OoOoO at Touché: Argument-Structure-Aware RoBERTa Ensembles for Fallacy Detection

Touché at CLEF 2026

Ruilan Lin, Yusheng Yi*, Haoliang Qi, Zhenyang Cheng, Wenlong Liu and Zonghua Yang

Foshan University, Foshan, China

Abstract

This paper describes our system for Touché 2026 Task 1: Fallacy Detection. The task requires determining whether a given argument contains an informal fallacy and can be formulated as a binary classification problem over argumentative texts. We fine-tune RoBERTa-base under both base and enhanced input settings. To obtain robust validation estimates on the small training set, we use 5-fold cross-validation to produce out-of-fold predictions and tune the decision threshold by maximizing F1 on validation probabilities. To reduce variance caused by random seeds and input formats, we further use an ensemble method that averages the predicted fallacy probabilities from multiple models and re-tunes the decision threshold for the ensemble. Our experiments show that the base setting is mainly affected by false positives: many non-fallacious arguments that resemble fallacy patterns are predicted as fallacies. To mitigate this issue, we design an argument-first input format that places claims and supporting statements before the remaining context, improving precision in the base setting. The best base system, a 6-model mixed ensemble of original and argument-first inputs, achieves an out-of-fold F1 score of 0.7280. In the enhanced setting, a 3-seed RoBERTa-base ensemble achieves an out-of-fold F1 score of 0.8950. In the official Touché 2026 results, our enhanced run achieved a macro F1 score of 0.940 and ranked third in both the Task 1 enhanced ranking and the combined Task 1 ranking. The results suggest that explicit argument structure is important for fallacy detection.

Keywords

CLEF 2026, Touché, fallacy detection, RoBERTa, argument mining, ensemble learning, input formatting

1. Introduction

Informal fallacy detection is an important problem in computational argumentation and natural language processing. In online discussions, arguments are often expressed through natural-language claims, supporting evidence, context, and pragmatic assumptions rather than formal logical formulae [6]. Fallacy detection therefore requires models to capture the relation between a claim and its support, instead of relying only on surface lexical patterns [7]. Touché 2026 includes Fallacy Detection as one of its shared tasks on argumentation systems [1, 2]. In Sub-Task 1, systems must determine whether a given argument is fallacious, which we formulate as a binary classification problem [3].

The main challenge of this task is not simply detecting explicit fallacy keywords, but distinguishing genuine fallacies from arguments that are superficially similar to fallacies while still being valid. The official task description provides a `resembles_fallacy` field for non-fallacious examples to indicate which fallacy type the argument can be confused with. This design requires systems to separate genuine fallacies from superficially similar but valid arguments [3].

These observations also reveal a practical issue in this shared task: if a model relies only on the semantic representation of the whole text, it may over-predict fallacies for valid arguments that merely resemble fallacy patterns; if it relies on complex manual or structural analysis, the implementation and reproducibility cost may increase. Since CLEF working notes emphasize reproducible systems and the training set is relatively small, we need a framework that can be deployed offline, fine-tuned stably, and

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

2577877903@qq.com (R. Lin); yiys@fosu.edu.cn (Y. Yi); haoliangqi163@163.com (H. Qi); cheng_shock@foxmail.com (Z. Cheng)

0009-0005-1273-0136 (R. Lin); 0009-0006-7098-3681 (Y. Yi); 0000-0003-1321-5820 (H. Qi); 0009-0008-6262-5790 (Z. Cheng)

still use argument-structure information. RoBERTa-base provides a contextual Transformer encoder, while the task-provided argument fields and input-format design make it possible to emphasize the relation between claims and supports.

Based on these considerations, we build a RoBERTa-base system for fallacy detection. Instead of using a fixed threshold of 0.5, we search for the threshold that maximizes F1 on 5-fold out-of-fold (OOF) predictions. For the enhanced setting, we use the structured information in `text_enhanced` and `argument_enhanced` and apply probability ensembling across random seeds. For the base setting, our error analysis shows a large number of false positives, so we design an argument-first input format that places the claim and supports at the beginning of the input to reduce the influence of contextual noise and input truncation. Our main contributions are:

- We build a RoBERTa-base system for Touché 2026 Task 1 that combines 5-fold cross-validation, OOF threshold tuning, and probability ensembling.
- We compare base and enhanced settings and show that enhanced structured fields substantially improve fallacy detection performance.
- We analyze false positives in the base setting and show that argument-first input formatting partly mitigates the tendency to classify `resembles_fallacy` non-fallacious examples as fallacies.

