NUST PsyAI at SemEval-2026 Task 10: Parameter-Efficient RoBERTa for Conspiracy Detection and Character-Level Marker Extraction

M. Husnain Akram and **Mehwish Fatima**
School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST), Islamabad, Pakistan
{makram.mscs24seecs, mehwish.fatima}@seecs.edu.pk

Abstract

We present the NUST PsyAI system for SemEval-2026 Task 10 (PsyCoMark), targeting document-level conspiracy detection and character-level psycholinguistic marker extraction from Reddit discourse. Our system ranks 7th in Extraction and 8th in Detection on the leaderboard. We benchmark feature-based and transformer approaches, adopting RoBERTa-large with LoRA for parameter-efficient fine-tuning. For detection, RB-DET-LoRA outperforms all baselines, achieving weighted F1 >0.79 (dev) and 0.76 (test), with robust generalization under blinded evaluation. For extraction, we contrast a unified multi-type BIO scheme with a decomposed per-type setup; the latter mitigates cross-label interference and improves boundary consistency, reaching Overlap F1 of 0.16 (dev) and 0.21 (test). Results reveal a clear asymmetry: detection benefits from contextual semantic modeling, while extraction is limited by sparse supervision and boundary-sensitive evaluation.

1 Introduction

The rapid proliferation of conspiracy narratives on social media undermines public discourse and information integrity. SemEval-2026 Task 10 (PsyCoMark) [Ghosh et al., 2026] formalizes this problem through two coupled objectives: (i) document-level conspiracy detection and (ii) character-level extraction of psycholinguistic markers—*Actor*, *Action*, *Victim*, *Effect*, and *Evidence*. These objectives impose fundamentally different modeling requirements: detection depends on global semantic coherence and discourse-level signals, while extraction requires boundary-sensitive sequence labeling under sparse supervision. Bridging this gap demands models that jointly capture contextual semantics and fine-grained span structure.

We design a unified, transformer-centric framework with parameter-efficient adaptation to address both sub-tasks. We systematically evaluate feature-based baselines and pretrained encoders, and adopt RoBERTa-large with LoRA to balance representational capacity and efficiency. For detection, RB-DET-LoRA consistently outperforms both feature-based methods and fully fine-tuned transformer baselines, demonstrating strong generalization under blinded evaluation. For extraction, we explicitly compare a unified multi-type BIO formulation with a decomposed five-model setup; the latter assigns an independent classifier per marker type, reduces cross-label interference, and improves boundary stability.

Our contributions are threefold. First, we provide a systematic comparison of feature-based, augmented, and parameter-efficient transformer models for conspiracy detection. Second, we present an empirical analysis of unified versus decomposed formulations for character-level span extraction. Third, we conduct a focused analysis of boundary errors under IoU-based evaluation, exposing structural limitations of current approaches to fine-grained psycholinguistic marker extraction.

2 Related Work

Conspiracy detection evolves from feature-engineered pipelines to context-aware transformer models. Early approaches use psycholinguistic, lexical, and dictionary-based features with classifiers such as SVMs and Random Forests to capture cognitive and emotional signals in conspiratorial discourse [Bessi et al., 2015, Klein et al., 2019, Ferrara, 2020, Miani et al., 2021]. These methods remain interpretable and efficient, but depend on static vocabularies and degrade under linguistic drift and platform-specific variation.

Transformer-based models improve performance by modeling contextual semantics and long-range dependencies. Fine-tuned BERT variants achieve strong results on misinformation and social media

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1298–1307

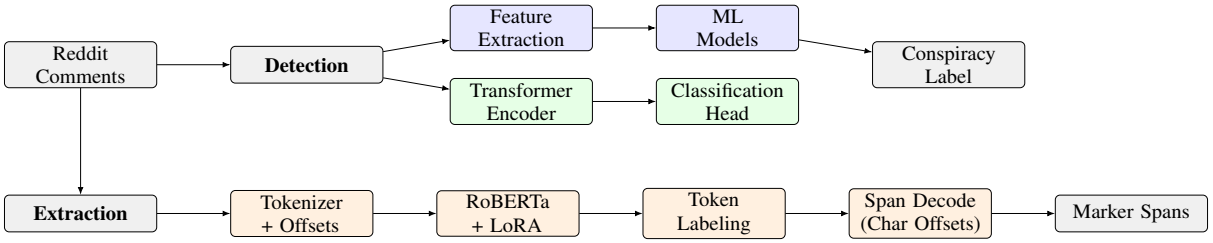

Figure 1: Two-branch conspiracy analysis framework: the detection branch combines feature-based models with transformer encoders for document-level prediction, while the extraction branch performs BIO-based token classification with RoBERTa-large (LoRA) followed by character-level span reconstruction.

benchmarks [Hoseini et al., 2023, Wadden et al., 2020, Min et al., 2022], but typically operate at sentence or document level and provide limited resolution for span-level reasoning. Hybrid approaches integrate psycholinguistic features with contextual embeddings to balance interpretability and representational capacity [Giachanou et al., 2023].

