-Polarized, 1: Polarized).

- **Initial Screening Module (Stage 1):** a DeBERTa-based encoder that performs high-recall polarization screening and directly classifies high-confidence instances.
- **LLM-based Refinement Module (Stage 2):** an instruction-tuned large language model that processes low-confidence or ambiguous cases requiring deeper semantic reasoning.

The interaction and information flow between modules are illustrated in Figure 1

3.2 Stage 1: Initial Screening Module

The first stage of UIT-Polar system performs high-recall polarization screening using a DeBERTa-based encoder. Given an input post x, the model outputs a probability distribution over the binary labels $y \in \{0, 1\}$ (non-polarized, polarized):

$$p(y \mid x) = \text{Softmax}(W h_{[\text{CLS}]} + b), \qquad (1)$$

where $h_{[\text{CLS}]}$ denotes the contextual representation of the input sequence produced by DeBERTa, and W, b are trainable parameters. The model is fine-tuned on the provided English training data using cross-entropy loss.

Confidence-Aware Routing. For an input x, the Initial Screening Module produces class probabilities $p_{\text{pos}}(x)$ and $p_{\text{neg}}(x)$. The confidence score is defined as:

$$c(x) = \max\left(p_{\text{pos}}(x), p_{\text{neg}}(x)\right). \qquad (2)$$

Given a threshold τ, the final prediction is:

$$\hat{y} = \begin{cases} \arg\max_{y \in \{0,1\}} p_y(x), & \text{if } c(x) \geq \tau, \\ f_2(x), & \text{otherwise}, \end{cases} \qquad (3)$$

where $f_2(x)$ denotes the output of the LLM-based Refinement Module. High-confidence instances are classified directly for efficiency, while uncertain cases are forwarded for refined reasoning.

Design motivation. The dataset exhibits substantial class imbalance, with non-polarized instances dominating the distribution. Training a single-stage classifier tends to bias predictions toward the majority class, negatively affecting Macro F1. By introducing a confidence-based screening mechanism, Stage 1 acts as a lightweight filter that confidently classifies easy cases while deferring ambiguous instances to a more expressive large language model in Stage 2. This design improves computational efficiency and stabilizes minority-class performance. The threshold τ is selected on the development set to balance coverage and precision. Increasing τ reduces the number of instances accepted by Stage 1

and increases reliance on Stage 2, while decreasing τ improves efficiency but may propagate uncertain predictions.

3.3 Stage 2: LLM-based Refinement Module

The second stage performs refined classification on instances that are deemed uncertain by Stage 1. We employ an instruction-tuned Mistral model as a generative binary classifier. The model is fine-tuned using a fixed instruction-style prompt to align generation with the polarization detection objective.

Instruction fine-tuning. Instruction tuning has been shown to significantly improve the ability of large language models to follow task-specific instructions and generalize to unseen tasks (Ouyang et al., 2022; Wei et al., 2022; Chung et al., 2024). Each training instance is formatted using a fixed prompt template:

> You are a specialized AI for detecting ATTITUDE POLARIZATION in social media text.
>
> ### DEFINITION:
> Polarization is the process where opinions/behaviors become extreme, leading to hostility and "us vs. them" division. It is characterized by:
> - Stereotyping, vilification, dehumanization, or intolerance of others' views/identities.
> - Blind support for an in-group while inciting hatred/conflict toward an out-group.
>
> ### CLASSIFICATION CRITERIA:
> - Label 1 (Polarized): If the text clearly incites division, groupism, or displays extreme negative attitudes toward a specific group.
> - Label 0 (Non-Polarized): If the text is neutral, factual, a general insult without group-based division, or a descriptive analysis of social issues.
>
> ### CRITICAL RULES:
> 1. Ignore the reader's possible reaction.
> 2. Ignore opinions of third parties mentioned in the text.
> 3. Do NOT infer hidden intent; judge only what is explicitly written.
> 4. If the polarization is ambiguous or not clearly "Attitude Polarization", you MUST default to 0.
>
> ### OUTPUT FORMAT:
> - Return ONLY the digit 0 or 1.
> - No explanation, no intro, no punctuation.
>
> Text: "{x}"
> Output:

The target output is constrained to either `Polarized` or `Non-Polarized`. Fine-tuning is conducted on the English training data to adapt the model to the task-specific label space while preserving its general reasoning ability. To enable parameter-efficient adaptation, we fine-tune the Mistral model using Low-Rank Adaptation (LoRA) (Hu et al., 2021), updating only a small set of rank-decomposed matrices while keeping the backbone parameters frozen.