2. Related Work

2.1. Informal Fallacy Detection

Informal fallacy detection focuses on erroneous reasoning patterns in natural-language arguments. Unlike formal logical errors, informal fallacies often depend on context, argumentative goals, evidence sufficiency, and semantic relations. Sahai et al. study common informal fallacies in online discussions and build a Reddit-based dataset; their results indicate that contextual information can help neural models classify fallacies [6]. This is consistent with the design of the Touché data, which provides fields such as `text_raw_parent`, `text_raw_title`, and `text_base`.

Jin et al. propose the Logical Fallacy Detection task and the Logic/LogicClimate datasets. They emphasize that fallacy detection requires understanding the underlying logical structure of an argument and report that a simple structure-aware classifier can outperform several pretrained language models [7]. This suggests that fallacy detection is not only topic classification, but also requires judging the relation between a claim and its support.

Yeh et al. introduce CoCoLoFa, a dataset containing 7,706 news comments and 648 news articles. Each comment is annotated for fallacy presence and fallacy type, and BERT-based fine-tuned models achieve strong performance on fallacy detection and classification [9]. This supports our choice of a fine-tuned encoder model as a reproducible baseline system.

2.2. Argument Structure and Enhanced Representations

A key issue in fallacy detection is whether the support actually justifies the claim. Toulmin's model decomposes arguments into components such as claim, data, warrant, backing, and rebuttal, providing a classical framework for analyzing natural-language argument structure [11]. In computational argumentation, Stab and Gurevych model argument components and argumentative relations in persuasive essays, showing the importance of explicitly identifying argumentative components and relations [12].

The Touché 2026 task provides `argument_base` and `argument_enhanced` fields, where argument fields contain claims and supports, making the argumentative structure explicit [3]. The task description also notes that Reddit arguments are often unclear and depend on background assumptions; thus, `text_base` incorporates relevant information from the parent comment and title into a self-contained rewrite, while enhanced fields further surface scheme and fallacy-related information [3].

Lei and Huang propose Logical Structure Tree, which organizes connectives, relations, and textual arguments into a tree structure and incorporates it into large language models through hard and soft

Table 1
Dataset statistics.

Split	Examples	Fallacy	Non-fallacy
Train	938	465	473
Test	233	hidden	hidden

prompting. Their experiments show improvements in precision and recall for fallacy detection and classification [10]. We do not construct a full logical structure tree. Instead, in the shared-task setting, we use the enhanced fields and an argument-first input format to place structured argumentative information in positions that are easier for the model to exploit. Different from these structure-heavy approaches, our system uses the task-provided argument fields and an argument-first input format to inject structural information into a lightweight RoBERTa-based pipeline.

3. Task and Dataset

Touché 2026 Fallacy Detection contains three related subtasks: Task 1 determines whether an argument is fallacious, Task 2 identifies the fallacy type, and Task 3 identifies the argument scheme for non-fallacious arguments. This paper focuses on Task 1 for two reasons. First, Task 1 is the basic detection task that precedes fine-grained fallacy type and argument scheme analysis. Second, given the short submission cycle and the offline training environment, we prioritize a reproducible and stable binary classification system. Task 2 and Task 3 can be further studied using multi-task learning on top of the present framework. The official evaluation uses a held-out test set and reports precision, recall, and F1-score [3].

The training set contains 938 examples, including 465 fallacious examples and 473 non-fallacious examples. The test set contains 233 examples with hidden labels. The training data are derived from the informal fallacies dataset of Sahai et al. and released in the `touchefallacy_2026` format [3, 6]. Table 1 summarizes the dataset statistics.

The fallacious training examples cover eight fallacy types: authority, population, natural, tradition, worse problems, black-or-white, hasty generalization, and slippery slope. Non-fallacious examples include a `resembles_fallacy` field indicating which fallacy type they most closely resemble. This field is important for analyzing hard negatives.

4. Method

4.1. System Overview

Our system consists of four parts: input construction, RoBERTa binary classification, OOF threshold tuning, and probability ensembling. Given an argumentative example, the system constructs an input text according to the base or enhanced setting, feeds it into a RoBERTa-base sequence classifier, and obtains $P(\text{fallacy})$. During validation, 5-fold cross-validation produces OOF probabilities for all training examples. During inference, we average probabilities from multiple models and apply the threshold tuned on OOF predictions.

4.2. RoBERTa Binary Classifier

RoBERTa improves BERT pretraining through several training choices, including dynamic masking, removing the NSP objective, and using larger batches and more data [8]. We use RoBERTa-base as the encoder and add a binary classification head. The model input is the concatenated argumentative text, and the output is a probability distribution over two labels. We use the probability of the fallacy class as $P(\text{fallacy})$. The training objective is binary cross-entropy, and labels are provided by the `fallacy_exists` field.

