YNU-HPCC at SemEval-2026 Task 10: Pretrained DistilBERT Models for Conspiracy Marker Extraction and Detection

Junpei Chen, You Zhang*, Jin Wang, Dan Xu and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
jpchen@stu.ynu.edu.cn, yzhang0202@ynu.edu.cn

Abstract

In this paper, we present our submission to the SemEval-2026 Psycholinguistic Conspiracy Shared Task (Task 10), which consists of two tasks: conspiracy marker extraction and conspiracy detection. For conspiracy marker extraction, we formulate the problem as a token classification task and fine-tune pretrained language models, achieving performance above the official baseline and ranking 6th on the final leaderboard. For conspiracy detection, we apply data preprocessing, regularization, and ensemble inference strategies, resulting in improvements over the baseline and a 10th-place ranking. Overall, our results demonstrate the effectiveness of pretrained language models for both tasks. Our code is publicly available at https://github.com/junpeiChen/code/tree/main/YNU-HPCC-at-SemEval-2026-Task10.

1 Introduction

We participate in SemEval-2026 Task 10: PsyCoMark (Samory et al., 2026), part of the Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026) (Ghosh et al., 2026). The task focuses on psycholinguistic conspiracy marker extraction and conspiracy detection, aiming to uncover how conspiracy theories are expressed in everyday conversations. Previous studies have shown that conspiracy-related discourse contains distinctive psychological and linguistic patterns (Fong et al., 2021; Giachanou et al., 2023; Gambini et al., 2023). Such discourse often includes psycholinguistic markers, such as Actors, Actions, Effects, Victims, and Evidence, which describe the roles, behaviors, consequences, targets, and supporting claims involved in conspiratorial narratives. This task is challenging because these markers may be implicit, context-dependent, overlapping, or nested. This fine-grained formulation is important because detecting whether a text is conspiratorial alone does not explain which linguistic cues contribute to the prediction.

Conspiracy-related text classification is more complex than ordinary sentiment analysis, as it often involves implicit psychological expressions and context-dependent meanings (Rosenthal et al., 2017; Zhang and Wang, 2020). Transformer-based models (Vaswani et al., 2017), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2020), have shown strong performance in text representation and classification. For example, the phrase "It's time to take action" may indicate a normal action in a supportive context, but may imply harmful or victimizing behavior in a negative or illegal context.

In this work, we initially explored CNN-based (Kim, 2014; Zhang et al., 2015) and LSTM-based (Wang et al., 2018, 2019) approaches, but their performance did not surpass the official baseline. We therefore adopted DistilBERT (Sanh et al., 2019) as the primary model due to its efficiency and suitability for resource-constrained training. To improve performance, we applied data preprocessing, data augmentation, R-Drop regularization (Liang et al., 2021), and ensemble-based inference. Experimental results show that our system achieved 6th place in conspiracy marker extraction and 10th place in conspiracy detection.

2 Methodology

2.1 Conspiracy Marker Extraction

In conspiracy marker extraction, there are three layers: Tokenizer Layer builds the mapping between labels and markers. The Training Layer trains the model for each marker separately and saves the trained models. The Inference Layer synthesizes the inference results based on the checkpoints of the saved models. The model il-

* Corresponding authors.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 741–747

