Transformer-Based Multi-Task Text Classification of Cryptocurrency Social Media Text with DeBERTaV3-Small

V. T. Rushi Kannan[1,*], Sara Renjit[2]

[1]Koneru Lakshmaiah University, Aziz Nagar, Telangana, India
[2]Indian Institute of Information Technology Kottayam, Kerala, India

Abstract

We present a transformer-based multi-task framework for cryptocurrency discussions: hierarchical subjectivity classification across YouTube, Twitter, and Reddit, and Reddit query–answer relevance prediction. Built on `microsoft/deberta-v3-small` and enhanced with Focal Loss, Dice Loss, Label Smoothing, and Supervised Contrastive Learning, it addresses class imbalance and improves representation learning. In the Forum for Information Retrieval Evaluation (FIRE 2025) shared task on Opinion Extraction and Question Answering from Cryptocurrency-Related Tweets and Reddit Posts, our system achieved 83.83% validation accuracy and 78.32% Macro-F1 in Level 1, with consistent relevance prediction Macro-F1 near 70%. Official evaluation confirmed that our system ranked top on the hidden test sets, attaining a Macro-F1 of 1.0 and showing robust generalization. This underscores the effectiveness of hierarchical modeling, fold-based ensembling, and advanced loss functions for opinion mining and relevance detection in noisy cryptocurrency discussions.

Keywords

Transformers, Subjectivity Analysis, Relevance Prediction, DeBERTaV3-small, Supervised Contrastive Learning, Focal Loss, Dice Loss, Label Smoothing

1. Introduction

User-generated content on platforms such as Reddit, Twitter, and YouTube has opened new opportunities for NLP, while also introducing significant challenges [1, 2]. In the context of cryptocurrency, online discussions are often characterized by informal language, domain-specific jargon, sarcasm, and rapidly evolving terminology [3, 4]. These properties make automated analysis particularly valuable for understanding sentiment, market behavior, and the spread of misinformation [5, 6].

This study was carried out as part of the Forum for Information Retrieval Evaluation (FIRE 2025) CryptoQA shared task, *"Opinion Extraction and Question Answering from Cryptocurrency-Related Tweets and Reddit Posts"* [7], which provides the dataset, task definitions, and evaluation framework.

To address the challenges of noisy and imbalanced data, we propose a modular framework based on DeBERTaV3-small [8, 9]. Our approach integrates Dice Loss [10], Label Smoothing [11], Focal Loss [12], and supervised contrastive learning [13, 14], supported by stratified cross-validation to reduce overfitting. Experiments demonstrate competitive, and in some cases state-of-the-art, performance, highlighting the effectiveness of hierarchical modeling and domain-aware fine-tuning in this high-variance setting.

2. Related Work

Transformer-based architectures have become the foundation of modern NLP, largely replacing recurrent and convolutional models through the use of self-attention for efficient sequence modeling [2]. Building on this paradigm, pretrained language models such as BERT [1] and T5 [15] have achieved strong generalization across a wide range of NLP tasks. Later advances like DeBERTa [9] and its improved variant DeBERTaV3 [8] introduced disentangled attention and embedding sharing, which further

Forum for Information Retrieval Evaluation, December 17-20, 2025, Varanasi, India
*Corresponding author.
rushikannan@gmail.com (V. T. Rushi Kannan); sararenjit@iiitkottayam.ac.in (Sara Renjit)

enhanced contextual representations. Given this balance between accuracy and efficiency, we selected `deberta-v3-small` as the backbone encoder in our framework.

For hierarchical subjectivity classification, prior work has shown that cascaded architectures can improve decision-making by progressively narrowing the classification space [3]. A key challenge in these multi-stage pipelines, however, is class imbalance, which often biases models toward majority classes. To mitigate this, researchers have explored specialized loss functions: Focal Loss to emphasize hard-to-classify examples [12], Dice Loss to directly optimize overlap-based metrics [10], and Label Smoothing to improve calibration and generalization [11]. In parallel, contrastive learning has emerged as an effective technique for enhancing representation quality. Originally popular in computer vision [13], contrastive methods have since been adapted to NLP through models like SimCSE [16] and extended to supervised settings for fine-tuning large language models [14, 16].

For query–comment relevance prediction, earlier work in semantic similarity and information retrieval has demonstrated that transformer encoders can effectively capture contextual alignment between text pairs. In addition, optimization strategies such as threshold tuning for macro-F1 have proven useful for handling skewed class distributions.

Cryptocurrency-related discourse introduces further complexity due to its evolving jargon, sarcastic tone, and frequent misinformation [4, 17]. To address these challenges, specialized datasets like CryptOpiQA [18] have enabled systematic evaluation of sentiment, intent, and relevance modeling in this domain. Our work builds on these foundations by combining hierarchical modeling, domain-adapted transformers, advanced loss functions, and contrastive learning into a unified framework for opinion mining and relevance detection in cryptocurrency-focused social media.

3. Proposed Methodology

Our approach is intentionally flexible and domain-agnostic, allowing it to be applied to a wide range of multi-level text classification problems. It integrates a compact transformer backbone with a sequential (hierarchical) decision process, complementary loss terms for handling imbalance and improving representation quality, and a robust evaluation protocol. The complete architecture is presented in Figure 1.