Recent work explores large language models, prompting strategies, and graph-based formulations to model narrative structure and information propagation [Zhang et al., 2024, Bang et al., 2023, Pan et al., 2023, Wang et al., 2024]. These methods expand semantic coverage and reasoning ability, but increase computational cost and remain underexplored for boundary-sensitive extraction.

Parameter-efficient fine-tuning methods, particularly LoRA, reduce adaptation cost while preserving the expressive power of large transformers [Hu et al., 2021, Ding et al., 2023, Chen et al., 2025]. However, existing work focuses primarily on coarse-grained classification. Fine-grained, character-level extraction of psycholinguistic roles—especially under strict boundary-based evaluation—remains insufficiently studied.

3 Our Framework

We propose a two-branch architecture addressing (i) document-level detection and (ii) character-level marker extraction (Figure 1). The design decouples global semantic modeling from boundary-sensitive span prediction. To ensure reproducibility, we release our exploratory data analysis, processing pipelines, and full framework[1].

3.1 Document-Level Conspiracy Detection

This section discusses document-level detection pipeline.

[1]NUST PsyAI: EDA-Rehydrated Notebooks

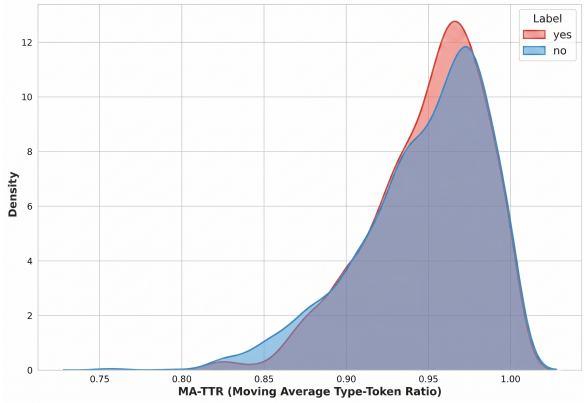

Figure 2: MA-TTR density plot showing lexical diversity differences between conspiracy (Yes) and non-conspiracy (No) texts.

3.1.1 Feature-Based Modeling

We compute lexical diversity (TTR, MATTR, MTLD [McCarthy and Jarvis, 2010]), readability indices (Flesch, FKGL, SMOG), POS distributions, discourse markers, and sentiment features using standard NLP toolkits [Bird et al., 2009, Honnibal and Montani, 2017, Hutto and Gilbert, 2014]. This yields a 62-dimensional feature space capturing stylistic and structural variation beyond surface lexical cues. For example, MA-TTR distributions (Figure 2) highlight clear differences in lexical diversity between conspiracy and non-conspiracy texts, indicating distinct narrative construction patterns. Following task guidelines, we use binary labels (*Yes*, *No*) and exclude ambiguous cases.

3.1.2 Statistical Filtering

We perform univariate t-tests to identify discriminative features ($p < 0.05$), including stopword ratio, adverb usage, interrogatives, pronouns, and passive constructions (Table 1). These patterns reflect stylistic and rhetorical differences in conspiratorial discourse.

3.1.3 Classifiers

We evaluate Linear Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and XG-

Feature	t	p
stopword_ratio	+3.87	**0.00011**
n_advs_x	+3.81	**0.00014**
qmarks	+3.11	**0.0019**
commas	-3.00	**0.0027**
n_nouns_x	-2.82	**0.0049**
n_adjs_x	-2.67	**0.0076**
discourse_count	+2.45	**0.0145**
ttr	+2.43	**0.0150**
n_pronouns_x	+2.34	**0.0195**
n_sentences_x	+2.22	**0.0267**
allcaps_ratio	+2.04	**0.0419**
passive_count	+2.05	**0.0406**
hedge_count	-1.74	0.0810
url_count	+0.19	0.8500

Table 1: Univariate t-test results for psycholinguistic features; bold p-values denote significance ($p < 0.05$).

Boost on the engineered feature space to model stylistic separability.

3.1.4 Transformer-Based Detection

We fine-tune pretrained encoders directly on raw text. DistilBERT and BERT serve as full fine-tuning baselines.

3.1.5 Data-Augmented RoBERTa

We evaluate data augmentation through two configurations: RB-DET-AG2× (contextual synonym substitution using RoBERTa-base) and RB-DET-AG3× (a hybrid of synonym substitution and EN→DE→EN back-translation via MarianMT). We accept augmented samples only if they pass a surface-form quality gate that filters degenerate or identical variants. We integrate this synthetic data using a layered fine-tuning strategy, enabling the model to adapt to the expanded feature space while preserving core semantic representations (see Appendix C for pipeline details and failure analysis).