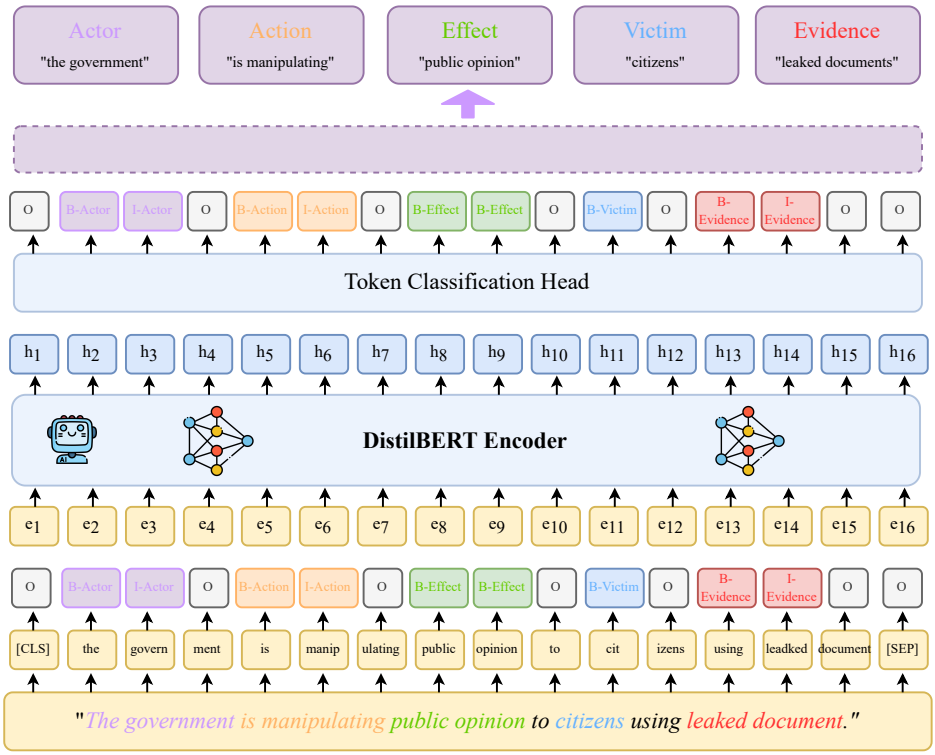

Figure 1: Model architecture of Conspiracy marker extraction.

lustration is shown in Figure 1.

2.1.1 Preprocess

Preprocessing removes unannotated data, which cannot provide correct labels and may interfere with model learning. For example, when loading the dataset, only annotated samples are used for training; unannotated samples are directly discarded.

To handle label incompatibility in overlapping and nested spans, our pipeline avoids forcing tokens into a single multi-class BIO schema. Instead, we train five independent binary classifiers—one for each marker type—so that cross-type overlaps (e.g., a token acting as both ”Actor” and ”Action”) are handled naturally. Within each classifier, we replace the traditional BIO scheme with a simplified IO scheme to focus on region detection. Nested or overlapping spans of the same marker type are flattened into contiguous spans, preserving context for region-level detection. This design avoids the incompatibility between overlapping annotations and a single multi-class BIO sequence.

2.1.2 Tokenizer Layer

A tokenizer splits text into a series of tokens, which are then used by the language model to generate responses. By using a tokenizer, a sentence is broken down into smaller subword units, facilitating subsequent processing of individual lexical units.

2.1.3 DistilBERT Layer

DistilBERT is a distilled version of BERT, retaining 95% of its performance while reducing model size by 60%. It has six Transformer layers with hidden size 768, enabling efficient inference. Trained via knowledge distillation, DistilBERT approximates BERT’s output distribution, maintaining strong performance on classification and sequence labeling tasks with fewer parameters and lower computational overhead.

2.1.4 Inference Layer

During inference, the model trained with DistilBERT and saved in the standard Hugging Face model format is loaded into inference mode. For the sentences to be inferred, multiple checkpoints are used to perform combined inference, and the final token predictions are aggregated. By first training with DistilBERT and then performing inference with the same DistilBERT model, the inference capability of the model can be maximized. In this way, the model can directly produce the required output files.

