Learning from Annotator Disagreement for Enthymeme Detection and Reconstruction on Social Media

Moez Ur Rehman[1], Haiyan Alam[2]

[1, 2] *National University of Computing and Emerging Sciences (FAST-NUCES)*

Abstract

The paper discusses the approach followed by team ArgMine FN for the task on Missing Pieces and Misinformation shared task of MediaEval 2026. The Task 1 (Enthymeme Classification with binary or three class labels) was performed by fine-tuning BERTweet base under two supervised techniques: weighted cross entropy on majority-vote hard labels (Run 1) and KL-divergence with a mixed loss combined weighted cross-entropy against the full annotator distribution (Run 2). For Task 2 (generation of missing pieces) Llama-3.1 -8B-Instruct with QLoRA was fine-tuned, this was done for conditioning each generated piece on the class predicted labels from the corresponding Task 1 run. It was noted that on the Three label classification task, soft-label supervision increased macro-F1 (0.5087 vs 0.4865) and a relative gain of 47% in conclusion F1 (0.27 vs 0.18) was observed. No improved metrics on the binary task offered by the soft labels and cross entropy produced was higher against one-hot gold, this observation was attributed to evaluation metrics mismatch rather than a calibration failure. A rare conclusion class downgraded Task 2 qualitative results on those occurrences where we documented models missing a large chunk of relevant information. A negative failure resulted from a Flan T5-base generator that produced degenerated outputs and was abandoned in favor of Llama.

1 INTRODUCTION

An enthymeme is an argument where a proposition either a supporting premise/evidence or a final piece (conclusion) is left missing and an assumption to infer that from a shared context is established. Enthymemes are pervasive, impactful on social media and because of their length constraints authors are forced to make concise arguments and this causes them to lose logic [1]. An argument mining application consists of reconstruction of these missing components following their detection, fact-checking and classification of misinformation. The MediaEval 2026 Missing Pieces and Misinformation shared task is categorized into three sub-tasks: binary classification (Task 1A), three-class classification with premise / conclusion / none labels (Task 1B) and a text-generation of the missing implicit component (Task 2).

This paper discusses the ArgMine FN submission of all three sub-tasks. Our main methodological question concerns how the incorporation of annotators during training can be accomplished. The dataset provides up to five independent annotations per tweet classified into having premise/conclusion or none labels and inter annotator agreement is moderate (Fleiss $\kappa \approx 0.574$ three class, 0.629 binary), these are driven basically by a genuine ambiguity on the rare conclusion class. The standard approach followed is to take the majority-vote label while discarding the minority-vote as noise. We establish a comparison against an alternative route in which the full empirical distribution of the votes of the annotators is used as a soft training target via a KL-divergence penalty.

For Task 2 we used a QLoRA-fine-tuned Llama-3.1-8B-Instruct generator conditioned on the predicted Task 1 labels, in doing so only one reconstruction per non-none test instance was produced. Initially we used a Flan-T5-base model but observed unrealistic and degenerate outputs so we ended up switching to a more powerful model that is finely tuned on instructions.

The paper makes three experimental contributions. First on three label classification tasks, we observed a rise in macro-F1 from 0.4865 to 0.5087 when done on soft labels, these results are driven from a gain of relative 47% on the rare conclusion class. Second, on the binary label classification task soft-labels are analytically indistinguishable and this suggested that soft labels help on fine-grained label spaces. Third, the rare-class ceiling in Task 1 propagates into Task 2: even the better classifier recovers only two of six gold conclusion instances, capping reconstruction quality on those rows by construction.

2 RELATED WORK

An enthymeme detection on social media platforms was formulated by Flaccavento et al. [2], it is he whose dataset and annotations are the foundations of this MediaEval 2026 shared task. The detection converges to the shared tasks of argument mining and pragmatic inference, so this opens up for the possibility of broader work in the identification of implicit suggestions in argumentative discussions.

For shorter text classification tasks on social media, pre-trained transformers that are fine-tuned on tweet text are considered the most principal approach. BERTweet [3] which is a RoBERTa-base [4] architecture that is pre-trained on 850 million English tweets outshines a more generalized version- BERT [5] across a tweet domain NLP tasks. Therefore, we used

MediaEval'26: Multimedia Evaluation Workshop, June 15–16, 2026, Amsterdam, Netherlands and Online
khawajamoez07@outlook.com (M.Ur Rehman); haiyan.alamshykh@gmail.com (H. Alam)

BERTweet-base for Task 1 for this reason. A degraded validation performance metrics when attempting to use BERTweet-large was observed as the dataset size was too insufficient so to prevent over fitting when fine tuning on the larger model we decided to go with BERTweet-base.

