CPS Lab 1@MMLow 2026: Hierarchical Multimodal Classification of Tamil Political Memes via Logit-Conditioned Stance-Target Prediction

Vishal Kumar[1,*], Anika Srivastava[1,†], Harshita Sharma[1,‡], Nitisha Aggarwal[1,§], Geetika Jain Saxena[2,¶], Amit Pundir[2,‖], Sanjeev Singh[1,]**

[1]Institute of Informatics and Communication, University of Delhi
[2]Department of Electronics, Maharaja Agrasen College, University of Delhi

[*]researchwithvishal@gmail.com, [†]anikasrivastava151@gmail.com, [‡]harshitasharma@south.du.ac.in, [§]nitisha@south.du.ac.in, [¶]gsaxena@mac.du.ac.in, [‖]amitpundir@mac.du.ac.in, [**]sanjeev@south.du.ac.in

Abstract

Political memes have become an effective tool for online political communication and engagement. Their interpretation is particularly challenging in low-resource and code-mixed languages such as Tamil, where memes often include Tamil-English, use culturally relevant lexicon, incorporate political entities, and utilize references to prominent personalities and parties. This paper presents our submission to the MMLow shared task at AACL-IJCNLP 2026 on multi-level classification of Tamil political memes, which requires jointly predicting (i) a coarse-grained stance label (Support vs. Troll) and (ii) a fine-grained target label identifying whether the meme concerns a political party or a person. We proposed a lightweight hierarchical classifier built on frozen, pretrained multimodal encoders: images are represented using CLIP (ViT-B/32) and text using the multilingual BGE-M3 embedding model, with the two modalities combined via early, non-parametric concatenation fusion. A shared encoder feeds two prediction heads, where the target-identification head is explicitly conditioned on the stance head's output logits via a gradient-detachment mechanism, allowing the model to respect the task's label hierarchy without corrupting the coarse-level classifier's gradients. To address pronounced class imbalance in both label levels, we applied balanced loss weighting at each level. On the test set of unlabeled data, our system achieved 3rd place, with macro-F1 scores of 0.856 for stance classification and 0.532 for target identification, yielding a final score of 0.694, indicating promising performance on the Tamil political meme classification problem.

1 Introduction

Social media has become a battleground for opinions, and memes have emerged as one of the most powerful tools for expressing political views on them. By combining images, text, symbols, and contextual information, memes allow for communicating ideas effectively to a broad audience that may not fully understand the underlying message's linguistic meaning. The control on these memes becomes increasingly challenging in low-resource and code-mixed language scenarios. In social media meme posts where both Tamil and English are used together. This creates a barrier for monolingual or English-only NLP pipelines to perform well.

The Shared Task on Multi-Level Political Meme Classification, organized as part of MMLow@AACL-IJCNLP 2026 (Rajiakodi et al., 2026), addresses this challenge by proposing a benchmark for automated culturally aware analysis of code-mixed Tamil-English political memes. A sample of the dataset is shown in Figure 1. This shared task is structured as a multi-level classification problem:

- **Level 1 - Stance Identification:** Identify whether a meme expresses support or trolling.
- **Level 2 - Target Identification:** Determine whether the meme concerns a political party or a person, conditioned on Level 1 (e.g., support for a party under a "support" label or troll against a person under a "troll" label).



Figure 1: Sample of the original dataset representing different labels at each stance and target level

Proceedings of the Fifth Workshop on Multimodal Machine Learning in Low-resource Languages, pages 97–102
November 9, 2026 ©2026 Association for Computational Linguistics

This hierarchical structure reflects the natural organization of political meme discourse, in which the target of a meme is dependent on its stance. Accurate predictions require this dependency to be modeled, rather than approached as two independent classification tasks, in order to obtain labels that are internally consistent with the hierarchical structure defined by the shared task.

Our system combines frozen pretrained multimodal encoders: images are represented using CLIP (ViT-B/32) (Radford et al., 2021), and text using the multilingual BAAI/bge-m3 embedding model (Chen et al., 2024), with a jointly trained hierarchical classifier. In this classifier, predictions at Level 2 (target) depend directly on those at Level 1 (stance) through a gradient-detached logit-conditioning mechanism. This mechanism captures the label hierarchy structurally while maintaining separate optimization paths for the two output layers.

2 Related Work

Multimodal meme analysis has attracted increasing interest due to memes' ability to encode information through the combination of text, images, symbols, and context. (Suryawanshi et al., 2020) proposed the Tamil Memes dataset for troll classification, which emphasizes the importance of addressing the challenges of multimodal Tamil meme classification.

