A Lightweight Contrastive System for Misinformation Detection in Social Media Tweets

Priyam Saha[1]

[1]Associate in Cyber Risk Advisory, Grant Thornton Advisors LLC, India

Abstract

A compact classification system was developed and submitted to Prompt RecOvery For MisInformation Detection (PROMID) Subtask 3 for the detection of misinformation in tweets about the Russia–Ukraine conflict on Twitter platform as provided by the workshop organisers. The proposed solution combines a frozen RoBERTa encoder, a small projection head trained with a supervised contrastive objective, and a lightweight classifier trained jointly with binary cross-entropy. Design choices were driven by compute and memory constraints; several practical implementation details and evaluation outcomes are reported to support reproducibility of results. The submission of predictions computed on the test dataset as provided by the organizers was made on the Codabench platform as team 'priyam_saha17' and submission id as 431064. On the official test set, the methodology produced a weighted F1 score of 0.82 (precision 0.87, recall 0.80), thereby securing the 5th rank in the track leaderboard, accessible at Link. For a comparison, the leaderboard was topped by team 'ClimateSense' who achieved a weighted F1 score of 0.91 (precision 0.91, recall 0.91). The approach, training pipeline and error analysis are documented in order to assist future participants and applied researchers working under limited resource conditions.

Keywords

misinformation detection, contrastive learning, RoBERTa, social media, PROMID

1. Introduction

Misinformation on social media has been recognized as a substantial challenge for public discourse and policy. Automated detection systems were requested in PROMID Subtask 3 to classify tweets related to the Russia–Ukraine conflict as *misinformation* or *non-misinformation*. This work documents a memory-efficient pipeline that was designed to operate on a single 16GB Tesla P100 GPU by freezing the transformer encoder and training compact head modules. The central design objective was to maximize representational separation between labeled classes using supervised contrastive learning while keeping the number of trainable parameters low on account of constrained compute resources.

2. Related Work

Contrastive representation learning has been widely adopted for visual and textual tasks due to its effectiveness at structuring embedding spaces. Classical methods for self-supervised contrastive learning were popularized by Chen et al. [1], which demonstrated the power of data augmentations and large-batch contrastive losses. Supervised variants that exploit label information were later proposed by Khosla et al. [2], showing improved downstream classification performance when positive pairs are formed from examples with the same label.

Pretrained language encoders such as RoBERTa [3] and BERT [4] have been extensively used for classification tasks; their contextualized representations are commonly fine-tuned end-to-end for high performance. Under compute constraints, however, head-only fine-tuning (freezing the encoder) is a pragmatic alternative and has been used in applied settings to balance cost and accuracy as put forward by Zhang et al. [8]. Recent work has also shown that combining contrastive objectives with supervised classification can increase robustness and separation in learned spaces [5].

Forum for Information Retrieval Evaluation, December 17-20, 2025, India

*Corresponding author.

impriyamsaha@gmail.com (P. Saha)

0009-0002-1167-3529 (P. Saha)

Table 1
Dataset statistics used in the experiments.

Split	# samples	Notes
Misinfo (original)	364	positives extracted from misinfo_train.csv
Non-misinfo (original)	34,174	full negative pool before downsampling
Combined (balanced)	728	positives + downsampled negatives
Train (80%)	582	stratified split
Validation (20%)	146	stratified split
Test (held-out)	2,414	final predictions were produced for these samples

The Prompt Recovery for Misinformation Detection (PROMID) shared task has been introduced to systematically study misinformation detection under prompt recovery and generalization settings [9]. The task statement, subtasks, datasets and evaluation metrics are described in detail by the organisers, providing a unified benchmark for multilingual and topic-focused misinformation detection in social media [10].

The PROMID Subtask 3 dataset collection was informed by a link-based annotation framework, namely, AMUSED proposed by Shahi et al. [6] and the subtask dataset has been shared by the organisers [7].

3. Dataset and preprocessing

The PROMID Subtask 3 dataset was provided by the organisers and consisted of manually annotated tweets collected during the first year of the Russia–Ukraine war. Two labeled CSV files were supplied: `misinfo_train.csv` (positive class) and `nonmisinfo_train.csv` (negative class). A held-out test CSV without labels was provided for final predictions.

A compact summary of data used in experiments is shown in Table 1. The negative class was heavily over-represented in the original collection and was downsampled to form a balanced training set so that contrastive positives and negatives were both balanced at a count of 364 during mini-batch training. Empty or very short text entries were removed. Tokenization was performed using the RoBERTa tokenizer with truncation to a maximum length of 512 tokens.