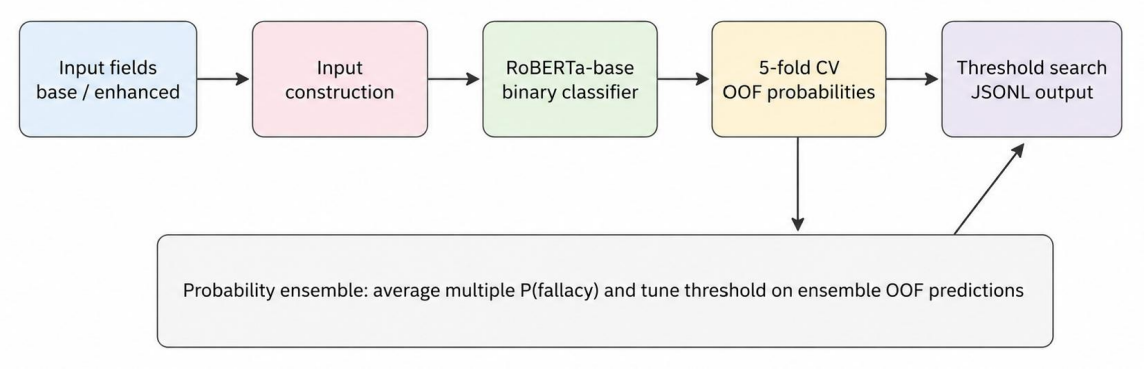

Figure 1: Overview of our fallacy detection pipeline.

We choose RoBERTa-base for two reasons. First, it can be fine-tuned stably on a small dataset. Second, its model size is suitable for reproducible experiments on a shared server without external network access. We do not use test labels for model selection.

4.3. Input Construction

In the enhanced setting, the model uses fields such as `text_enhanced` and `argument_enhanced`. These fields provide clearer argumentative expressions and help the model focus on the relation between claim and support. In the base setting, the original input contains the discussion title, parent comment, self-contained base text, and the argument structure in `argument_base`. Initial experiments show that the original base input has high recall but low precision, indicating that the model tends to over-predict the fallacy label.

To reduce false positives in the base setting, we design an argument-first input format. It places the claim and supports from `argument_base` at the beginning of the input and then appends `text_base`. The motivation is that fallacy detection mainly depends on the relation between claim and support rather than the raw title or parent comment. In addition, since long inputs may be truncated, placing the core argument structure earlier helps preserve key information.

4.4. Threshold Tuning and Probability Ensembling

Since the task is evaluated mainly by F1, a fixed threshold of 0.5 is not necessarily optimal. We search for a threshold on OOF predictions and select the one that maximizes F1. For ensemble models, we first average the $P(\text{fallacy})$ values from multiple models and then tune a threshold on the ensemble OOF probabilities.

Our ensemble method is probability averaging rather than majority voting. Majority voting only keeps the final labels and loses confidence information. Probability averaging preserves the strength of each model's prediction for the fallacy class and is better suited to complementary predictions from different random seeds and input formats.

4.5. Algorithm

Algorithm 1: Fallacy detection with OOF threshold tuning and probability ensembling

1. **Input:** training set D, test set T, and a set of model configurations M.
   Output: a JSONL prediction file for the test set.
2. Split D into five folds stratified by `fallacy_exists` and `resembles_fallacy`.
3. For each model configuration m, train five fold-specific models. Each time, use four folds for training and one fold for validation, and save validation-fold $P(\text{fallacy})$.

Table 2
Main experimental settings.

Item	Setting
Base model	RoBERTa-base
Task form	Binary sequence classification
Maximum length	512
Epochs	10
Batch size	16
Learning rate	2×10^{-5}
Random seeds	42, 2026, 3407
Validation	5-fold cross-validation
Threshold selection	OOF F1 maximization
Main metrics	Precision, Recall, F1

Table 3
Main out-of-fold validation results.

Setting	System	Precision	Recall	F1
Base	RoBERTa-base, original, best seed	0.5922	0.9118	0.7180
Base	Original 3-seed ensemble	0.6015	0.8860	0.7165
Base	Argument-first 3-seed ensemble	0.6529	0.8172	0.7259
Base	Original + argument-first 6-model ensemble	0.6031	0.9183	**0.7280**
Enhanced	RoBERTa-base, best seed	0.8745	0.9140	0.8938
Enhanced	Enhanced 3-seed ensemble	0.8747	0.9161	**0.8950**

4. Merge the five validation predictions to obtain OOF probabilities for model m. Also predict probabilities for the test set.
5. If ensembling is used, average the OOF probabilities across models and search for the F1-optimal threshold.
6. Average test probabilities across models, apply the threshold, and output `fallacy` or `non-fallacy` in the TIRA JSONL format.