3.1. Overall Framework

The following subsections (3.2–3.5) expand on these components in detail, grouped into preprocessing, model design, training strategy, and training setup. The system operates as a modular pipeline that converts raw text into structured predictions through the stages presented in Figure 1:

1. **Preprocessing:** Normalize and tokenize raw text to form model-ready inputs.
2. **Transformer encoding:** Extract contextual embeddings using a pretrained DeBERTaV3-small encoder [9, 8].
3. **Task-specific heads:** Attach lightweight classification heads for each level of the hierarchy.
4. **Loss computation:** Combine focal-like calibration and overlap-aware objectives with supervised contrastive fine-tuning [12, 10, 13, 14].
5. **Evaluation:** Assess performance using complementary quantitative metrics and qualitative embedding inspection [19, 20].

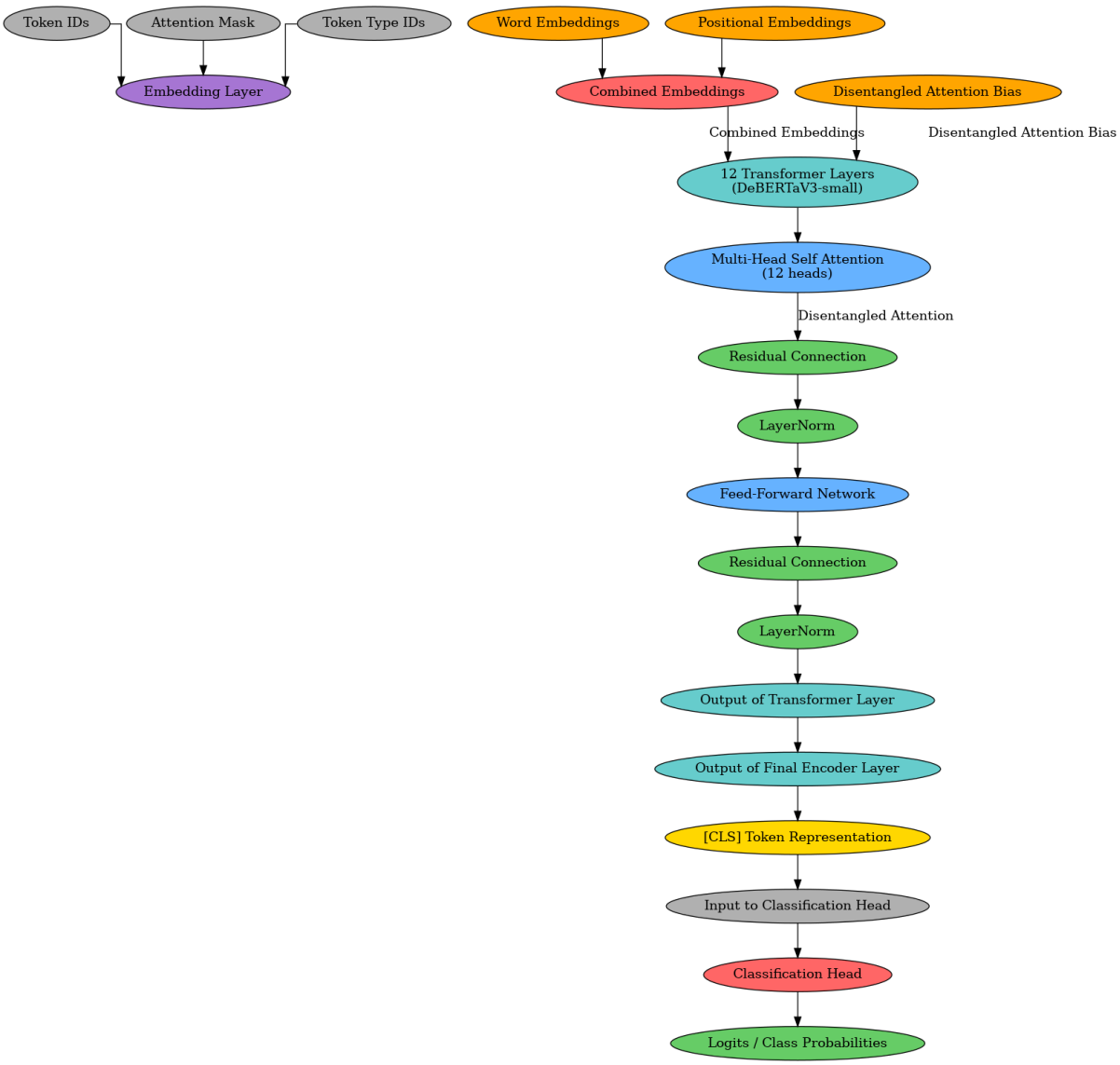


Figure 1: Overall architecture of the Transformer-based DeBERTaV3-small model.

In operation, the pipeline first performs minimal preprocessing, including basic textual cleaning and tokenization. Token-level inputs are fed into the DeBERTaV3-small transformer encoder to obtain contextualized sequence representations. The pooled output (from the `[CLS]` token) is passed to a task-specific classification head, which produces the final prediction. Figure 1 illustrates the complete pipeline and a close-up of a single encoder block.

3.2. Preprocessing pipeline

Preprocessing is intentionally kept minimal to preserve the original structure and meaning of the text. The same procedure is applied to all data sources (Reddit, Twitter, YouTube):

- Merge datasets into a single unified corpus.
- Standardize column names across sources.
- Concatenate relevant text fields with platform-specific tags.
- Retain HTML tokens where present.
- Adjust sequence lengths to a fixed maximum for efficient batching.

No additional cleaning, stemming, or data augmentation is applied.

3.3. Model design

3.3.1. Transformer backbone

The core of the model is the `microsoft/deberta-v3-small` encoder, chosen for its balance between performance and computational efficiency.

3.3.2. Classification heads

A lightweight linear layer projects the pooled sequence embedding into the output space, followed by an activation function suited to the task. Heads can be trained jointly in a multi-task setting or independently.