4. Methodology

The pipeline was intentionally simple and reproducible. The core modules are:

1. **Encoder.** A pretrained `roberta-base` model was used to extract contextual token embeddings. The encoder parameters were frozen during head-only training to reduce memory consumption and runtime.
2. **Pooling.** Mean pooling across token embeddings (masked by the attention mask) was used to obtain a single vector representation per tweet from the token level embeddings obtained from encoder.
3. **Projection head.** A small feedforward projection head (Dense → Dropout → Dense → LayerNorm) was trained with stochastic dropout active during training so that calling the projection head twice produced two stochastic views of the same example.
4. **Classifier head.** A compact classifier (Dense(256, gelu) → Dropout → Dense(1, sigmoid)) was trained jointly to produce final binary predictions.

The training objective combined a supervised contrastive loss and a binary cross-entropy loss so that the representation space was encouraged to bring same-label examples closer while the classifier learned decision boundaries on the pooled vectors.

4.1. Contrastive learning: conceptual and mathematical description

Contrastive learning aims to structure the representation space such that similar (positive) pairs are close while dissimilar (negative) pairs are separated. In supervised contrastive learning, the class labels are used to generate positive pairs for each anchor.

Given a minibatch of N examples, two stochastic views of each example were produced through dropout in the projection head, resulting in $2N$ projections $\{\mathbf{z}_i\}_{i=1}^{2N}$ (each $\mathbf{z}_i$ is ℓ_2-normalized). Let y_i denote the integer label for the example corresponding to projection $\mathbf{z}_i$; labels are duplicated to match the $2N$ projections.

The supervised contrastive loss used in this work is defined per anchor i as:

$$\ell_i = -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{z}_i^\top \mathbf{z}_p / \tau)}{\sum_{a \neq i} \exp(\mathbf{z}_i^\top \mathbf{z}_a / \tau)} \tag{1}$$

where $P(i) = \{p : y_p = y_i, p \neq i\}$ is the set of positive indices for anchor i, and $\tau > 0$ is a temperature hyperparameter. The final contrastive loss is averaged across anchors:

$$L_{\text{contrastive}} = \frac{1}{2N} \sum_{i=1}^{2N} \ell_i. \tag{2}$$

The supervised contrastive formulation encourages clusters corresponding to the same label while using all other examples in the batch as implicit negatives, improving utilization of batch information compared to pairwise binary losses. For reference, early self-supervised instantiations such as SimCLR [1] used two augmented views of the same instance and an InfoNCE loss; the supervised extension is discussed extensively in [2].

4.2. Combined loss and optimization

The classifier produced a scalar probability $\hat{y} \in (0, 1)$. The binary cross-entropy loss was computed in the usual way:

$$L_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right]. \tag{3}$$

The final training objective was a weighted sum:

$$L = \alpha L_{\text{contrastive}} + \beta L_{\text{BCE}},$$

with $\alpha = \beta = 1.0$ selected after light tuning. Gradient updates were applied only to the projection and classifier head parameters (unless an experimental unfreeze of top encoder layers was explicitly activated).

The overall system architecture is summarised in the flowchart shown in Figure 1, which illustrates the preprocessing steps, the frozen encoder, the contrastive projection pathway, and the classifier used to produce final predictions.

5. Implementation details

The system was implemented in TensorFlow 2.x using HuggingFace models. Key configuration choices were:

- **Model:** `roberta-base` (encoder), projection dim = 64.
- **Input:** Tokenization via RoBERTa tokenizer, max length = 512, truncation enabled.
- **Batching:** batch size = 16, training with shuffling and prefetch for tf.data pipelines.
- **Optimizer:** Adam with initial learning rate 3×10^{-4} on head parameters.

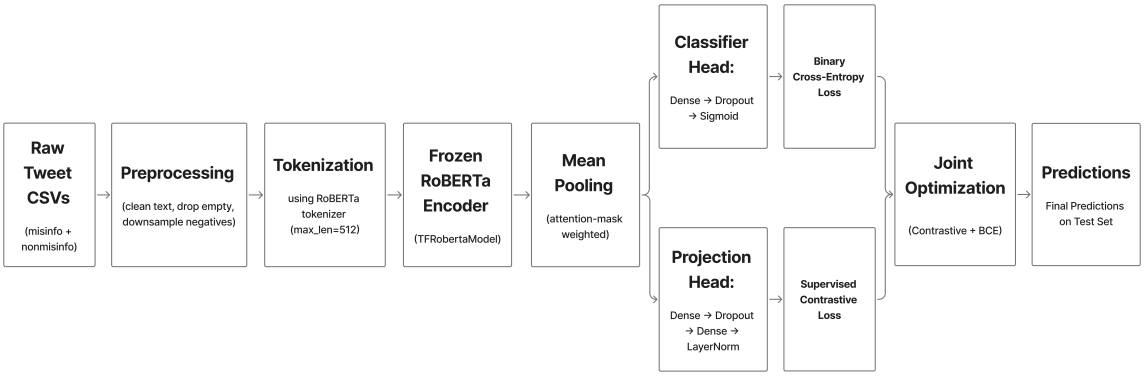


Figure 1: Flowchart illustrating the complete pipeline of the proposed lightweight contrastive system.