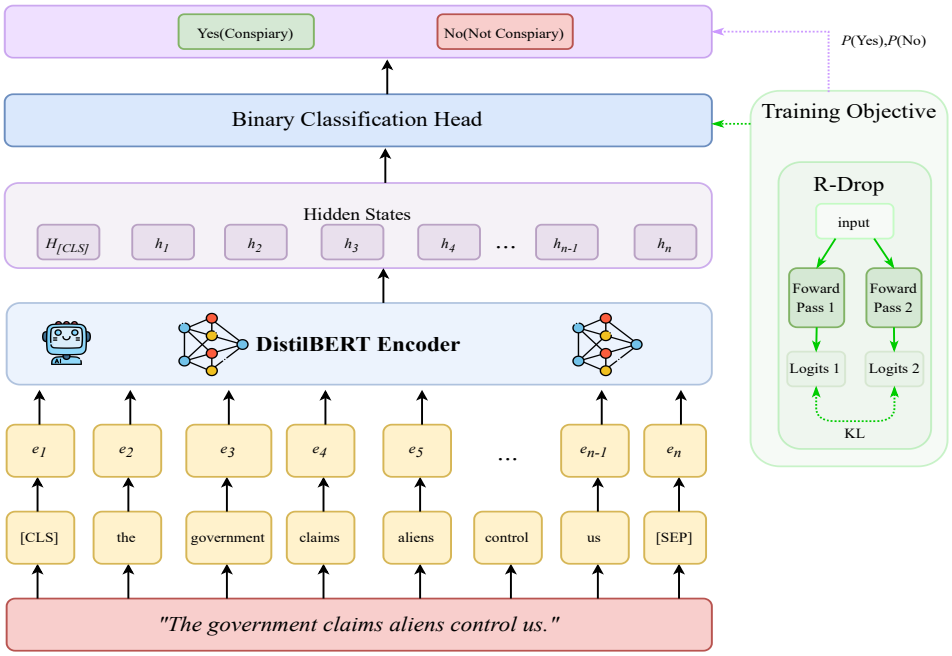

Figure 2: Model architecture of Conspiracy detection.

2.2 Conspiracy Detection

In conspiracy detection, the model consists of three layers: the Tokenizer Layer converts the input text into subword tokens and adds special tokens; the Training Layer fine-tunes DistilBERT for binary conspiracy classification; and the Inference Layer aggregates predictions from multiple checkpoints using soft voting. The model illustration is shown in Figure 2.

2.2.1 Preprocess

Preprocessing helps reduce bias, leading to fairer and more accurate results. The dataset contains a ”conspiracy” field, usually ”Yes” or ”No”. Entries with other values are removed, as inaccurate labels can negatively impact training. For example, only samples with ”Yes” or ”No” in the ”conspiracy” field are retained.

To augment the dataset and prevent the model from memorizing specific ”conspiracy trigger words,” we apply token-level random deletion (word dropout). For sequences longer than four words, a single token is randomly dropped with 20% probability, encouraging the model to rely on broader syntactic and semantic context and improving generalization.

2.2.2 Tokenizer Layer

DistilBERT uses the WordPiece tokenization algorithm (Wu et al., 2016), a subword segmentation method that breaks words into smaller fragments to cover a large vocabulary and handle out-of-vocabulary words. WordPiece is trained by iteratively merging character-level units, selecting the pair whose merging maximizes the likelihood increase ΔL on the training data:

$$\Delta L = \frac{\text{freq}(xy)}{\text{freq}(x) \cdot \text{freq}(y)} \quad (1)$$

where freq(xy) is the frequency of the merged symbol, and freq(x), freq(y) are the original frequencies. Higher ΔL indicates stronger co-occurrence. For example, “I love AI.” is tokenized into [“I”, “love”, “AI”, “.”], enabling better semantic understanding for downstream tasks.

2.2.3 DistilBERT Layer

DistilBERT is a distilled version of BERT, retaining 95% of its performance while reducing model size by 60%. It has six Transformer layers with hidden size 768, enabling efficient inference. Trained via knowledge distillation, DistilBERT approximates BERT’s output distribution, maintaining strong performance on classification and sequence labeling tasks with fewer parameters and lower computational overhead.

2.2.4 Inference

Before inference, training parameters are tuned with Optuna (Akiba et al., 2019) to find the optimal configuration, and the best model is saved.