3.4. Training strategy

To handle class imbalance and label noise, a composite loss function is employed. This combines focal loss, supervised contrastive loss, Dice loss, and label smoothing. The total loss is calculated as a weighted sum, with weights tuned for stability and performance.

3.5. Training setup and hyperparameters

Most training settings are shared across tasks, with only a few parameters differing slightly. Key components include the base model (`microsoft/deberta-v3-small`), optimizer (`AdamW`), learning rate scheduling (CosineAnnealingLR with warmup), mixed precision training (AMP), and training for up to 5 epochs with early stopping. The complete set of standard hyperparameters used across experiments is summarized in Table 1. Any task-specific variations (e.g., batch size, input length, or loss function) are detailed in the case study section.

Table 1
Standard training setup and hyperparameters for transformer-based text classification.

Component	Setting
Model	`microsoft/deberta-v3-small` (transformer encoder)
Epochs	5 (early stopping on validation Macro-F1)
Optimizer	AdamW
Learning Rate Scheduler	Cosine annealing with warmup
Mixed Precision	Enabled (Automatic Mixed Precision, AMP)
Batch Size	16 (adjustable to hardware)
Maximum Input Length	256 tokens
Tokenization	HuggingFace DeBERTa tokenizer
Loss Function	Focal Loss (commonly used for class imbalance), optionally combined with Dice Loss, Label Smoothing, or Supervised Contrastive Loss
Evaluation Metric	Macro-F1, CM, Recall, Precision, UMAP
Cross-Validation	5-fold stratified
Ensembling	Majority voting across folds

3.5.1. Cross-validation Strategy

To ensure robust and unbiased training and evaluation, we employ stratified k-fold cross-validation with $k = 5$. This approach preserves the class distribution across folds, which is crucial given the hierarchical and imbalanced nature of the data. For each fold, the model is trained on 80% of the data

and validated on the remaining 20%. Final performance metrics are computed by averaging results across all folds, providing a more reliable estimate of model generalization.

4. Case Study

4.1. Dataset

The dataset used in this study was released as part of the FIRE 2025 Shared Task on Cryptocurrency-Related Social Media Analysis. It comprised two distinct components:

- **Task 1:** Hierarchical opinion classification of cryptocurrency-related posts.
- **Task 2:** Query–comment relevance prediction in online discussion threads.

The posts were sourced primarily from Twitter and Reddit, with a smaller portion from YouTube. These sources reflected a range of communication styles, from short promotional snippets to emotionally charged opinions and technical discussions. Although the dataset was multilingual, only the English subset was used for this work.

In Task 1, each post was annotated across three hierarchical levels, whereas in Task 2, question–comment pairs were labeled as either *relevant* or *not relevant*.

4.2. Task 1 — Hierarchical Opinion Classification

Task 1 categorizes cryptocurrency-related social media posts according to a three-level hierarchy (see Figure 2):

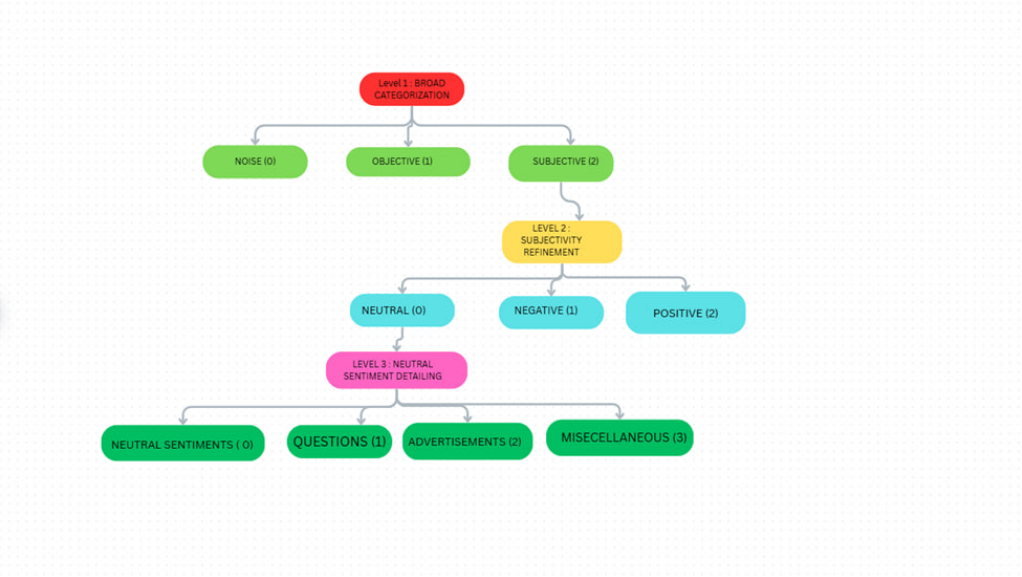


Figure 2: Task 1 – Hierarchical classification pipeline.

1. **Level 1 — Content Type:** *Noise*, *Objective*, or *Subjective*. Example: "Buy cheap crypto coins here!!!" → Noise (spam-like, non-informative). Filtering at this stage reduces spam and irrelevant content, which has been shown to improve opinion mining accuracy.
2. **Level 2 — Sentiment Polarity (only for Subjective posts):** *Positive*, *Negative*, or *Neutral*. Example: "Crypto is a scam, I lost everything" → Negative Subjective (strong negative sentiment).
3. **Level 3 — Communicative Intent (only for Neutral Subjective posts):** *Neutral Statement*, *Question*, *Advertisement*, or *Miscellaneous*. Example: "Will Ethereum outperform Bitcoin in 2025?" → Neutral Question.

This sequential hierarchy improves interpretability and progressively filters content for finer classification.