Prior work by (Das et al., 2022) investigated multimodal and ensemble methods for classifying Tamil troll memes, finding that representation learning benefits significantly from combining text and vision. More recently, the task of multimodal meme analysis has been extended to the detection of misogynistic Tamil and Malayalam memes, with promising results obtained with pretrained language and vision models (Ratul et al., 2025).

The MMLow 2026 shared task (Rajiakodi et al., 2026) extends previous Tamil meme research by formulating political meme understanding as a multi-level classification problem. The first level identifies whether a meme expresses support or trolling, while the second level identifies the corresponding political person or party. Thus, MMLow 2026 moves beyond binary troll classification toward fine-grained political stance and target identification, providing an opportunity to develop culturally aware multimodal models for code-mixed Tamil political discourse.

3 Methodology

We formulate the MMLow 2026 political meme classification shared task as a hierarchical multi-level classification problem, as shown in Figure 2.

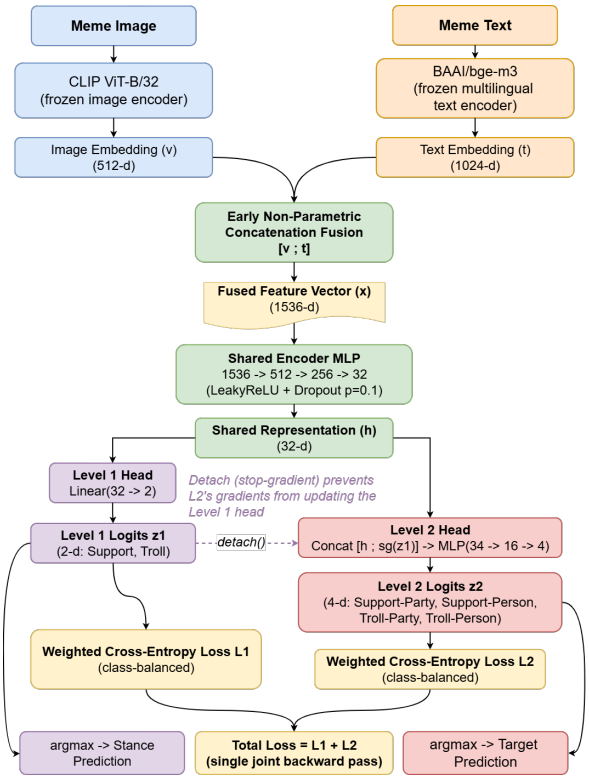


Figure 2: Flow diagram explaining the methodology of our hierarchical approach

Given a meme instance consists of an image I and its associated textual content in Tamil, T, our objective is to jointly predict:

- **Coarse-grained Level 1 label:** $y_1 \in \mathcal{Y}_1$, where $\mathcal{Y}_1 = \{\text{Support}, \text{Troll}\}$.
- **Fine-grained Level 2 label:** $y_2 \in \mathcal{Y}_2$, where $\mathcal{Y}_2 = \{\text{Support for Person}, \text{Support for Party}, \text{Troll against Person}, \text{Troll against Party}\}$.

where y_2 is semantically dependent on y_1. Level 2 prediction is explicitly conditioned on the Level 1 prediction, reflecting this label hierarchy structurally.

3.1 Dataset Description and Preprocessing

The dataset contained 969 labeled memes, where each data instance included a meme image along with its corresponding text present in the image. The labelled original dataset D (Rajiakodi et al., 2026) is partitioned into training and validation subsets, using a stratified 90/10 split on the Level

1 label. This strategy preserves class proportions across both subsets and mitigates the effect of the class imbalance (see Figure 4). Categorical labels at both levels are transformed to integer class indices using label encoding.

3.2 Multimodal feature extraction

To get semantically rich, modality-specific representations, we employ two frozen pretrained encoders:

- **Textual Encoder:** We experimented with two multilingual text encoders to represent textual information: BGE-M3 (Chen et al., 2024) and MuRIL (Khanuja et al., 2021). We found that BGE-M3 consistently outperformed the other model thus became our preferred text encoder, while MuRIL was used as a baseline for comparison. A detailed comparison of the two encoders is provided in Appendix Table 3. Specifically, the textual content t_i is encoded using the BGE-M3 (BAAI/bge-m3) multilingual encoder.
- **Visual Encoder:** Meme image v_i is encoded using a CLIP ViT-B/32 image encoder (Radford et al., 2021).

3.3 Multimodal feature fusion

Visual and textual embeddings are combined via early (Barnum et al., 2020), non-parametric concatenation fusion:

$$x_i = v_i \parallel t_i \tag{1}$$

where v_i is the input meme instance, t_i is the textual content present in the meme, and $\parallel$ is the concatenation operation producing a single fused feature vector x_i per instance.