Retrieval-based Few-shot Inference. During inference, we augment the fixed instruction template with dynamically retrieved in-context examples (Zhao et al., 2021; Liu et al., 2022). For each input instance x, we compute its dense representation using a Qwen embedding model (Zhang et al., 2025) and retrieve a ranked list of candidate training samples based on cosine similarity. To reduce class imbalance effects in in-context learning, we enforce a balanced selection strategy.

Let n denote the number of positive–negative example pairs (i.e., `num_pairs`). We select the top-n most similar positive samples and the top-n most similar negative samples from the retrieved pool. The final few-shot block therefore contains $2n$ examples.

To mitigate label priming bias, the selected examples are arranged in alternating order (Pos–Neg–Pos–Neg–. . .) before being prepended to the fixed instruction template. The final prompt is structured as follows:

```
### EXAMPLES:

Text: {pos_1}
Output: 1

Text: {neg_1}
Output: 0

...

Text: {pos_n}
Output: 1

Text: {neg_n}
Output: 0

Text: "{query}"
Output:
```

The hyperparameter n controls the number of in-context demonstrations used during inference.

Final prediction. The model generates a textual label, which is mapped to binary output (1 for po-

larized, 0 for non-polarized). By restricting generation to the two valid label tokens, we reduce decoding ambiguity.

Design motivation. Stage 2 focuses exclusively on ambiguous cases passed from Stage 1. The combination of instruction fine-tuning and balanced retrieval-based few-shot prompting enhances semantic reasoning and reduces majority-class bias. Empirically, this refinement stage improves Macro F1 by better handling subtle and context-dependent polarization signals that are difficult for encoder-only classifiers.

4 Experimental Setup

All experiments are implemented using **PyTorch** and the **HuggingFace Transformers** library. Parameter-efficient fine-tuning with LoRA is implemented using the **PEFT** framework. Experiments are conducted on NVIDIA GPUs with mixed-precision training.

4.1 Data Splits

We follow the official train–test split provided by the shared task. For Stage 1, we further reserve 10% of the training data as an internal validation set for model selection and threshold tuning. This validation split is sampled from the training data and is not used for final evaluation. The validation set is used for early stopping and for selecting the confidence threshold τ in the two-stage framework.

4.2 Preprocessing

For Stage 1 (DeBERTa), input texts are tokenized using the HuggingFace tokenizer corresponding to `microsoft/deberta-v3-large`. We set the maximum sequence length to 128 tokens. Sequences longer than this limit are truncated, and shorter sequences are padded to the maximum length within each batch.

For Stage 2 (LLM refinement), texts are formatted into an instruction-style prompt with optional retrieval-based few-shot examples. No additional normalization or text simplification is applied in order to preserve the original semantic cues.

4.3 Model Configuration

4.3.1 Stage 1: DeBERTa-based Screening

We fine-tune `microsoft/deberta-v3-large` for binary classification using a batch size of 16 and a learning rate of 5×10^{-6}. The model is trained for 3 epochs using the AdamW optimizer. The best checkpoint is selected based on validation performance.

The classifier outputs class probabilities p_{pos} and p_{neg}. Predictions with $\max(p_{\text{pos}}, p_{\text{neg}}) \geq \tau$ are accepted directly, while uncertain instances are forwarded to Stage 2.

4.3.2 Stage 2: Mistral with LoRA Adaptation

We use **Mistral-7B-Instruct-v0.2** (Jiang et al., 2023), a 7-billion-parameter decoder-only Transformer with 32 layers and 4096 hidden dimensions.

To enable parameter-efficient adaptation, we apply Low-Rank Adaptation (LoRA) with rank $r = 32$, scaling factor $\alpha = 64$, and dropout rate 0.05. LoRA modules are injected into attention projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and the `gate_proj` module, while backbone parameters remain frozen.

We configure LoRA with rank $r = 32$, scaling factor $\alpha = 64$, and dropout rate 0.05. LoRA layers are applied to the attention projection modules (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and the `gate_proj` module.

Only LoRA parameters are updated during training, while the backbone model remains frozen.Training is conducted with a batch size of 8 using a standard causal language modeling objective under the instruction-tuning format.

4.4 Evaluation Metrics

We report Accuracy and Macro-F1 as the primary evaluation metrics. Macro-F1 is particularly important due to class imbalance in the dataset.