A label-distribution learning [6,7] is done for training the model against the experimental spread of human evaluation and majority-vote labels are not put into account, this justifies that disagreement on hard cases is explanatory therefore is not treated as noise. On the higher-inner annotator calculation of variance has shown gains when this technique is utilized, but due to class imbalance and with hard-gold metrics of evaluation this still remains understudied.

For Task 2, we utilized an instruction tuned large language model via (LoRA) [8] and it's more subtle variant QLoRA [9] which has become a more of a spotlight approach to controlled text generation when under low compute budgets. For this task we integrated Llama-3.1-8B-Instruct with QLoRA after we observed that Flan-T5-base [10] was producing unstable and frequent illogical outputs.

3 APPROACH

3.1 Data

The released dataset (merged_annotations_v2.csv) contains tweets each independently annotated by five annotators with one of the three labels - premise, conclusion or none and a rebuilt implicit text of the tweet. The class distribution is heavily biased: approximately 66.2% none, 29.6% premise and 4.2% conclusion. Across the five annotator jury the Fleiss κ is approximately 0.574 on three-class label task and is recorded to be 0.629 on binary label classification task. A distortion between annotators is recorded when per-annotator analysis is carried out, as one of the annotators was labeling conclusions approximately 6.6x more frequently than another, this suggests more deviation rather than noise.

Training was done on 89/11 split having a seed = 42 that leaves us with 1,185 training tweets and 148 validation tweets. The (test_v2.csv) is the blind test set containing 148 additional tweets. On the experimental spread of annotator votes we compute soft-labels for every tweet, for example a tweet that was labelled as premise by three annotators and none by two of them the soft label recorded is [0.4, 0.6, 0.0]. The character length of tweet has mean of 172 and maximum length is evaluated is 280; so as per this we set the tokenizer max_length attribute of BERTweet to be 128 subword tokens, in order to cover more than 99% of the inputs.

3.2 Task 1: BERTweet Classification

BERTweet-base (125M parameters) with standard sequence classification having (768-dim) for its hidden state with a dropout (p=0.1) and is linearly projected to three class logits. We used AdamW with β_1 set to 0.9, β_2 set to 0.999 and with weight decay as 0.01, learning rate equals $2x10^{-5}$, linear warm-up over the first 10% of steps and the remainder followed by a linear decay, a batch size of 16 was used with FP16 mixed precision, gradient clipping settings were set to 1.0 and this ran on 10 epochs and with early stop on the macro score F1 of validation set.

Run 1 (hard labels): The loss function used is the weighted cross entropy on majority -vote labels. Class weights are calculated using the inverse of the class frequency in the training set, this gets us a conclusion-class weight of approximately 7.795. This discards minority annotator vote labels and treats majority vote labels as ground truth.

Run 2 (soft labels): The loss function we used here is the combination of weight cross-entropy on the majority vote labels and KL-divergence against the soft-label.

$$L_2 = \alpha \cdot CE\,(hard) + (1 - \alpha) \cdot KL\,(soft \mid model) \quad (1)$$

Initially α = 0.5 (equal weightage) we noted the conclusion class to collapse and the model converged to a state where it was unable to predict the conclusion class on the validation set. When we increased α = 0.7 we noticed a stable training pattern, the KL divergence term normalized the predictions toward the experimental annotator spread and then we observed the CE term to produce a more powerful gradient signal from the majority vote labels. So we used the value of α = 0.7 throughout.

3.3 Task 2: Llama Generation

For this task, we fine-tuned the Llama-3.1-8B-Instruct model using QLoRA. We used a 4-bit NF4 quantization of the base model weights and set the compute data type to bfloat16. The LoRA adapters were configured with a rank of 16 and a scaling factor of 32, applied to the multi-layered perceptron projection matrices like gate, up and gate down and the attention projections, q, k, v, o. During training, we used a learning rate of 2×10^{-4} with the AdamW optimizer in 8-bit mode. We trained the model for 3 epochs with a seed value of 42. The training was done on a Colab Pro with an Nvidia A100 GPU.

We acquired the training data by flattening five annotator format files that consisted of 2,298 (tweet, label, implicit_text) to one annotator per tweet and those tweets that were annotated as premise or having a conclusion label we rebuilt the tweet until it is fully reconstructed to make sense. This replicates a Llama-3.1's native chat template. Using a system prompt with a user turn tweet and a label question ("What is the missing premise?" or "What is the missing conclusion?") and then an assistant to reconstruct the annotated piece of tweet. The training and validation is done on the tweet level to prevent annotator specific leakage.