Preprocessing: All content from Reddit, Twitter, and YouTube was standardized into a single `text` field. For Reddit, the `title`, `selftext`, and `main` columns were concatenated. For Twitter, the `tweet` column was used directly, and for YouTube, the `comment` column was used. A platform-specific tag (`[REDDIT]`, `[TWITTER]`, or `[YOUTUBE]`) was prepended to encode the source. We intentionally did not strip HTML tags, as the `DeBERTa-v3-small` model can leverage raw text including markup [2, 9]. This unified dataset enabled joint training without separate models [16], while platform tags preserved linguistic cues and reduced bias [12].

This preprocessing stage forms the foundation of our hierarchical opinion classification framework, and its role in the overall workflow is shown in Figure 3.

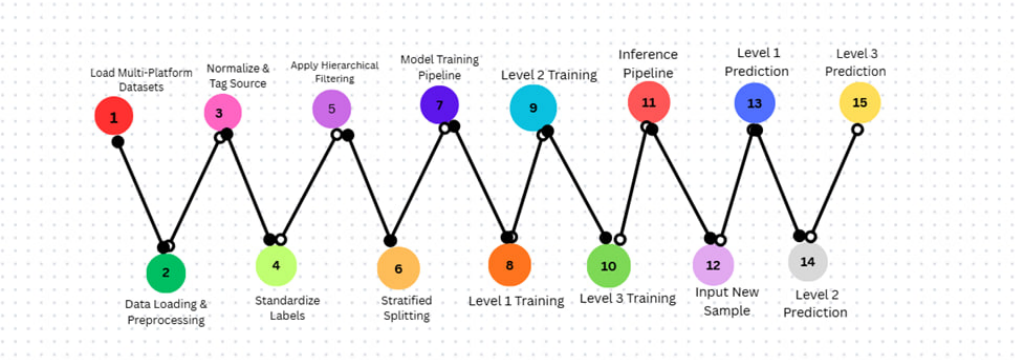


Figure 3: Hierarchical Opinion Classification Model Architecture for Task 1.

Filtering for Training: From Level 1 to Level 2, only samples predicted as Subjective were passed to sentiment classification. From Level 2 to Level 3, only Neutral Subjective samples were retained for communicative intent classification.

Model and Training Setup: We adopted `DeBERTaV3-small` for its balance of efficiency and accuracy. A 5-fold stratified cross-validation ensured robustness, and final Task 1 results were computed by averaging predictions across folds in an ensemble.

Loss Strategy: Level 1 used Focal Loss to address class imbalance. Levels 2 and 3 used a combined loss:

$$L_{\text{total}} = L_{\text{focal}} + \lambda_1 L_{\text{sup\_con}} + \lambda_2 L_{\text{dice}}, \tag{1}$$

where $\lambda_1 = 0.3$ and $\lambda_2 = 0.2$. Here:

- **Focal Loss** — focuses on hard misclassified samples.
- **Supervised Contrastive Loss** — improves embedding separability.
- **Dice Loss** — optimizes label overlap in imbalanced settings.

Label smoothing ($\epsilon = 0.1$) further reduced overconfidence.

Intermediate Outputs and Pipeline Flow: After each level, predicted class probabilities and labels were saved as `.pkl` files. These served as the filtered training set for the subsequent level, ensuring end-to-end consistency across the hierarchical pipeline.

4.3. Task 2 — Query–Comment Relevance Prediction

In this task, we aim to determine whether a Reddit comment genuinely addresses the question it follows. This is not as straightforward as matching keywords — as shown in Figure 4, many replies weave in sarcasm, wander off-topic, or present misleading information. Correctly identifying such cases as irrelevant is key to ensuring the quality of the relevance predictions.The relevance task was treated as binary classification, with labels 0 = Irrelevant and 1 = Relevant.

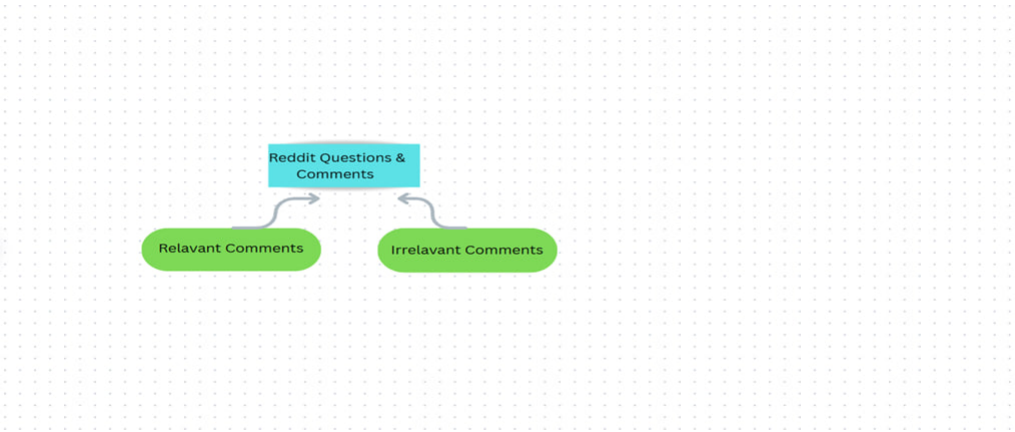


Figure 4: Task 2 — Relevance prediction between Reddit questions and comments.