Table 1: Experimental Results of Model Optimization for Conspiracy Marker Extraction

Type	Learning Rate	Batch Size	Epochs
Value	2e-06	16	10

Table 2: Experimental Results of Model Optimization for Conspiracy Detection

Type	Learning Rate	Batch Size	Epochs	Warmup Ratio	Weight Decay	Dropout Rate
Value	7.35e-06	32	9	0.05626	0.0815	0.4630

Table 3: Comparison of Methods for Conspiracy Marker Extraction

Method	**F1 Score**
Base Classification	0.15
Data Augmentation	0.17
Token Classification	**0.21**
RoBERTa-BIO	0.20
DeBERTa-BIO	0.20

During inference, the saved DistilBERT model is loaded, and predictions from multiple checkpoints are aggregated using soft voting to produce the final classification results.

3 Experimental Results

3.1 Dataset

The psycholinguistic conspiracy subtask in SemEval-2026 Task 10 aims to analyze conspiracy markers in texts and determine whether they contain conspiracy content. The conspiracy markers include Actor, Action, Effect, Victim, and Evidence. The dataset comprises over 4,800 annotated samples, covering more than 4,100 unique Reddit posts from over 190 subreddits. Approximately 4,000 comments contain at least one psycholinguistic marker annotation. The distribution of binary conspiracy labels in the training data is relatively balanced, with the numbers of "Yes" and "No" labels being roughly equal.

3.2 Evaluation Metrics

For conspiracy marker extraction, we use Macro F1 as the primary evaluation metric, which averages the F1 scores of all marker classes and treats each class equally regardless of its sample size.

For conspiracy detection, we use F1 score as the evaluation metric, as it balances precision and recall and is suitable for binary classification.

3.3 Methods Comparison

3.3.1 Conspiracy Marker Extraction

Base Classification: Standard training was performed with padding ignoring, class weights, warmup, and a maximum sequence length. During inference, Softmax and confidence filtering were applied.

Data Augmentation: Extends Base Classification by incorporating simple synonym replacement across five categories during training, while inference follows the same procedure as Base Classification.

Token Classification: Token-level classification was employed during training, with Softmax and confidence filtering applied during inference.

RoBERTa-BIO: Replaces the base model with RoBERTa-base and uses the BIO (Lample et al., 2016) tagging scheme for token classification. Confidence filtering is applied during inference.

DeBERTa-BIO: Replaces the base model with DeBERTa-v3-small and uses the BIO tagging scheme for token classification. Confidence filtering is applied during inference.

Finally, we selected the Token Classification method. The comparison of all methods is shown in Table 3. The final hyperparameters for conspiracy marker extraction were obtained through manual multiple-round tuning on the development set and are shown in Table 1.

3.3.2 Conspiracy Detection

CNN Baseline: The CNN model was used for training, with standard inference applied.

Weighted Loss + R-Drop: Data cleaning and augmentation were applied. During training, R-Drop, Label Smoothing, and a Weighted Loss formula were used. Standard inference was performed.

R-Drop Only: Retained data cleaning, data augmentation, and R-Drop, while removing Label

Table 4: Comparison of the Baseline and the Final Method for Conspiracy Marker Extraction

Method	Development Phase F1	Evaluation Phase F1
+WeightedLoss+smooth_predictions+threshold Distilbert-base	**0.21**	**0.21**
Distilbert-base (baseline)	0.15	0.15

Table 5: Comparison of the Baseline and the Final Method for Conspiracy Detection

Method	Development Phase F1	Evaluation Phase F1
+R-Drop+soft_voting+threshold Distilbert-base	**0.81**	**0.74**
Distilbert-base (baseline)	0.75	0.72

Table 6: Development-set comparison of methods for Conspiracy Detection.

Method	**F1 Score**
CNN Baseline	0.70
Weighted Loss + R-Drop	0.72
R-Drop Only	0.73
Softmax Thresholding	**0.74**
RoBERTa	0.72
DeBERTa	0.72
Augmented Data	0.71

Smoothing and Weighted Loss. Standard inference was performed.