When inference is done each test was assigned to the predicted label that came from task 1 through (Run 1: hard label classifier, Run 2: soft-label classifier). Where the tweet was annotated as none we produced an empty generated implicit text field. We used temperature settings of 0.5 throughout after testing on T = (0.3, 0.5, 0.7) we concluded T= 0.5 produced the best results for this task. T =0.3 produced shorter outputs and often inverted the idea of the missing piece suggestion as the polarity of the proposition was inverted. While T =0.7 caused noisy outputs and broken syntax. So the T = 0.5 was observed to produce more fluent and high fidelity missing text generation outputs and this perfectly aligned with tasks evaluation metrics (RecEval Inter-step Correctness and ROSCOE-SS) while maintaining semantic similarity.

As previously highlighted about the negative result with Flan-T5, we fine-tuned google/flan-t5-base (FP-32) on the same task but due to abnormal outputs on the validation set and illogical generations, repetitive patterns and being irrelevant to the input tweet we discarded it for this task. So for a full precision and task of this argumentative nature and a dataset of limited size we can report this model to underperform. Therefore abandoning it in favor of Llama was the right choice for us.

4 RESULTS AND ANALYSIS

4.1 Task 1 Results

By the task organizers an evaluation was conducted on the 148 blind test set tweets. The Table 1 below summarizes the results for both these runs

Table 1: **Task 1 evaluation on the blind test set**

Task	Run	Macro-F1	Accuracy	Cross-entropy
1A (binary)	Run 1	0.6679	0.6824	0.7021
1A (binary)	Run 2	0.6673	0.6892	0.8391
1B (three-class)	Run 1	0.4865	0.6486	0.8691
1B (three-class)	Run 2	0.5087	0.6486	1.0071

Table 2: **Per-class F1 on Task 1B**

Class	Support	Run 1	Run 2	Δ
none	97	0.7444	0.7487	+0.004
premise	45	0.5333	0.5106	−0.023
conclusion	6	0.1818	0.2667	+0.085

Here two observations are most prominent. First the macro F1 score of task 1B is improved by 0.022 by the soft-label. This is previously discussed as the relative gain of 47% is observed on the rare conclusion class. But this comes up with a small trade-off that is a small loss on the premise class so this suggests KL divergence term affects probability and causes a shift in spread from the more prominent class to a class that is under-represented.

Second, on the binary task the (Δ macro-F1 = −0.0006) which suggest the similarity and when the label space collapses and merges both labels for an enthymeme as being a premise and having a conclusion, the class that is highly disagreed upon by the annotators no longer exists as a separate target therefore soft-labels offer no advantage over information. So this result we analyzed through our study is consistent with that soft labels only help when (a) the label space is fine grained (b) underrepresentation of one or more class (es). And the (c) disagreement of the annotator on these classes is non-trivial.

4.2 The Cross-Entropy Observation

In both Task 1A and 1B, Run 1 logs a lower cross-entropy than Run 2, by about 0.14 nats in every instance. At first look, this seems to conflict with the macro-F1 outcome on Task 1B: Run 2 is a superior classifier yet an inferior predictor based on the metric most closely associated with probability calibration.

We believe this is due to a misalignment between Run 2's training goal and the assessment protocol. Run2 is designed to align with empirical annotator distributions, which in challenging instances are spread out (e.g., [0.5, [0.3, 0.2]). Cross-entropy is calculated using one-hot true majority labels. A scattered yet the output from correct-on-argmax Run 2 results in a loss of around -log(0.5) ≈ 0.69 based on its prediction class while a more accurate Run 1 forecast of [0.85, 0.10, 0.05] for the same

item yields only -log(0.85) is approximately equal to 0.16. The metric consistently incentivizes over-confidence, even when the scattered results more accurately represent the annotator distribution. We elaborate on this more in our Quest For Insight (QFI) accompanying document.

4.3 Cascading Errors into Task 2

ROSCOE-SS (Facebook/roscoe-512-roberta-base), was used by the organizers for the evaluation of Task 2 generations. The cosine similarity is calculated for the missing text suggestion against what annotators reconstruct, then it is rescaled and a mean is calculated. Two aggregates are evaluated, where the generation exists with a gold reference, the matched-only score is calculated as an average value over these tweets. If a generation without a reference exists or a reference exists without a generation the zero score assigns zero. Table-3 illustrates both

Table 3: Task 2 ROSCOE-SS on the blind test set

Run	Matched-only	With-Zeros
Run 1	0.9004 (n=34)	0.3779 (n=81)
Run 2	0.9042 (n=30)	0.3478 (n=78)

As from the table it can be seen that matched-only scores are high and nearly similar across both runs, this indicates that when an enthymeme is detected by the classifier the generator rebuilds the missing pieces of suggestion with high confidence. The 'With Zeros' scores are far lower as they fold in coverage failures. The two aggregates difference evaluated is almost a pure measure of cascading error, therefore we could see that generator here is not the bottleneck but this issue is caused by the upstream classifier.