Data Setup For this task, each input instance was constructed by concatenating multiple textual fields from the Reddit dataset in the following order:

$$\text{Input Text} = \texttt{title} + \texttt{selftext} + [\text{MAIN}] + \texttt{comment}.$$

This representation ensures that the model receives the post's title and self-description before the target comment, separated by a special [MAIN] token to mark the transition from query to comment. Given the substantial class imbalance—where irrelevant comments greatly outnumber relevant ones—we adopted a class-weighted Focal Loss [12] to mitigate bias towards the majority class, rather than relying on oversampling or synthetic augmentation. The complete architecture used for this classification task is shown in Figure 5,

This shows how the encoded representation of the concatenated input flows through the model's components to produce the final relevance prediction.

Model Configuration: We fine-tuned the `microsoft/deberta-v3-small` model [9] with a binary classification head. The focal loss hyperparameters were set to $\alpha = 0.75$ and $\gamma = 1.5$, following recommendations in [12]. Optimization was performed using the AdamW algorithm with weight decay, combined with a cosine learning rate scheduler incorporating warmup phases for stable convergence.

Training employed 5-fold stratified cross-validation to ensure robustness across balanced splits. Automatic Mixed Precision (AMP) was used to improve computational efficiency. Unlike Task 1, no model ensembling was applied.

Model Architecture: Figure 5 illustrates the model pipeline: the concatenated input text is tokenized and passed through the DeBERTa transformer backbone [2, 9], followed by a task-specific classification head to predict the relevance score.

This approach enables effective learning from combined multi-source textual fields while handling class imbalance through focal loss, thereby improving the model's focus on difficult, minority-class examples.

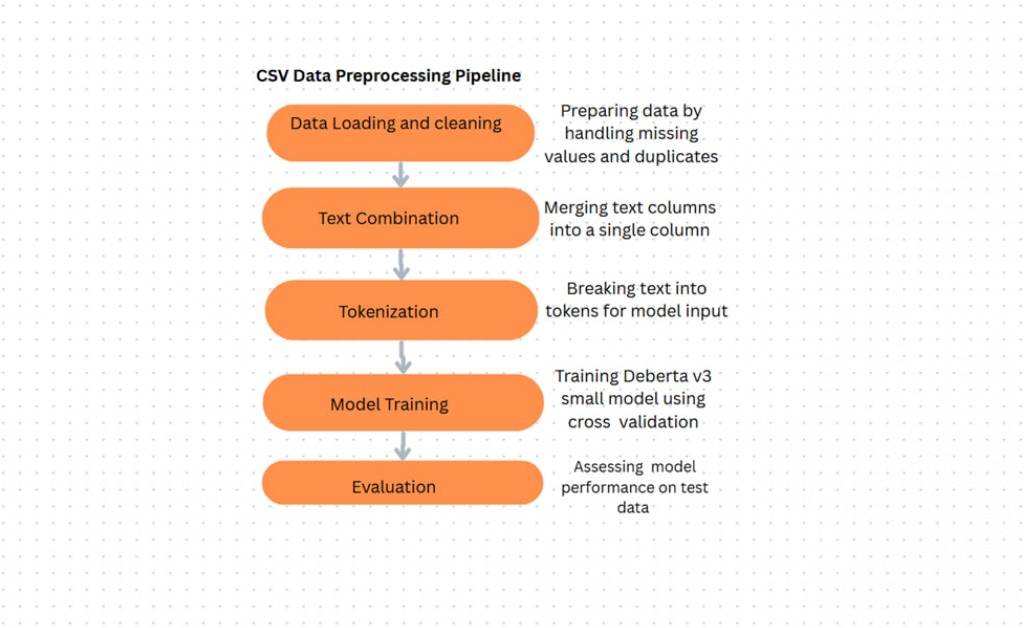


Figure 5: Architecture for Reddit query–comment relevance classification.

5. Results and Discussion

5.1. Task 1: Hierarchical Classification Results Summary

The hierarchical classification pipeline was evaluated on 1,429 validation samples from Reddit (500), YouTube (500), and Twitter (429). It consists of three levels, with the detailed metrics for each platform presented in Table 2, and the corresponding confusion matrices (CMs) and UMAP visualizations shown in Figure 6.

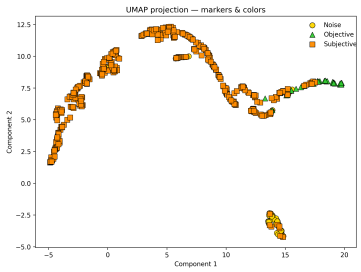
- **Level 1 (Subjectivity Detection)** achieved an overall accuracy of 83.8% and a Macro F1 score of 0.78. Reddit and YouTube performed best at this level, with accuracies of 89.4% and 90.8% respectively, while Twitter lagged behind at 69.2%.
- **Level 2 (Sentiment Classification)** processed subjective posts and reached an overall accuracy of 81.9% and a Macro F1 score of 0.64. Reddit and YouTube maintained strong performance (83.4% and 82.5% accuracy), while Twitter scored lower at 75.2%.
- **Level 3 (Intent Classification)** was the most challenging, showing an overall accuracy of 76.0% and a Macro F1 score of 0.41. Reddit and YouTube achieved 80.1% and 88.9% accuracy, but Twitter dropped to 23.5% due to noisy and short content as well as limited class diversity.

Overall, the pipeline demonstrates robust performance on Reddit and YouTube across all levels, while Twitter's results reflect inherent platform challenges affecting classification quality.

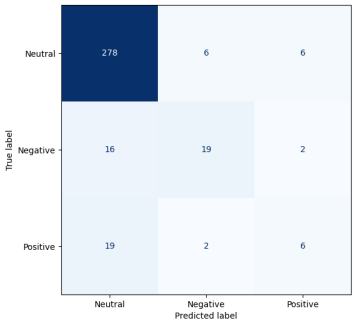
Table 2
Hierarchical classification results per platform and level.