- **Regularization:** Dropout in heads and gradient clipping (global norm = 1.0).
- **Training policy:** Early stopping after 6 epochs with no validation F1 improvement; LR reduced by factor 0.5 after 3 non-improving epochs.
- **Compute:** Encoder parameters were frozen to limit memory; A standard Kaggle notebook was utilized for fine-tuning the model, leveraging a Tesla P100 GPU with 16 GB of memory, a maximum of 29 GB of RAM, and up to 57.6 GB of disk space.

During training, the projection head was invoked twice per batch with training=True (dropout active) to generate the two stochastic views without performing two encoder forward passes. This choice was made to minimize memory and computation while still obtaining the stochasticity required for contrastive training.

6. Experiments and results

6.1. Validation dynamics

Training was monitored with average losses (contrastive and classifier) and validation precision/recall/F1. The contrastive loss decreased rapidly in early epochs as the projection head learned to structure the embedding space; classifier loss dominated later epochs indicating head-level weight adjustments for the classification boundary. Learning rate reductions and early stopping were used to prevent overfitting on the small balanced training set.

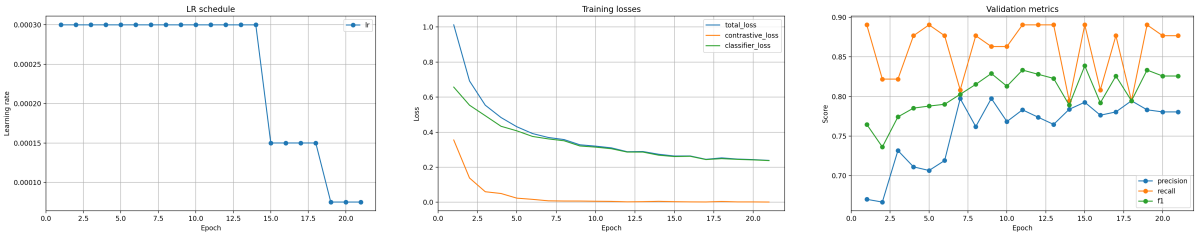
6.2. Final evaluation

Final predictions were produced on the provided test CSV and scored by the organizers' program. The official classification report returned the following per-class and aggregated metrics:

- **nonmisinfo:** precision = 0.96, recall = 0.80, F1 = 0.87, support = 2002.
- **misinfo:** precision = 0.4567, recall = 0.8285, F1 = 0.59, support = 414.
- **weighted average F1:** 0.8213.

The submission was made by the name of `priyam_saha17` (ID 431064) on the PROMID leaderboard for the hackathon hosted by PROMID Subtask 3 organisers on CodaBench.

The three panels in Figure 2 summarize the optimization trajectory. The learning rate remains flat until the scheduler triggers two stages of reductions. Contrastive loss converges rapidly as the projection space stabilises, while classifier loss declines more gradually as the boundary is refined. The validation curves reflect the model's high recall behavior throughout training and improving F1 until early stopping.



(a) Learning rate schedule (b) Training losses (c) Validation metrics

Figure 2: Training dynamics of the lightweight contrastive system. Panel (a) shows scheduled learning–rate reductions. Panel (b) shows rapid decrease in contrastive loss and slower BCE convergence. Panel (c) shows stable recall with more variable precision, resulting in steady F1 improvement.

6.3. Ablation study

An ablation study was conducted to evaluate the contribution of the supervised contrastive objective and the binary cross-entropy classifier loss to the final performance based on validation split off the training dataset. The full model jointly optimizes both objectives with equal weights ($\alpha = \beta = 1.0$). Two reduced variants were evaluated: (i) a classifier-only configuration where the contrastive term was removed ($\alpha = 0$), and (ii) a contrastive-only configuration where the classifier loss was removed ($\beta = 0$) in Table 2.

The results indicate that the joint optimization of contrastive representation learning and supervised classification is necessary to achieve a balanced precision–recall trade-off and best F1 scores under constrained compute settings. Further, it is observed that weights of $\alpha = \beta = 1.0$ yields a higher F1 score on validation data rather than an averaged approach ($\alpha = \beta = 0.5$). Hence, the configuration $\alpha = \beta = 1.0$ was adopted for prediction on test data.

Table 2
Ablation study showing the impact of individual loss components. α denotes the weight of the contrastive loss and β the weight of the classifier (binary cross-entropy) loss.