3.1.6 PEFT with LoRA

We adopt LoRA over RoBERTa-large (RB-DET-LoRA) and inject low-rank adapters ($r = 16$, $\alpha = 32$), reducing trainable parameters while preserving representational capacity.

3.2 Marker Extraction Architecture

This section presents how markers are extracted.

3.2.1 Task Formulation

We cast extraction as token-level BIO tagging aligned to character spans via tokenizer offsets. Predictions are projected back to character space for evaluation.

3.2.2 Decoding and Span Reconstruction

Predicted BIO sequences are merged into spans and mapped to character offsets. Invalid fragments are discarded, and overlaps are resolved via confidence-based selection. Outputs conform strictly to required span format.

3.2.3 Model Variants

We compare: (i) a unified multi-type model (**RB-EXT-MT**) with shared label space, and (ii) a decomposed setup (**RB-EXT-5M**) with five independent classifiers. We apply LoRA to improve parameter efficiency across both formulations.

4 Experimental Design

This section covers the details of experimental configurations.

4.1 Dataset

We use the PsyCoMark dataset of SemEval-2026 Task 10[2]. It comprises three splits: Train ($N = 2{,}929$), Development (hereafter `dev_rehydrated`) ($N = 100$), and Test (hereafter `test_rehydrated`) ($N = 938$). Each instance contains a Reddit comment with detection labels in $\{$*Yes*, *No*, *Can't tell*$\}$ and character-level annotations for marker extraction.

We use the Train set with gold labels for supervised learning. We initially use the Development set without labels for model development and later incorporate released labels for internal evaluation and refinement. We treat `test_rehydrated` as the held-out set for official evaluation on CodaBench[3]. For detection, we restrict experiments to binary labels (*Yes*, *No*) and discard ambiguous cases.

4.2 Evaluation Protocol

We evaluate detection using weighted F1 (W-F1) (official metric) and report Macro F1 (Ma-F1) for additional analysis. For extraction, we use Overlap F1 (O-F1) under a token-IoU protocol (IoU ≥ 0.5): we map character spans to token sets and enforce type-constrained one-to-one alignment. This setup makes boundary precision the dominant factor (see Appendix A, Table 8).

4.3 Extraction Setup

We align subword tokens with character-level spans using tokenizer offset mappings and as-

[2]Official Starter Pack
[3]Competition Page

4-Fold Cross-Validation (Train: $N = 2{,}929$)			
Model	**Acc.**	**Ma-F1**	**W-F1**
GNB	0.57	0.51	0.53
L-SVM	0.58	0.54	0.56
RF	**0.59**	0.55	0.57
LR	0.59	**0.56**	**0.57**
XGB	0.57	0.55	0.56
DT	0.49	0.48	0.49
MLP	0.56	0.50	0.55
Codabench dev_rehydrated ($N = 100$)			
Model	**Acc.**	**Ma-F1**	**W-F1**
GNB	**0.68**	0.47	0.57
L-SVM	0.65	0.56	0.62
RF	0.69	**0.58**	**0.64**
LR	0.61	0.46	0.54
XGB	0.62	0.57	0.62
DT	0.56	0.50	0.55
MLP	0.55	0.50	0.54

Table 2: Performance of feature-based models for binary conspiracy detection under cross-validation and official evaluation settings.

sign BIO tags. A single model predicts marker-specific BIO labels (`B/I-Action`, `B/I-Actor`, `B/I-Effect`, `B/I-Evidence`, `B/I-Victim`). In the decomposed setup, we train five one-vs-type BIO models and aggregate their outputs at inference time. We address label imbalance in decomposed training using class-weighted cross-entropy and apply early stopping based on validation F1.

4.4 Training and Implementation

We implement all models in PyTorch and train them in mixed precision (FP16) on 8× AMD MI250X GPUs. We use an effective batch size of 32 (via gradient accumulation) and a maximum sequence length of 512. We tune learning rates in $[5 \times 10^{-5}, 6 \times 10^{-5}]$ and train for 10–20 epochs depending on configuration.We apply LoRA to reduce trainable parameters and memory footprint, enabling efficient adaptation of RoBERTa-large without full fine-tuning. Additional implementation details, including libraries and preprocessing components, appear in Appendix A.

5 Results and Analysis

We present results for both detection and extraction, analyzing model behavior across feature-based and transformer settings.

5.1 Document-Level Conspiracy Detection

We evaluate models for document-level prediction under both feature-driven and contextual paradigms.