Platform	Level	Accuracy	F1 Weighted	F1 Macro
Overall	Level 1	0.8383	0.8424	0.7832
Overall	Level 2	0.8190	0.8001	0.6440
Overall	Level 3	0.7598	0.7099	0.4122
Reddit	Level 1	0.8940	0.8971	0.8159
Reddit	Level 2	0.8338	0.8102	0.5546
Reddit	Level 3	0.8013	0.7563	0.3836
Twitter	Level 1	0.6923	0.6847	0.6786
Twitter	Level 2	0.7520	0.7243	0.5523
Twitter	Level 3	0.2353	0.1664	0.2167
YouTube	Level 1	0.9080	0.9077	0.6915
YouTube	Level 2	0.8254	0.8154	0.6098
YouTube	Level 3	0.8898	0.8829	0.5951

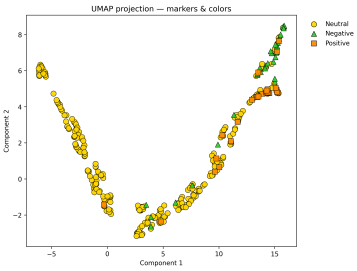
(a) Reddit Level 1 CM
Minor misclassifications between Subjective and others.



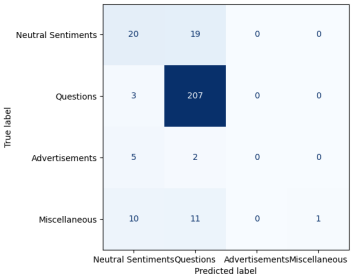
(b) Reddit Level 1 UMAP Visualization
Good class separation; clean cluster boundaries.



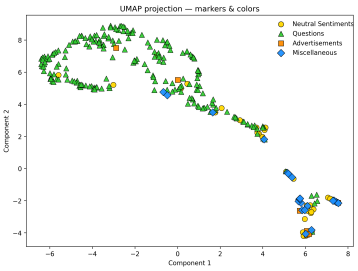
(c) Reddit Level 2 CM
Neutral dominates; Positive and Negative often confused.



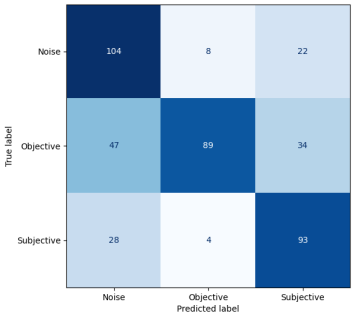
(d) Reddit Level 2 UMAP Visualization
Dense Neutral clustering; minority classes spread and overlap.



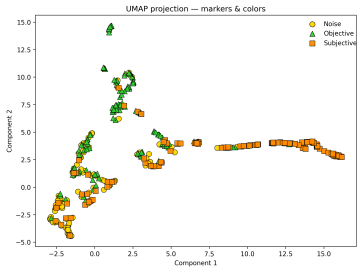
(e) Reddit Level 3 CM
Heavy bias toward Questions; Ads and Miscellaneous ignored.



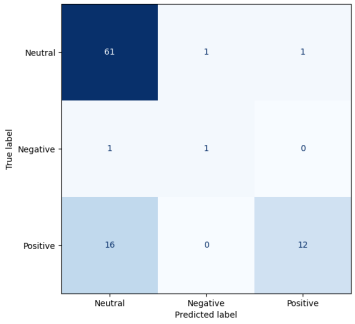
(f) Reddit Level 3 UMAP Visualization
Sparse class points blend into dominant clusters.



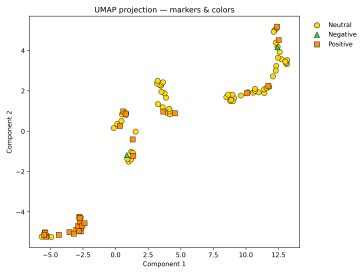
(g) Twitter Level 1 CM
Noticeable confusion between Subjective and Objective posts.



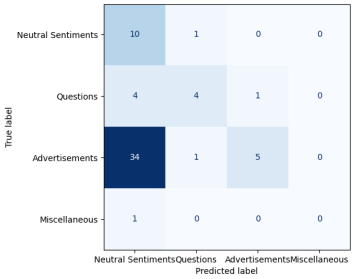
(h) Twitter Level 1 UMAP
Overlapping clusters with less clear boundaries.



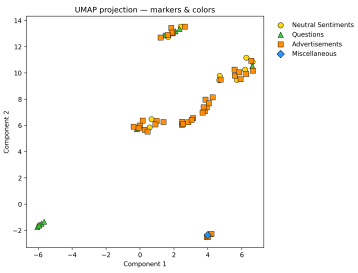
(i) Twitter Level 2 CM
Positive and Negative sentiments confused with Neutral.



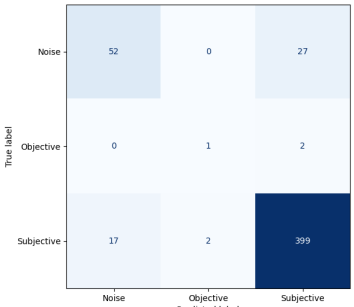
(j) Twitter Level 2 UMAP
Dense Neutral cluster with scattered minority classes.



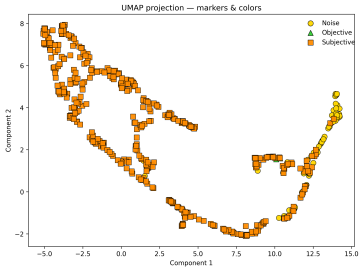
(k) Twitter Level 3 CM
Sparse classes and noisy data cause misclassifications.