3.4 Hierarchical Classification Architecture

We introduce a hierarchical classifier, composed of a shared trunk and two prediction heads with explicit inter-level conditioning. The complete model description can be seen in the Appendix Table 4

- **Shared encoder:** x_i was passed through a shared multilayer perceptron:

$$h_i = \mathrm{MLP}_{\mathrm{shared}}(x_i) \tag{2}$$

- **Level 1 head:** A linear classification layer maps the shared latent representations to Level 1 logits:

$$z_i^{(1)} = W_1 h_i + b_1 \tag{3}$$

- **Level 2 head (hierarchically conditioned):** It receives the concatenation of the shared latent representation and Level 1 logits, enabling Level 2 prediction to be conditioned on the model's Level 1 prediction:

$$z_i^{(2)} = \mathrm{MLP}_{L2}\left(h_i \parallel \mathrm{sg}\left(z_i^{(1)}\right)\right) \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient (detach) operator. By detaching $z_i^{(1)}$ before it enters the Level 2 head, we permit the Level 2 head to exploit the coarse-grained prediction as an informative conditioning signal without allowing Level 2's gradients to backpropagate into, and potentially corrupt, the Level 1 head's optimization. This technique ensures decoupling Level 1 head learns only from its own loss, while the Level 2 head learns from its own loss.

3.5 Class Imbalance Mitigation

To confirm the effectiveness of class weighting, we compared the results of using a weighted and an unweighted cross-entropy loss function. As shown in Table 1 the application of a weighted loss improved the average of the macro-F1 scores for both classification levels: from 0.7652 to 0.7738. Specifically, for Level-1, the macro-F1 score increased from 0.7710 to 0.7869. As for Level-2, the performance was comparable, with the weighted loss achieving a slightly higher macro-F1 score of 0.7608 compared to 0.7593 for the unweighted loss. Overall, the use of class weighting led to a better macro-F1 performance for both levels, which suggests that it can be beneficial for addressing the class imbalance problem and improving the minority class recognition.

Table 1: Comparison of weighted and unweighted loss functions for multimodal training using the BGE-M3 text encoder.

Split	Loss	Avg. Macro-F1	Std. Dev.
Level-1	Unweighted	0.7710	0.0198
Level-1	Weighted	**0.7869**	0.0508
Level-2	Unweighted	0.7593	0.0329
Level-2	Weighted	**0.7608**	0.0382
Average	Unweighted	0.7652	0.0264
Average	Weighted	**0.7738**	0.0445

3.6 Training Objective and Optimization

Our model is trained end-to-end via a joint multitask objective:

$$\mathcal{L} = \mathcal{L}_1\left(z_i^{(1)}, y_i^{(1)}\right) + \mathcal{L}_2\left(z_i^{(2)}, y_i^{(2)}\right) \tag{5}$$

computed and backpropagated in a single optimization step per batch, such that the shared encoder receives gradient signal from both levels simultaneously.

The model's learnable parameters are updated using AdamW optimizer with an initial learning rate of 1×10^{-3}. A ReduceLROnPlateau scheduler is employed, which monitors the average of Level 1 and Level 2 validation macro-F1 scores and slows down learning by reducing the learning rate by a factor of 0.5 after 2 epochs without improvement. The model is trained for up to 60 epochs with a batch size of 32. The checkpoint that achieves the highest average validation macro-F1 across both levels is retained for final evaluation and inference.

3.7 Evaluation

Model performance is assessed on a held-out validation split using:

- **Macro-averaged F1-score:** Computed independently for Level 1 and Level 2, serving as the primary, imbalance-robust evaluation metric and model selection criterion (see Figure 3).
- **Accuracy:** Reported as a secondary metric for both levels.

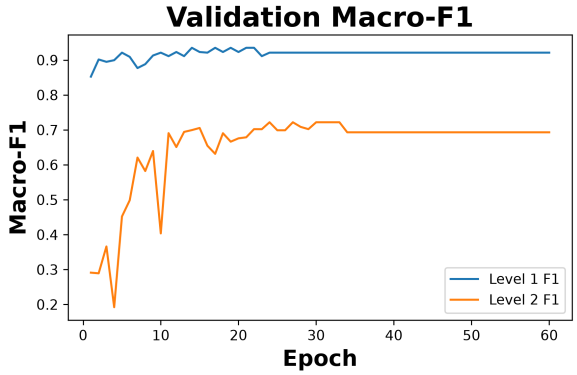


Figure 3: Macro-F1 score for both coarse-grained level 1 and fine-grained level 2 on each epoch

3.8 Inference

At test time, the identical feature-extraction and fusion pipeline is applied to unlabeled test instances. The best checkpointed model was used to produce Level 1 and Level 2 logits, from which the argmax class is taken at each level:

$$\hat{y}_1^{(i)} = \arg\max z_i^{(1)}, \qquad \hat{y}_2^{(i)} = \arg\max z_i^{(2)} \tag{6}$$

Predicted class indices are decoded back to their original categorical labels by using label encoders fitted on the training data, and final predictions are exported for submission.