TreeSHAP	**t-test**	t**-statistic**	p**-value**
adj_ratio	stopword_ratio	+3.87	0.0001
noun_ratio	n_advs_x	+3.81	0.0001
num_ratio	qmarks	+3.11	0.0019
adp_ratio	commas	-3.00	0.0027
adv_ratio	n_nouns_x	-2.82	0.0049

Table 3: Comparison of top-ranked features from TreeSHAP (by impact) and univariate tests (by significance).

5.1.1 Feature-Driven Detection

Logistic Regression achieves the strongest cross-validation performance (Ma-F1 = 0.56, W-F1 = 0.57), indicating that linear boundaries capture a meaningful portion of stylistic variation. On the `dev_rehydrated` set, Random Forest generalizes better and achieves the highest W-F1 (0.64). Overall, feature-based models capture moderate signal but remain clearly below contextual models. Table 2 summarizes these results across cross-validation and held-out evaluation.

We analyze feature relevance using univariate t-tests and TreeSHAP for Random Forest (Table 3). The two analyses reveal complementary patterns: t-tests highlight frequency-based signals such as `stopword_ratio`, `n_advs_x`, and `n_pronouns_x`, while TreeSHAP emphasizes structural ratio features (e.g., `adj_ratio`, `noun_ratio`). This alignment at the category level—particularly for adverb-related features—indicates that the model captures linguistically meaningful signals rather than spurious correlations.

TreeSHAP further highlights structural and part-of-speech ratios as dominant signals, complementing the frequency-based patterns identified by univariate tests. The model places greater emphasis on ratio-based features, suggesting that relative distributional patterns provide stronger discriminative power than raw counts. We provide additional analysis of feature attribution for the Random Forest (RF) classifier using TreeSHAP, extending the observations discussed in Appendix B.

5.1.2 Transformer-Based Detection

Table 4 summarizes results on `dev_rehydrated`. DistilBERT achieves high accuracy (0.79) but lower W-F1(0.67), indicating imbalance sensitivity. BERT further underperforms, particularly on the minority class. Data augmentation provides inconsistent gains. While RB-DET-AG variants improve recall for the *Yes* class, they introduce noise that degrades overall performance. Qualitative analysis shows that back-translation distorts conspiracy-specific cues, limiting effectiveness.

Model	Acc.	Ma-F1	W-F1	No-F1	Yes-F1
DistilBERT	0.79	0.75	0.67	0.85	0.64
BERT	0.65	0.59	0.63	0.75	0.43
RB-DET-AG2×	0.70	0.63	0.68	0.79	0.47
RB-DET-AG3×	0.62	0.41	0.52	0.76	0.06
RB-DET-LoRA	**0.83**	**0.82**	**0.83**	**0.87**	**0.76**

Table 4: Post-submission detection performance on `dev_rehydrated` ($N = 100$); official system achieves 0.79 (dev) and 0.76 (test, $N = 938$) weighted F1.

We provide a detailed analysis of back-translation effects in Appendix C. Table 7 illustrates these failure modes, including lexical hallucination and semantic drift introduced by back-translation. Our layered fine-tuning strategy (unfreezing top 12–18 layers) does not compensate for this semantic drift. Consequently, parameter-efficient adaptation via LoRA proves more robust than expanding the training set with noisy synthetic samples.

RB-DET-LoRA achieves a W-F1 of 0.79 during the official development phase, where we evaluate on CodaBench using the organizers' pipeline and hidden labels. After the development phase ends and gold labels for the `dev_rehydrated` ($N = 100$) set become available, we continue task-specific fine-tuning and re-evaluate using the same official script, achieving 0.83 on the same set. Figure 3 illustrates the corresponding class-wise prediction behavior.

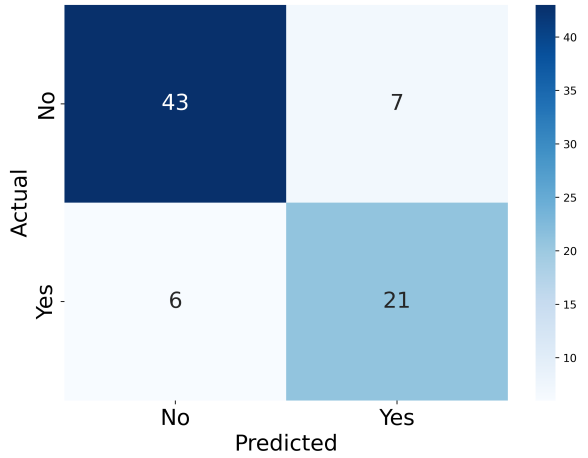

Figure 3: Confusion Matrix for the RoBERTa-large PEFT (LoRA) model on the `dev_rehydrated` set.

We further validate robustness using the official CodaBench evaluation. As shown in Figure 4, RB-DET-LoRA achieves a W-F1 of 0.76 on the held-out `test_rehydrated` set, confirming stable generalization under a larger, fully blinded distribution.