Run 1 and Run 2 have almost the same quality when it comes to matched-only results, but Run 1 performs better in the zeros metric, which depends on coverage. Run 1 is better at predicting overall enthymemes and recovers about 68.6% of the gold positive instances, which is 35 out of 51, compared to Run 2's 60.8%, or 31 out of 51. Run 2's predictions are more conservative, while Run 1's are focused more on premises. The gold positive instances are mostly labeled as premises (45 out of 51), so Run 1 provides better coverage. The high recall of the conclusion class in Task 1 for Run 2 doesn't have much impact because there are very few conclusion occurrences, so they don't affect the overall results much.

When Run 1 errors are manually inspected a connection is established with Task 2 that is heavily linked to annotators' consensus. All 16 false negatives are almost all premise (15/16). These are tweets that are less likely to be treated as implicit by the annotators or they are conceived as direct argument. The 30 false positives are positives model treats as arguments, while the majority of the annotators have labelled them as none. They may contain intended questions or accusations that could surface an argument and the annotators' consensus is hidden on these missing components. The most challenging instances for the model align with the situations of lowest annotator consensus, reflecting the same characteristic that drives the soft-label method in Task 1.

Our matched only score of 0.90 suggests notably strong Llama generator reconstructions to match human level annotation quality on the instances. The generator component of our task 2 pipeline is not the bottleneck. The ceiling on Task 2 performance is set by the Task 1 recall and the model used by us in Task 1 i.e: BERTweet-base classifier is the real culprit. BERTweet-base trained on 125M parameters lacks natural tone of conversation and this limits its natural reasoning capacity in identification of disguised argumentative structure that a tweet may contain that may also depend on sarcasm sometimes and also pure and simple questioning, or the tweet may receive and interpretative framing, this phenomenon was also observed via patterns where the false positive and false negatives were reported. Therefore, improving the upstream classification task whether by using a more advanced BERTweet-large model or using an end to end jointly trained pipeline remains the highest-leverage direction for future work on this task.

5 CONCLUSIONS

We have described our submission to the MediaEval 2026 Missing Pieces and Misinformation shared task and reported results across all three sub-tasks. The central finding is when using normalization KL-divergence while modelling across the full spread of the annotators helped improve three -class classification macro-F1and also conclusion class F1 in particular. The benefit is noted when the label space is fine-grained and when the minority class is under-represented. An anomaly is also seen in cross entropy loss function when the trained does not follow one hot gold labels and is more faithful to experimental annotator spread and we present a cascading recall limit that limits the quality of Task 2 generation on the uncommon conclusion class.

Future research could explore three avenues: combining hard and soft regimes to regain the strengths of both; force-producing across all evaluation cases under a label-neutral prompt to separate Task 2 quality from Task 1 recall and investigating data-augmentation techniques for the uncommon result category. Apart from the leaderboard, the results indicate that assessment protocols for Label Distribution Learning (LDL) techniques ought to contrast with annotator distributions instead of majority-vote argmax, a conceptual aspect we elaborate on thoroughly in our accompanying Quest for Insight paper.

ACKNOWLEDGMENTS

We want to express our appreciation and thanks to the task organizers for managing the shared task smoothly and for offering their support, assistance and encouragement whenever we contacted them.

Declaration on Generative AI

While working on this paper, the authors used Grammarly to check for grammar and spelling mistakes. Once they were done with the tools, they went through the text again, made any necessary changes and were fully responsible for the content that was published.

REFERENCES

[1] Douglas Walton. Informal Logic: *A Pragmatic Approach* (2nd ed.). Cambridge University Press, 2008.

[2] Alessandro M. Flaccavento, Youen Peskine, Paolo Papotti, Riccardo Torlone and Raphaël Troncy. Automated detection of tropes in short texts. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025)*, pages 1420–1432, 2025.

[3] Dat Quoc Nguyen, Thanh Vu and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*, 9–14, 2020.

[4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[5] 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 (NAACL-HLT 2019)*, 4171–4186, 2019.

[6] Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (July 2016), 1734–1748, 2016.

[7] Joshua C. Peterson, Thomas L. Griffiths and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019)*, pages 9617–9626, 2019.

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proceedings of the 2022 International Conference on Learning Representations (ICLR 2022)*, article no. 32, 2022.

[9] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman and Luke Zettlemoyer. QLoRA: Efficient fine-tuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*, volume 36, pages 36211–36224, 2023.

[10] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. arXiv preprint *arXiv:2210.11416*, 2022.

[11] Martial Pastor and Nelleke Oostdijk. A Resource for Enthymeme Detection in Controversial Political Discourse. *arXiv preprint arXiv:XXXX.XXXXX*, 2026