Softmax Thresholding: Builds on previous best results, modifying inference by adding Softmax and setting a confidence threshold.

RoBERTa: Base model replaced with RoBERTa-base, training includes data cleaning and augmentation. Inference uses logit ensemble instead of Softmax.

DeBERTa: Base model replaced with DeBERTa-v3-small, training and inference procedures same as RoBERTa + Logit Ensemble.

Augmented Data: Training data augmented with ChatGPT-generated examples, retaining the inference procedure from the previous best method.

Finally, we selected the Softmax Thresholding method. The comparison of all methods is shown in Table 6. For conspiracy detection, we used Optuna (Akiba et al., 2019) for hyperparameter optimization, and the final configuration is shown in Table 2.

3.4 Comparative Results

For conspiracy marker extraction, the final F1 score reached 0.21, demonstrating the effectiveness of token classification and Weighted Loss for multi-class marker tasks. This improves over the baseline F1 of 0.15 by approximately 6 percentage points (Table 4).

For conspiracy detection, our best configuration achieved an F1 of 0.81 on the development set, dropping to 0.74 on the official evaluation set, indicating slight overfitting. Despite this, the approach still outperformed the official baseline by 2 percentage points (Table 5).

3.5 Discussion

For the multi-class task (conspiracy marker extraction), token classification improves the macro F1 score, and data cleaning and augmentation further enhance it.

For the binary classification task (conspiracy detection), DistilBERT outperforms RoBERTa and DeBERTa. This is not due to architectural superiority, but because DistilBERT's smaller parameter footprint allows training with optimal configurations (adequate batch size and R-Drop), while larger models had to use reduced batch sizes under limited GPU memory, leading to unstable gradients and overfitting.

4 Conclusion

In conspiracy marker extraction, we used the token classification method and fine-tuned the relevant parameters, effectively addressing labeling inaccuracies and improving the accuracy of conspiracy marker identification. The final macro F1 score reached 0.21, surpassing the baseline score of 0.15 and achieving an overall rank of 6.

In conspiracy detection, we applied the classification method with Optuna-based fine-tuning, which similarly helped mitigate labeling issues. The final F1 score was 0.74, exceeding the baseline score of 0.72 and achieving a final rank of 10. Future work will explore larger models and more effective methods to improve performance.

Acknowledgements

This work was supported by the Scientific Research Fund of Yunnan Provincial Education Department under Grant No. 2026J0006 and by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2019)*. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 4171–4186. Association for Computational Linguistics.

Adam Fong, Jon Roozenbeek, Daniel Goldwert, Steven Rathje, and Sander van der Linden. 2021. The language of conspiracy: A psychological analysis of speech used by conspiracy theorists. *Group Processes & Intergroup Relations*.

Marco Gambini, Stefano Tardelli, and Maurizio Tesconi. 2023. The anatomy of conspirators: Unveiling traits using a comprehensive Twitter dataset. *arXiv preprint arXiv:2308.15154*.

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. Detection of conspiracy propagators using psycho-linguistic characteristics. *Journal of Information Science*, 49(1):3–17.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 260–270. Association for Computational Linguistics.

Xiaobo Liang, Lijun Wu, Jiwei Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-Drop: Regularized dropout for neural networks. *arXiv preprint arXiv:2106.14448*.

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

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, pages 502–518. Association for Computational Linguistics.

Mattia Samory, Felix Soldner, and Veronika Batzdorfer. 2026. SemEval-2026 task 10: PsyCoMark – psycholinguistic conspiracy marker extraction and detection. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval 2026)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*.

Jin Wang, Bo Peng, and Xuejie Zhang. 2018. Using a stacked residual LSTM model for sentiment intensity prediction. *Neurocomputing*, 322:93–101.

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2019. Investigating dynamic routing in tree-structured LSTM for sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 3432–3437, Hong Kong, China. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, and Klaus Macherey. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Lin Zhang and Qi Wang. 2020. Affective text classification for psychological counseling. *arXiv preprint arXiv:2001.03818*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS 2015)*.