5. Experiments and Results

5.1. Experimental Setup

Table 2 lists the main experimental settings. All models are trained in an offline local environment with locally cached RoBERTa-base weights. Test labels are not used for model selection.

5.2. Main Validation Results

Table 3 reports the main OOF validation results. The enhanced setting clearly outperforms the base setting. The enhanced 3-seed ensemble achieves an F1 score of 0.8950, while the best base system reaches 0.7280. This indicates that the enhanced fields provide useful structured argumentative information for fallacy detection.

In the base setting, the argument-first 3-seed ensemble improves precision from 0.6015 to 0.6529 compared with the original 3-seed ensemble, indicating that placing the argument structure earlier reduces false positives. The final base system uses a mixed 6-model ensemble of original and argument-first inputs and achieves the highest base F1 of 0.7280.

Table 4
Ablation on base input formats with seed 2026.

Input format	Threshold	Precision	Recall	F1
Original base	0.405	0.5922	0.9118	0.7180
Argument-first	0.445	0.6077	0.8860	**0.7209**
No-context	0.490	0.6090	0.8774	0.7189
Anti-false-positive prompt	0.525	0.6244	0.8366	0.7151
Argument-only	0.510	0.6214	0.8258	0.7091

Table 5
Official Touché 2026 Task 1 results for our submitted runs.

Ranking	Rank	Run	Tag	Precision	Recall	F1
Enhanced track	3	enhanced	enhanced	0.940	0.940	0.940
Base track	8	base-2	base	0.730	0.718	0.713
Base track	10	base	base	0.737	0.685	0.665
Combined Task 1	3	enhanced	enhanced	0.940	0.940	0.940

5.3. Ablation on Base Input Formats

The submissions were uploaded and evaluated through the TIRA platform [5]. Table 4 shows the base input-format ablation results using seed 2026. Argument-first is the best single-seed base variant. No-context is close to the original input, suggesting that the title and parent comment are not always beneficial. The anti-false-positive prompt obtains the highest precision but reduces recall substantially. Argument-only performs worst, indicating that claim and supports alone may lose useful context.

5.4. Official Touché Results

The official Touché 2026 results report macro-averaged precision, recall, and F1. Runs are ranked by macro F1, and the tag column indicates whether a system used only the original base fields or also the enhanced fields. Base and enhanced runs are ranked separately for each subtask, followed by a combined ranking [4].

Table 5 reports our official Task 1 results. Our enhanced run achieved a macro F1 score of 0.940 and ranked third in both the Task 1 enhanced ranking and the combined Task 1 ranking. Among our base submissions, `base-2` ranked eighth in the base track with a macro F1 score of 0.713, while `base` ranked tenth with a macro F1 score of 0.665.

6. Error Analysis

6.1. Overall Error Distribution

Table 6 compares the OOF confusion statistics of the base and enhanced ensembles. The base model has a severe false-positive problem. Among 473 non-fallacious training examples, 273 are incorrectly predicted as fallacies by the base 3-seed ensemble, corresponding to a false-positive rate of about 57.7%. This indicates that the base model tends to directly classify valid arguments that resemble certain fallacy types as fallacies.

The enhanced model reduces false positives from 273 to 61 while maintaining high recall. This suggests that enhanced fields help the model better distinguish genuine fallacies from similar non-fallacies.

Table 6
OOF error distribution.

Setting	TP	FP	FN	TN	Main error pattern
Base 3-seed ensemble	412	273	53	200	Many non-fallacies are predicted as fallacies
Enhanced 3-seed ensemble	426	61	39	412	Errors are fewer and more balanced

Table 7
Main errors of the base 3-seed ensemble by `resembles_fallacy`.

`resembles_fallacy`	Total	FP	FN	Error count	Error rate
slippery_slope	126	46	6	52	0.4127
natural	123	41	8	49	0.3984
authority	114	38	8	46	0.4035
hasty_generalization	116	35	7	42	0.3621
blackwhite	110	28	10	38	0.3455
tradition	125	34	3	37	0.2960
population	103	29	5	34	0.3301
worse_problems	121	22	6	28	0.2314

6.2. Errors by `resembles_fallacy`

Table 7 summarizes the main OOF errors of the base 3-seed ensemble by `resembles_fallacy`. False positives are concentrated in categories such as slippery slope, natural, authority, and hasty generalization. These categories require fine-grained judgments about evidence sufficiency, causal chains, and whether an authority or generalization is used appropriately.