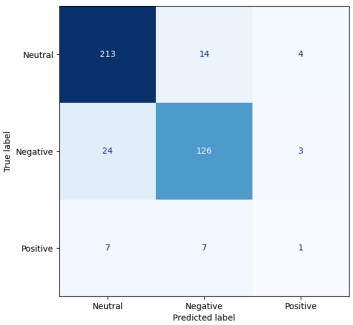
(l) Twitter Level 3 UMAP
Sparse class points overlap dominant clusters.



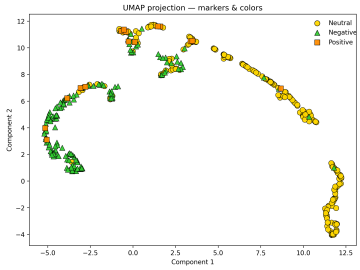
(m) YouTube Level 1 CM
Clear distinction for Subjective and Noise classes.



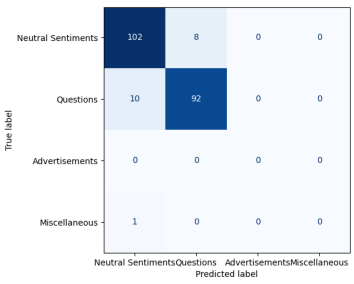
(n) YouTube Level 1 UMAP
Clear class separation suggests semantic coherence.



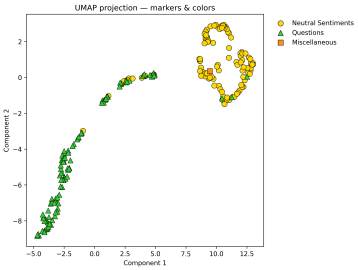
(o) YouTube Level 2 CM
Some confusion between Neutral and Negative classes.



(p) YouTube Level 2 UMAP
Strong separation even with class imbalance.



(q) YouTube Level 3 CM
High alignment with dominant ground truth classes.



(r) YouTube Level 3 UMAP
Strong cluster cohesion for key classes.

Figure 6: Confusion Matrices (CM) and UMAP visualizations for Reddit, Twitter, and YouTube across all three classification levels

5.2 Task 2: Relevance Prediction across Cross-validation Performance

Based on the cross-validation results presented in Table 3, Fold 2 was selected for the final submission. Although its accuracy (0.8819) was slightly below the mean, it achieved the highest Macro F1 score (0.7002) among all folds, which was prioritized as the primary evaluation metric. The confusion matrix for Fold 2 is shown in Figure 7. Due to time constraints, ensemble learning techniques—such as model averaging or stacking across all five folds—were not explored. Such methods could potentially improve performance beyond relying on a single fold model.

Table 3: Performance metrics across 5 folds

Fold	Accuracy	Macro F1	Weighted F1	Macro Precision	Macro Recall
Fold 1	0.8829	0.6853	0.8667	0.7687	0.6509
Fold 2	0.8819	0.7002	0.8699	0.7574	0.6702
Fold 3	0.8858	0.6663	0.8631	0.8035	0.6282
Fold 4	0.8844	0.6839	0.8671	0.7767	0.6480
Fold 5	0.8862	0.6865	0.8686	0.7861	0.6491
Mean	**0.8842**	**0.6844**	**0.8671**	**0.7785**	**0.6493**
Std. Dev.	0.0017	0.0124	0.0025	0.0153	0.0154

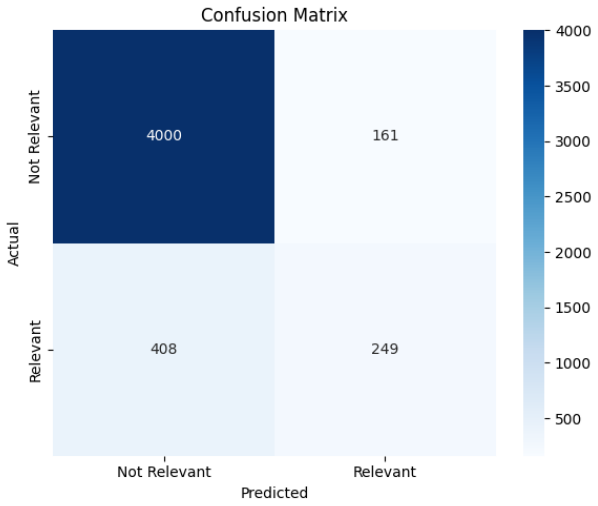


Figure 7: Confusion Matrix for Fold 2

5.3 FIRE 2025 Task 1 and Task 2 Test Evaluation

While our framework showed competitive performance during validation, it also exhibited some limitations. In Task 1, validation accuracy was comparatively lower for the Twitter subset, and in Task 2, relevance prediction results indicated room for improvement. However, according to confirmation from the FIRE 2025 CryptoNLP task organizers, our system ultimately achieved the top rank in both Task 1 and Task 2 on the hidden test leaderboard with a reported Macro-F1 score of 1.0. These complementary results suggest that, despite weaker validation outcomes in certain subsets, the approach generalized exceptionally well on the unseen test dataset, underscoring the robustness of hierarchical modeling combined with advanced loss functions.