All experiments are implemented using PyTorch and the HuggingFace Transformers library, with LoRA adaptation implemented via the PEFT framework.

5 Results

5.1 Main Results

Table 1 presents the official test set results according to the shared task evaluation protocol. Our two-stage framework achieves strong performance in terms of Accuracy and Macro-F1, outperforming the official baseline.

System	Macro-F1
Baseline	78.02
Our Two-Stage System	**81.62**

Table 1: Results on the official test set.

In the final competition ranking, UIT-Polar system placed **5-th** among all participating teams.

5.2 Ablation Study

Effect of Few-shot Prompting (Pretrained LLM). We first evaluate the impact of retrieval-augmented few-shot prompting using the *pretrained* Mistral model without LoRA fine-tuning. All results in this section are reported on the development set.

Setting	Macro-F1
Zero-shot (pretrained LLM)	80.61
Few-shot (1+1)	81.32
Few-shot (2+2)	81.43
Few-shot (3+3)	**82.54**
Few-shot (4+4)	82.01
Few-shot (5+5)	81.65

Table 2: Impact of balanced few-shot prompting using the pretrained LLM (Dev set).

Few-shot prompting consistently improves performance over zero-shot inference, demonstrating that in-context demonstrations help calibrate the model's decision boundary. We observe that using three positive–negative pairs yields the best results, while increasing the number of examples beyond this point does not consistently improve performance, possibly due to prompt length saturation.

Effect of LoRA Fine-tuning. We further evaluate the impact of instruction fine-tuning with LoRA on the Mistral model.

Model	Macro-F1
Pretrained LLM (Zero-shot)	80.61
Pretrained LLM (Few-shot 3+3)	82.54
Fine-tuned LLM (Few-shot 3+3)	**84.83**

Table 3: Impact of LoRA fine-tuning on development set performance.

LoRA adaptation provides additional gains beyond in-context learning, suggesting that parameter-efficient fine-tuning helps the model better internalize the task definition and decision criteria.

5.3 Analysis of the Two-Stage Framework

We analyze the behavior of the confidence-based routing mechanism. On the development set, Stage 1 directly classifies approximately 70% of instances with confidence above threshold $\tau = 0.8$, while the remaining 30% are forwarded to Stage 2.

Among the forwarded instances, Stage 2 improves Macro-F1 by approximately 6.93 points compared to Stage 1 alone. This indicates that the refinement stage is particularly effective for ambiguous or borderline cases.

System Variant	Macro-F1 (Dev)
Stage 1 only (DeBERTa)	79.64
Stage 2 only (LLM)	84.83
Two-Stage Framework	**86.57**

Table 4: Comparison of system components on development set.

The two-stage framework achieves 86.57 Macro-F1, outperforming both Stage 1 (79.64) and Stage 2 (84.83) when used independently. This improvement suggests that the two models exhibit complementary strengths. Stage 1 provides reliable predictions for high-confidence instances, while Stage 2 refines ambiguous cases through instruction-based reasoning and contextual understanding. By routing only uncertain instances to the LLM, the framework effectively reduces error propagation and leverages the strengths of both discriminative and generative modeling paradigms.

6 Limitations

Our system uses a strong classification strategy, but the threshold τ is difficult to tune. This leads to a drop in performance on the test set compared to the validation set (81.62 vs 86.57). Since the number of test submissions is limited to 5, we cannot effectively tune this hyperparameter.

7 Conclusion

We propose a two-stage framework that combines a DeBERTa-based classifier with an instruction-tuned LLM for attitude polarization detection. The confidence-aware routing mechanism enables the system to leverage the efficiency of a discriminative encoder while refining ambiguous cases through instruction-guided reasoning.

Experimental results show that this hybrid approach consistently outperforms single-stage baselines. However, the overall performance is sensitive to the confidence threshold τ, which controls the trade-off between precision and LLM refinement. Careful tuning of τ is therefore essential to fully realize the benefits of the two-stage design.

References

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations (ICLR)*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. 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, "Ozge Alacam, Cengiz Acart"urk, Aisha Jabr, Saba Anwar, Abinew Ali Ayele, Elena Tutubalina, Aung Kyaw Htet, Xintong Wang, Surendrabikram Thapa, Tanmoy Chakraborty, and 15 others. 2026a. SemEval-2026 task 9: Detecting multilingual, multicultural and multievent online polarization. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*. Association for Computational Linguistics.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

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

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.