These results explain why precision is low in the base setting. The model does not mainly fail by missing fallacies; rather, it lacks a clear decision boundary on hard negatives.

6.3. Enhanced Error Analysis

The enhanced 3-seed ensemble makes far fewer errors, with 61 false positives and 39 false negatives. When grouped by `resembles_fallacy`, the errors are more dispersed, although population, tradition, hasty generalization, worse problems, and natural examples remain somewhat difficult. Overall, the error analysis supports our main observation: the central challenge of fallacy detection is distinguishing genuine fallacies from superficially similar non-fallacies. Enhanced fields directly provide clearer argument structure and therefore substantially reduce false positives, while the base setting can only partially mitigate this issue through input formatting.

7. Conclusion

This paper presents our system for CLEF 2026 Touché Task 1: Fallacy Detection. We formulate the task as binary classification and fine-tune RoBERTa-base with 5-fold OOF validation, threshold tuning, and probability ensembling.

Our experiments show that the base setting is mainly limited by false positives: many non-fallacious examples that resemble fallacy patterns are predicted as fallacies. Placing claims and supporting statements at the beginning of the input improves precision, and the mixed ensemble of original and argument-first models achieves an OOF F1 score of 0.7280 in the base setting.

In the enhanced setting, structured fields lead to a clear performance improvement. The RoBERTa-base 3-seed enhanced ensemble achieves an OOF F1 score of 0.8950. In the official Touché 2026 results, our enhanced run achieved a macro F1 score of 0.940 and ranked third in both the Task 1 enhanced

ranking and the combined Task 1 ranking. Overall, explicit argument structure is a key factor for improving fallacy detection.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62276064). We thank the Touché 2026 organizers for providing the dataset, evaluation platform, and helpful feedback during the shared task.

Declaration on Generative AI

During the preparation of this work, the authors used generative AI tools for grammar and spelling checks, wording refinement, and research assistance. The authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Johannes Kiesel, Marc Feger, Tim Hagen, Sebastian Heineking, Maximilian Heinrich, Maik Fröbe, Katarina Boland, Christine Mathews, Wilhelm Pertsch, Julia Romberg, Ines Zelch, Stefan Dietze, Matthias Hagen, Martin Potthast, and Benno Stein. Overview of Touché 2026 — Argumentation Systems. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 17th International Conference of the CLEF Association (CLEF 2026)*, Springer, 2026.

[2] Johannes Kiesel, Marc Feger, Tim Hagen, Sebastian Heineking, Maximilian Heinrich, Maik Fröbe, Katarina Boland, Christine Mathews, Wilhelm Pertsch, Julia Romberg, Ines Zelch, Stefan Dietze, Matthias Hagen, Martin Potthast, and Benno Stein. Overview of Touché 2026 — Argumentation Systems — Extended Version. In *CLEF 2026 Working Notes*, CEUR-WS.org, 2026.

[3] Webis Group. Touché 2026 Fallacy Detection Task Website. https://touche.webis.de/clef26/touche26-web/fallacy-detection.html, 2026.

[4] Webis Group. Touché 2026 Fallacy Detection Results. https://touche.webis.de/clef26/touche26-web/fallacy-detection-results.html, 2026.

[5] Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In *Advances in Information Retrieval. ECIR 2023*, pages 236–241. Springer, 2023.

[6] Saumya Sahai, Oana Balalau, and Roxana Horincar. Breaking Down the Invisible Wall of Informal Fallacies in Online Discussions. In *Proceedings of ACL-IJCNLP 2021*, pages 644–657. Association for Computational Linguistics, 2021.

[7] Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schölkopf. Logical Fallacy Detection. In *Findings of EMNLP 2022*, pages 7180–7198. Association for Computational Linguistics, 2022.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

[9] Min-Hsuan Yeh, Ruyuan Wan, and Ting-Hao Kenneth Huang. CoCoLoFa: A Dataset of News Comments with Common Logical Fallacies Written by LLM-Assisted Crowds. In *Proceedings of EMNLP 2024*, pages 660–677. Association for Computational Linguistics, 2024.

[10] Yuanyuan Lei and Ruihong Huang. Boosting Logical Fallacy Reasoning in LLMs via Logical Structure Tree. In *Proceedings of EMNLP 2024*, pages 13157–13173. Association for Computational Linguistics, 2024.

[11] Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

[12] Christian Stab and Iryna Gurevych. Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics*, 43(3):619–659, 2017.