6. Conclusion and Future Work

This study explored two important NLP problems under the FIRE 2025 Shared Task — hierarchical opinion classification and Reddit QA relevance prediction. For Task 1, a level-wise modular architecture was built using DeBERTaV3-small with a hierarchical pipeline of three classifiers. Advanced loss

functions like Focal Loss, Dice Loss, and Supervised Contrastive Learning were used instead of data augmentation. Intermediate filtered outputs (.pkl) enabled efficient level-based training, and ensemble evaluation improved generalization across social media platforms. For Task 2, a binary DeBERTa-based model was trained with class balanced focal loss and threshold tuning, achieving stable macro-F1 scores. Fold 2 was chosen for reporting due to its consistent precision-recall trade-off. Future improvements could include scaling up to larger backbone models (DeBERTaV3 base/large) and applying ensemble inference techniques such as soft or majority voting. Additionally, retrieval-augmented generation (RAG), task-specific pretraining, or contrastive alignment between question and comment pairs could enhance the semantic understanding of long form crypto discourse. Addressing data sparsity at fine-grained levels and incorporating prompt based transformers are also promising directions

Acknowledgements

We thank the FIRE 2025 CryptOQA organizers and coordinator for their guidance, and acknowledge Jim Ureel (Abilene Christian University) for materials that supported improvements in manuscript clarity.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly for grammar correction. Figures were designed by the authors using Canva. All outputs were reviewed and finalized by the authors to ensure accuracy and originality. No generative AI tools were used for producing the core text or experimental results.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proc. Conf. North American Chapter of the Association for Computational Linguistics (NAACL), 2019. [Online]. Available: https://arxiv.org/abs/1810.04805.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems (NeurIPS), 2017. [Online]. Available: https://arxiv.org/abs/1706.03762.

[3] S. Mohapatra, R. Sharma, Crypto-emotion bert: A multi-label transformer-based framework for emotion detection in cryptocurrency tweets, Procedia Computer Science 199 (2022) 872–879. doi:10.1016/j.procs.2022.01.109, [Online]. Available: https://doi.org/10.1016/j.procs.2022.01.109.

[4] K. I. Roumeliotis, N. D. Tselikas, D. K. Nasiopoulos, Llms and nlp models in cryptocurrency sentiment analysis: A comparative classification study, Big Data and Cognitive Computing (2024). doi:10.3390/bdcc8060063, [Online]. Available: https://doi.org/10.3390/bdcc8060063.

[5] S. C. Long, Y. Xie, Z. Zhou, B. M. Lucey, A. Urquhart, From whales to waves: Social media sentiment, volatility, and whales in cryptocurrency markets, Journal of Financial Markets (2025). doi:10.1016/j.bar.2025.101682, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0890838925001325.

[6] S. C. Long, Y. Xie, Z. Zhou, A. Urquhart, Sentiment matters for cryptocurrencies: Evidence from tweets, Data 10 (2023). doi:10.3390/data10040050, [Online]. Available: https://www.mdpi.com/2306-5729/10/4/50.

[7] FIRE 2025 Shared Task Organizers, Cryptoqa 2025: Subjectivity and relevance analysis on social media [online], https://sites.google.com/view/cryptoqa-2025/task-description, 2025. Accessed: July 2025.

[8] S. Huang, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient disentangled embedding sharing, in: arXiv preprint arXiv:2111.09543, 2022. [Online]. Available: https://arxiv.org/abs/2111.09543.

[9] P. Zhang, X. He, J. Gao, W. Chen, Deberta: Decoding-enhanced BERT with disentangled attention, arXiv preprint arXiv:2006.03654 (2021). [Online]. Available: https://arxiv.org/abs/2006.03654.

[10] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, J. Li, Dice loss for data-imbalanced nlp tasks, in: Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 465–476. [Online]. Available: https://aclanthology.org/2020.acl-main.45.

[11] Y. Gao, S. Si, H. Luo, H. Sun, Y. Zhang, Revisiting label smoothing in transformer-based text sentiment classification, Expert Systems with Applications 220 (2023) 119482. [Online]. Available: https://arxiv.org/abs/2312.06522.

[12] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. H. S. Torr, P. K. Dokania, Calibrating deep neural networks using focal loss, arXiv preprint arXiv:2002.09437 (2020). URL: https://arxiv.org/abs/2002.09437. doi:10.48550/arXiv.2002.09437, accepted at NeurIPS 2020.

[13] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, in: Advances in Neural Information Processing Systems (NeurIPS), 2020. [Online]. Available: https://arxiv.org/abs/2004.11362.

[14] B. Günel, J. Du, R. Takács, P. Chang, K. Lee, Supervised contrastive learning for pre-trained language model fine-tuning, in: Proc. Int. Conf. Learning Representations (ICLR), 2021. [Online]. Available: https://arxiv.org/abs/2011.01403.

[15] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. [Online]. Available: https://arxiv.org/abs/1910.10683.

[16] T. Gao, X. Yao, D. Chen, Simcse: Simple contrastive learning of sentence embeddings, in: Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 2021. [Online]. Available: https://arxiv.org/abs/2104.08821.

[17] V. Koltun, I. Yamshchikov, Pump it: Twitter sentiment analysis for cryptocurrency price prediction, Information 14 (2023). doi:10.3390/info14070401, [Online]. Available: https://www.mdpi.com/2078-2489/14/7/401.

[18] S. Sarkar, A. Badwal, A. Roy, K. Rudra, K. Ghosh, Cryptopiqa: A new opinion and question answering dataset on cryptocurrency, in: Proc. 31st Intl. Conf. Computational Linguistics (COLING), 2025. [Online]. Available: https://aclanthology.org/2025.coling-main.736.pdf.

[19] J. Opitz, S. Burst, Macro f1 and macro f1, in: Proc. Int. Conf. Machine Learning and Data Mining (MLDM), 2019, pp. 35–48. [Online]. Available: https://arxiv.org/abs/1911.03347.

[20] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018). [Online]. Available: https://arxiv.org/abs/1802.03426.