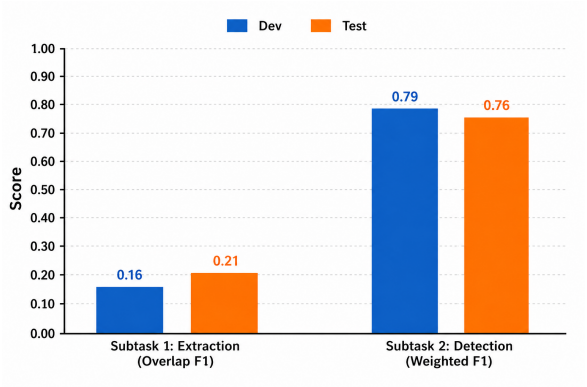

Figure 4: CodaBench scores for extraction (O-F1) and detection (Weighted F1) on dev and test sets.

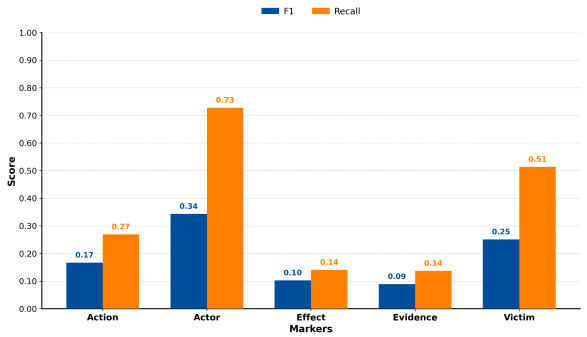

Figure 5: Per-role extraction performance showing F1 and recall across psycholinguistic categories.

5.2 Conspiracy Markers Extraction

We evaluate span-level extraction under unified and decomposed labeling strategies.

5.2.1 Transformer-Based Extraction

Table 6 shows that the unified multi-type model (RB-EXT-MT) performs poorly ($F1 = 0.05$), indicating instability under a shared label space. In contrast, the decomposed formulation (**RB-EXT-5M**) achieves significantly higher stability. During the official development phase, our system reaches an Overlap $F1 = 0.16$. After the release of the development `dev_public` ($N = 100$), we refine the model and reach 0.22 F1. We use the same optimized configuration for the final test submission, where it achieves a stable O-F1 of 0.21 on the blinded `test_rehydrated` set ($N = 938$). Figure 4 highlights that separating label spaces reduces cross-type interference and improves boundary consistency under strict IoU constraints.

5.2.2 Boundary Behavior and Error Patterns

Table 5 shows that extraction errors are dominated by boundary misalignment rather than categorical misclassification. Models successfully localize relevant discourse regions but systematically

Role	Precision	Recall	F1	TP	Pred	Gold
Action	0.1207	0.2692	0.1667	28	232	104
Actor	0.2245	0.7279	0.3432	99	441	136
Effect	0.0806	0.1408	0.1026	10	124	71
Evidence	0.0658	0.1370	0.0889	10	152	73
Victim	0.1659	0.5139	0.2508	37	223	72
Micro Avg	0.1570	0.4035	0.2260	184	1172	456

Table 5: Per-role token-overlap results (IoU ≥ 0.5) on `dev_rehydrated` vs. `dev_public`; TP, Pred, Gold denote true positives, predicted, and gold spans.

over-generate spans (1,172 predicted vs. 456 gold), resulting in high recall and low precision. This behavior arises from the type-separated decoding strategy, which preserves valid multi-role overlaps but increases false positives under strict matching (see Appendix D for decoding details).

The token-level Intersection-over-Union (IoU ≥ 0.5) evaluation protocol (Table 8) further amplifies this effect. Mapping character spans to discrete token sets makes the metric highly sensitive to boundary shifts: even minor misalignment or subword fragmentation reduces IoU below the threshold, converting semantic matches into false positives. Per-role analysis (Figure 5) shows that concrete roles (*Actor*, *Victim*) remain more robust to these shifts, while abstract roles (*Effect*, *Evidence*) exhibit greater degradation due to diffuse boundaries.

5.3 Comparative Observations

Across tasks, transformer models consistently outperform feature-based approaches for detection. RB-DET-LoRA achieves the best trade-off between performance and efficiency. In contrast, extraction remains bottlenecked by sparse supervision and boundary sensitivity; RB-EXT-5M provides a more stable formulation than RB-EXT-MT under these constraints.