Configuration	α	β	Validation F1	Validation Recall	Validation Precision
Classifier only	0.0	1.0	0.6667	1.0000	0.5000
Contrastive only	1.0	0.0	0.0941	0.0548	0.3333
Contrastive + Classifier	0.5	0.5	0.7973	0.8082	0.7867
Contrastive + Classifier	1.0	1.0	0.8387	0.7927	0.8904

7. Error analysis and discussion

The system exhibited a high recall for the misinformation class but a comparatively low precision, indicating a tendency to over-predict the positive class. Manual inspection of false positives revealed common patterns:

- Tweets that quoted or criticised a claim were sometimes classified as endorsing it because the local text included keywords associated with misinformation; the model lacked explicit modeling of quotation or negation scope.
- Very short tweets, or tweets consisting primarily of a URL or an image reference, were often misclassified due to missing contextual signals.

Some practical remediation strategies are suggested as follows for future considerations:

- Enriching tweet inputs with surrounding context (linked article title or claim summary) when available.

- If resources permit, unfreezing the encoder or at least top transformer layers for a small number of epochs to allow encoder to learn features corresponding to this domain.

8. Limitations

The principal limitations are the reliance on a relatively small balanced training set obtained by downsampling and the use of a frozen encoder which limits representation adaptation. The resulting trade-off was computational feasibility vs. maximum attainable performance. The reported system therefore should be interpreted as a strong baseline for low-resource scenarios rather than a final state-of-the-art submission.

9. Conclusion

A lightweight supervised contrastive plus classifier system was described and evaluated for PROMID Task 3. The system produced a weighted F1 of 0.8213 on the provided test data. The design choices prioritized memory efficiency and reproducibility: encoder freezing, stochastic projection views via dropout, and a joint contrastive/BCE objective. The results indicate that contrastive separation helps achieve high recall for the misinformation class under constrained resources, and that further improvements are likely if complete or selective unfreezing is introduced.

Acknowledgments

The PROMID organizing committee is gratefully acknowledged for the dataset and the scoring infrastructure. The AMUSED framework and the dataset references provided by the organizers were used as background during the model pipeline design.

Declaration on Generative AI

During the preparation of this work, the author used GPT-5.1 for organization and better polishing of the phrases and language used in the article and Grammarly for grammar and spelling check. After using these services, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] T. Chen, S. Kornblith, M. Norouzi, G. Hinton: A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning (ICML)*, 2020.

[2] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, R. Liu, C. Wu: Supervised Contrastive Learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov: RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*, 2019.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding: In *NAACL-HLT*, 2019.

[5] B. Gunel, T. Afouras, A. Baş, M. C. Miller, A. Zisserman: Supervised Contrastive Learning for Limited Labels. *ICLR Workshop*, 2020.

[6] G. K. Shahi, T. A. Majchrzak: AMUSED: An Annotation Framework of Multi-modal Social Media Data. Technical report / preprint, 2022.

[7] G. K. Shahi, Y. Mejova: Too Little, Too Late: Moderation of Misinformation around the Russo-Ukrainian Conflict. *Websci '25*, 2025. DOI:10.1145/3717867.3717876.

[8] X. Zhang, A. Smith: Head-Only Fine-Tuning: A Practical Approach for Low-Resource Adaptation. *Workshop Report*, 2021. (Discussion of head-only strategies.)

[9] G. K. Shahi, A. Hegde, S. Satapara, P. Mehta, S. Modha, D. Ganguly, D. Nandini, H. L. Shasirekha, A. K. Jaiswal, G. Pasi, T. Mandl: Overview of the First Shared Task on Prompt Recovery for Misinformation Detection (PROMID 2025). *Working Notes of FIRE 2025, CEUR Workshop Proceedings*, 2025.

[10] A. Hegde, G. K. Shahi, S. Satapara, P. Mehta, S. Modha, D. Ganguly, D. Nandini, H. L. Shasirekha, A. K. Jaiswal, G. Pasi, T. Mandl: Prompt Recovery for Misinformation Detection at FIRE 2025. *Proceedings of the Forum for Information Retrieval Evaluation (FIRE)*, Association for Computing Machinery, 2025.

A. Reproducibility checklist

- Code: training script uses TensorFlow 2.x and HuggingFace TFRobertaModel.
- Model: roberta-base, projection dim = 64, classifier head as described.
- Hyperparameters: MAX_LEN=512, BATCH_SIZE=16, LR=3e-4, gradient clipping norm=1.0.
- Loss weights: $\alpha = \beta = 1.0$.
- Training: early stopping patience = 6, LR reduce factor = 0.5 after 3 non-improving epochs.

B. Code and notebook

The training notebook script, prediction CSVs, plots used in the article and the public Kaggle notebook carrying all fine-tuned model artifacts can be found over GitHub[1].

[1] https://github.com/priyam-saha-17/A-Lightweight-Contrastive-System-for-Misinformation-Detection-in-Social-Media-Tweets