4 Results

Table 2 presents the official evaluation results of our submitted system on the unlabeled test set. The system is evaluated using macro F1-score metrics described further in Appendix A.1.

Table 2: Evaluation results as per the organizers.

	Level 1	Level 2	Final (average)
Score	0.856	0.5321	0.694

5 Conclusion

Our results demonstrate that a hierarchical classifier built on frozen multimodal embeddings can effectively capture the coarse-to-fine label structure of the MMLow task. All reported metrics are averaged across five random seeds [7, 10, 42, 56, 100], providing a more robust estimate of model performance and reducing the impact of training variance. The strong Level 1 validation macro-F1 (0.898) and accuracy (0.918) confirm the comparative easiness to separate stance (*Support* vs. *Troll*) from fused CLIP + BGE-M3 features. Whereas the lower Level 2 validation macro-F1 (0.676) and accuracy (0.736) reflect increased difficulty of identifying the specific *target* of a meme.

Several limitations should be noted. First, the dataset is small (only 969 labeled memes). Second, by relying on only frozen pretrained encoders with simple concatenation fusion, we essentially miss out on any cross-modal interactions that could be learned by fine-tuning. Third, no named entity recognition is explicitly performed, which arguably could be crucial for target identification; perhaps adding an entity recognition/grounding module would be a useful direction for improving Level 2 performance in the future.

References

George Barnum, Shirin Talukder, and Yisong Yue. 2020. On the benefits of early fusion in multimodal representation learning. *arXiv preprint arXiv:2011.07191*.

Jia Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. 2022. hate-alert@dravidianlangtech-acl2022: Ensembling multi-modalities for tamil trollmeme classification. *arXiv preprint arXiv:2204.12587*.

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 Talukdar. 2021. Muril: Multilingual representations for indian languages. *Preprint*, arXiv:2103.10730.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Premjith B, Subalalitha CN, Rahul Ponnusamy, Anshid K A, Bhuvaneswari Sivagnanam, Jananayagan V, Ragavan N, Santhini P, and Bharathi Raja Chakravarthi. 2026. Overview of the shared task on multilevel political meme classification in Tamil and Malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 101–113, Underline (Virtual). Association for Computational Linguistics.

MD Musa Kalimullah Ratul, Sabik Aftahee, Tofayel Ahmmed Babu, Jawad Hossain, and Mohammed Moshiul Hoque. 2025. Cuet_networksociety@dravidianlangtech 2025: A multimodal framework to detect misogyny memes in dravidian languages. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 529–535, Albuquerque, New Mexico. Association for Computational Linguistics.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Shivali Suryawanshi, Bharathi Raja Chakravarthi, Pranav Verma, Mihael Arcan, John P. McCrae, and Paul Buitelaar. 2020. A dataset for troll classification of tamilmemes. In *Proceedings of the Second Workshop on Abusive Language Online*.

A Appendix

A.1 Evaluation Metric: Macro F1 Score

The macro F1 score (Sokolova and Lapalme, 2009) is a metric used to evaluate the performance of multi-class classification models. It is a harmonic mean of the F1 scores of all classes, where each class is weighted equally. It is suitable for imbalanced data sets and cases that require equal importance to be assigned to each class.

The calculation of this metric consists of two steps:

1. For each class, calculate an F1 score. The F1 score is the harmonic mean of precision and recall:
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

2. Take the arithmetic average of these per-class F1 scores:
$$\text{Macro F1} = \frac{1}{N} \sum_{i=1}^{N} F1_i \quad (8)$$

where N is the total number of classes, and $F1_i$ is the F1 score for the i-th class.

A.2 Comparison of Text Encoders

Table 3 compares the performance of BGE-M3 and MuRIL in the weighted multimodal scenario. BGE-M3 achieved higher average macro-F1 scores than MuRIL at both classification levels. At Level-1, BGE-M3 achieved a macro-F1 score of 0.7518 compared to 0.7158 for MuRIL, an improvement of 3.6 percentage points. At Level-2, BGE-M3 achieved 0.7346, while MuRIL achieved 0.7218, corresponding to an improvement of 1.3 percentage points. Although the outperformance margin reduces at the finer-grained Level-2 classification task, BGE-M3 remains superior as it facilitates better text representations for multimodal fusion.