6 Conclusions

We present the NUST PsyAI system for SemEval-2026 Task 10, where contextual transformers consistently outperform feature-based approaches. For detection, RB-DET-LoRA achieves the best performance, confirming that parameter-efficient adaptation effectively captures conspiratorial semantics while maintaining efficiency. For extraction, task decomposition (RB-EXT-5M) outperforms a unified formulation by reducing cross-type interference under sparse supervision. The results expose a clear asymmetry: document-level detection benefits from semantic modeling, whereas character-level extraction remains constrained by label sparsity and strict boundary evaluation. Future work will focus on unified multi-task learning to couple global and local signals, span-aware objectives (e.g., contrastive or boundary-aware losses) to improve alignment precision, and more robust training strategies to enhance cross-domain generalization without increasing inference complexity.

Configuration	Multi-Type	Five-Model
Label scheme	11 (joint BIO)	3 (O, B, I) per type
Max sequence length	256	512
Learning rate	3×10^{-4}	2×10^{-4}
Class weighting (B/I vs. O)	No	Yes
Effective batch size	16	16
Epochs / early stopping	10 / patience 3	10 / patience 3
LoRA rank (r)	16	16
Dev O-F1	0.05	**0.22**

Table 6: Comparison of Multi-Type vs. Five-Model extraction (O-F1) on `dev_rehydrated` ($N = 100$); Five-Model improves beyond the official 0.16 F1 after post-submission fine-tuning.

Limitations

The system exhibits several limitations. First, the dataset is sourced from Reddit that biases the model toward long-form, text-only discourse, limiting transfer to platforms with shorter formats, multimodal signals, or different linguistic distributions. Second, psycholinguistic markers exhibit extreme sparsity (exceeding a $1:100$ marker-to-background token ratio), making character-level boundary detection inherently unstable and prone to fragmentation under strict (IoU ≥ 0.5) evaluation. Third, while the RB-EXT-5M setup improves precision, it increases inference latency and computational overhead compared to unified architectures. Finally, the system remains sensitive to temporal drift in conspiracy narratives and subjectivity in role annotation, highlighting the need for cross-domain robustness and adaptive modeling.

Acknowledgements

We thank the organizers and reviewers of SemEval-2026 Task 10 for the PsyCoMark challenge, dataset curation, and transparent CodaBench evaluation, which together establish a rigorous benchmark for psycholinguistic analysis and conspiracy detection.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, and 1 others. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Alessandro Bessi, Mauro Coletto, George Alexandru Davidescu, Antonio Scala, Guido Caldarelli, and Walter Quattrociocchi. 2015. Science vs conspiracy: Collective narratives in the age of misinformation. *PloS one*, 10(2):e0118093.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Wei Chen, Yuxiao Dong, and Jie Tang. 2025. Efficient fine-tuning of large language models for misinformation detection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, and 1 others. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Emilio Ferrara. 2020. What types of covid-19 conspiracies are populated by twitter bots? *First Monday*, 25(6).

Debanjan Ghosh, Kai North, Ekaterina Kochmar, Mamoru Komachi, and Marcos Zampieri, editors. 2026. *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*. Association for Computational Linguistics, San Diego, United States.

Anastasia Giachanou, Bilal Ghanem, and Paolo Rosso. 2023. A psycholinguistic and transformer-based approach to conspiracy theory detection. *Information Processing & Management*, 60(3):103288.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Charles R. Harris and 1 others. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python. https://spacy.io.

Mahmoud Hoseini, Shirin Nilizadeh, and A. E. Çelik. 2023. Conspiracy theory detection on social media using transformers and linguistic features. *IEEE Transactions on Computational Social Systems*, 10(4):1854–1865.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, pages 216–225.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Guibeau, Kenneth Heafield, Nikolay Bogoychev, Alham Fikree Moore, and Kenneth Hogane. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 129–135.

Colin Klein, Peter Clutton, and Adam Dunn. 2019. Pathways to conspiracy: The social and linguistic precursors of involvement in reddit's conspiracy theory forum. *PloS one*, 14(11):e0225098.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*.

Edward Ma. 2019. Nlp augmentation library for deep learning. *GitHub repository*.

Philip M. McCarthy and Scott Jarvis. 2010. MTLD, vocd-d, and HD-D: A validation study of advanced lexical richness measures. *Behavior Research Methods*, 42(2):381–392.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11).

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Wes McKinney. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 56–61.

Alessandro Miani, Thomas Hills, and Adrian Bangerter. 2021. Loco: The 88-million-word language of conspiracy corpus. *Behavior Research Methods*, 53(4):1790–1805.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, and 1 others. 2022. Divide and conquer: A program representation for fact-checking. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Yikang Pan and 1 others. 2023. On the risks of large language models in generating misinformation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Fabian Pedregosa and 1 others. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Pauli Virtanen and 1 others. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

David Wadden, Shanchuan Lin, Kyle Lo, and 1 others. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Liang Wang, Jinyan Li, and Huan Liu. 2024. Social network and discourse: Graph neural networks for early conspiracy detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yue Zhang, Ming Li, and Chenhao Tan. 2024. Llm zero-shot capabilities for conspiracy theory narrative extraction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

A Implementation Details

We implement all pipelines in Python 3. We use pandas [McKinney, 2010] and NumPy [Harris et al., 2020] for data processing; spaCy [Honnibal et al., 2020] for linguistic analysis; textstat and lexicalrichness for readability and lexical diversity; VADER [Hutto and Gilbert, 2014] and NRCLex for sentiment and emotion; Gensim [Řehůřek and Sojka, 2010] and BERTopic [Grootendorst, 2022] for topic modeling; sentence-transformers [Reimers and Gurevych, 2019] for embeddings; UMAP [McInnes et al., 2018] and HDBSCAN [McInnes et al., 2017] for clustering; and scikit-learn [Pedregosa et al., 2011], SciPy [Virtanen et al., 2020], and SHAP [Lundberg and Lee, 2017] for modeling and analysis. We generate visualizations using matplotlib and seaborn.

B Random Forest Interpretation with TreeSHAP

We report global feature attribution for the Random Forest (RF) conspiracy classifier using TreeSHAP and compare it with univariate statistical analysis.

B.1 TreeSHAP vs. univariate t-tests

We contrast model-based attribution with marginal statistical association. While univariate tests highlight frequency-based signals (e.g., stopwords, adverbs), TreeSHAP assigns higher importance to ratio-based structural features. Both analyses consistently identify adverb-related signals as salient, indicating robust stylistic markers of conspiratorial discourse.

Interpretation. Linguistic categories like adverbs (`adv_ratio` and `n_advs_x`) appear at the top of both rankings, marking them as robust signals of conspiratorial discourse. TreeSHAP places higher importance on ratio-based structural features, while t-tests highlight specific frequency counts like `stopword_ratio`.

C Back-Translation Augmentation

We describe the augmentation pipeline and analyze its impact, including failure modes introduced by back-translation.

C.1 Pipeline

We apply per-sample augmentation in the RB-DET-AG series after splitting the training corpus (N=2,929 yes/no samples) from the held-out evaluation set. We evaluate two configurations that differ in the number of augmented variants per sample and the augmentation methods used:

- **RB-DET-AG2**$\times$ generates *one* augmented variant per original sample (target 2$\times$ expansion; total 5,858 samples). The augmentation loop assigns the first slot to contextual synonym substitution; thus, this configuration applies **contextual synonym substitution only**. We replace approximately 5% of tokens per sentence with contextually plausible alternatives predicted by `roberta-base` via `nlpaug` [Ma, 2019]. We invoke back-translation only as a fallback.

- **RB-DET-AG3**$\times$ generates *two* augmented variants per sample (target 3$\times$ expansion; total 8,787 samples). We alternate augmentation methods across slots, producing roughly equal proportions of (1) contextual synonym substitution (same `roberta-base` setup) and (2) **back-translation** via a German pivot using MarianMT [Junczys-Dowmunt et al., 2018]: `opus-mt-en-de` followed by `opus-mt-de-en`, decoded with default beam search (`num_beams=4`).

We accept an augmented sample only if it is non-empty and differs from the source, enforcing a minimal surface-form quality gate. We also vary fine-tuning across configurations: RB-DET-AG2$\times$ unfreezes the top 12 of 24 transformer layers (lr

#	Label	Original	Backtranslated
1	*yes*	Tom Delonge has accumulated high level government officials … puts it into a chronological timeline … It is either a major false flag or real UFO disclosure …	Tom Delonge has gathered high-ranking government officials … puts it in a chronological timeline … It's either a large false flag or real UFO revelation …
2	*no*	Sessions also indicated that the federal government would soon take steps to license more entities to legally grow marijuana …	Meetings also indicated that the federal government would soon take steps to license more facilities to legally grow marijuana …
3	*no*	… two volunteer medics … rescued people under fire from the most devastated kibbutzim and moshavim.	… two voluntary doctors … saving people under fire from the most devastated kibbutzim and Moshabim.
4	*no*	Joe has some excellent rants. He is hilarious … Look him up! … I love to listen to his rants …	Joe has some excellent ranzes. He is hilarious … Check him out! … I love to hear his ranzes …
5	*yes*	… aggressively attempting to crush competitive primaries. Pushing Progressives out of Key races. They didn't learn rigging the primaries in 2016 …	… aggressively trying to crush competition priorities. Pushing progressives from key races. They didn't learn rigging the priories in 2016 …

Table 7: Some random examples of EN→DE→EN back-translation. Highlighted spans show lexical and semantic distortions introduced by the pipeline.