Table 3: Comparison of weighted multimodal training using BGE-M3 and MuRIL text encoders

Split	Text Encoder	Avg. Macro-F1	Std. Dev.
Level-1	BGE-M3	**0.7869**	0.0508
Level-1	MuRIL	0.7272	0.0579
Level-2	BGE-M3	**0.7608**	0.0382
Level-2	MuRIL	0.7282	0.0317
Average (BGE-M3)		**0.7738**	0.0445
Average (MuRIL)		0.7277	0.0448

A.3 Complete model architecture details

Table 4 presents the complete architecture of the proposed hierarchical model.

A.4 Code Availability

The implementation of our proposed framework is publicly available at: https://github.com/vishalprasad1912/CPS_Lab_1_MMLow_2026

Table 4: Complete description of the model architecture

Component	Type	Configuration	Parameters
shared	Sequential	Sequential((0): Linear(in=1536, out=512, bias=True) (1): LeakyReLU(negative_slope=0.01) (2): Dropout(p=0.1, inplace=False) (3): Linear(in=512, out=256, bias=True) (4): LeakyReLU(negative_slope=0.01) (5): Dropout(p=0.1, inplace=False) (6): Linear(in=256, out=32, bias=True) (7): LeakyReLU(negative_slope=0.01)	926,496
l1_head	Sequential	Sequential((0): Linear(in=32, out=2, bias=True)	66
l2_head	Sequential	Sequential((0): Linear(in=34, out=16, bias=True) (1): ReLU() (2): Linear(in=16, out=4, bias=True)	628
Total trainable	–	–	**927,190**

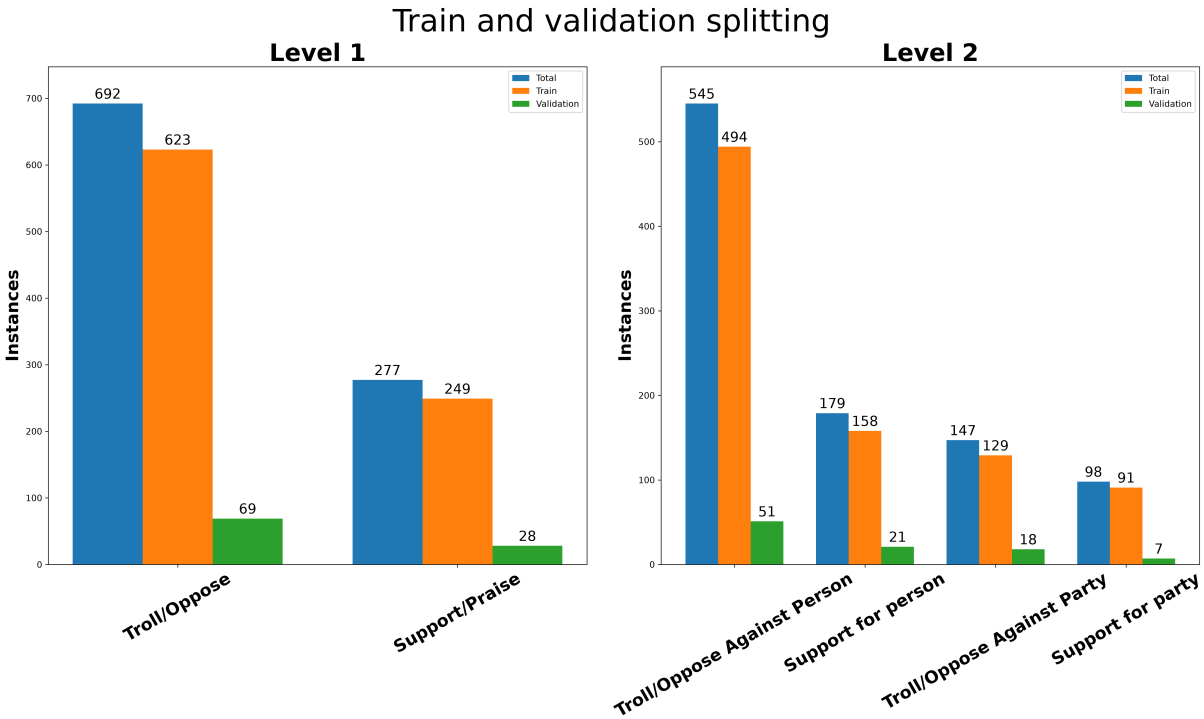


Figure 4: Distribution of the dataset for training and validation using the stratified sampling technique.