= 5×10^{-5}, no weight decay, 20 epochs), while RB-DET-AG3× unfreezes the top 18 layers (lr = 2×10^{-6}, weight decay = 0.01, 25 epochs). We freeze embeddings in both cases and keep classification heads trainable.

C.2 Failure Analysis

We observe that neither augmented configuration consistently outperforms non-augmented contextual baselines (Table 4). In RB-DET-AG2×, which relies solely on contextual synonym substitution, augmented samples remain semantically close to the originals but tend to smooth out idiosyncratic lexical cues that encode conspiratorial signals. In RB-DET-AG3×, where half of the augmented data comes from back-translation, we observe additional failure modes (Table 7). For example, *"gun Mason compass"* becomes *"Cannon Mason Compass"* (row 10), altering semantically critical terms.

- **Lexical hallucination.** Back-translation introduces non-existent or semantically shifted tokens, particularly for colloquial or domain-specific expressions: *"rants"* → *"ranzes"* (row 4), *"moshavim"* → *"Moshabim"* (row 3). These substitutions distort the original vocabulary distribution.
- **Syntactic distortion.** Round-trip translation alters sentence structure and argument relations. For instance, *"Is there any method of persuasion … that you think can penetrate a community that far off the deep end?"* becomes *"Is there a method … that you think that a community that can penetrate far away from the deep end?"* (row 7), changing the underlying meaning.

These effects introduce *label-preserving but semantics-altering* noise into training. Since conspiratorial signals depend on specific phrasing, rhetorical patterns, and named entities, such distortions weaken the model's ability to learn reliable cues. The absence of a semantic consistency filter (e.g., SBERT similarity constraints) further amplifies this issue at higher augmentation levels. These observations motivate our use of **RB-DET-LoRA**, which improves performance by adapting model representations rather than relying on noisy synthetic data.

D Error Analysis and Boundary Behavior

As shown in Table 5, our extraction models consistently exhibit high recall but low precision, particularly for the *Actor* and *Victim* roles. The models correctly localize relevant text regions but systematically over-generate spans (1,172 predictions vs. 456 gold), resulting in a high false positive rate.

We follow the official CodaBench-compatible token-overlap evaluation from the starter pack. We first map character spans to token index sets using the organizer's regex-based tokenizer, then perform matching via token-level Intersection-over-Union (IoU). A prediction counts as correct only if it matches the gold span type and achieves IoU ≥ 0.5 under a one-to-one alignment (best unmatched prediction per gold span). This criterion requires substantial overlap rather than mere boundary contact.

This setup tolerates minor boundary shifts when

Evaluation Component	CodaBench/Starter-Pack Criterion
Tokenization	Text is tokenized with the regex `(\w+|[^\w\s])`, so words and punctuation are separate tokens.
Span representation	Each character span is converted to a set of covered token indices (a token is included if character intervals overlap).
Matching policy	One-to-one, type-constrained matching: for each gold span, select the best *unmatched* predicted span of the same marker type.
True Positive (TP) rule	A match is counted as TP only when token-level IoU between gold and predicted token sets is ≥ 0.5.
False Negatives (FN)	Gold spans left unmatched after the matching step are counted as FN.
False Positives (FP)	Predicted spans left unmatched after the matching step are counted as FP.
Per-role scores	Precision, Recall, and F1 are computed independently for each marker type (Action, Actor, Effect, Evidence, Victim).
Aggregate scores	Micro scores are computed from global TP/FP/FN counts; Ma-F1is the mean of per-type F1 values.
Practical implication	Minor boundary shifts may still match if IoU remains high; fragmented or over-generated spans often fail IoU ≥ 0.5, reducing precision.

Table 8: Concise summary of the token-IoU evaluation protocol used by the official CodaBench-compatible starter-pack script and our `infer_5_models_dev_f1.py` pipeline.

overlap remains sufficient, but fragmented or partial spans often fall below the IoU threshold and are penalized. We count unmatched predictions as false positives and unmatched gold spans as false negatives. Consequently, precision drops primarily due to over-prediction and span fragmentation, while recall reflects misses under the same IoU-constrained matching.

D.1 Evaluation Methods

We follow the official CodaBench token-overlap protocol. We map character spans to token sets and match them using IoU ≥ 0.5 under one-to-one, type-constrained alignment. This formulation prioritizes boundary precision: partial overlaps often fail to meet the IoU threshold, producing false positives and false negatives. Consequently, over-generation and span fragmentation drive precision loss in our models. Table 8 summarizes the protocol.

D.2 Handling of Overlapping Spans During Decoding

We adopt a type-separated decoding strategy. We predict each marker type independently and merge spans without cross-type suppression. This design preserves valid multi-role overlaps but increases false positives, especially under IoU-based evaluation. We plan to incorporate type-aware NMS and confidence calibration to better balance recall and precision.