uzh_tt at Touché: Confidence-Adaptive Cascaded Fallacy Detection

Touché at CLEF 2026

Fatma-Zohra Rezkellah[1], Sophia Conrad[1]

[1]*Department of Computational Linguistics, University of Zurich (UZH), Zurich, Switzerland*

Abstract

We present a system for the Touché @ CLEF 2026 Fallacy Detection shared task, which comprises three subtasks: binary fallacy detection (Sub-Task 1), eight-class fallacy type classification (Sub-Task 2), and argument scheme classification (Sub-Task 3). Our approach combines a fine-tuned `roberta-large` classifier [1] with a confidence-adaptive two-tier cascade for Sub-Tasks 1 and 2. When RoBERTa's prediction confidence exceeds a learned threshold, its prediction is issued directly (Tier 1). This affects the majority of cases in the training set, thus limiting the need for expensive LLM inference. When confidence falls below the threshold, the argument is forwarded to GPT-5.4 for further reasoning (Tier 2). Sub-Tasks 1 and 2 are handled jointly if both fall below their respective thresholds. Then, a single GPT call first identifies the fallacy type and then determines whether the argument actually commits it, exploiting the theoretical dependency between detection and classification. For Sub-Task 3, the cascade is not applied: five-fold cross-validation and error analysis confirm that RoBERTa alone achieves near-ceiling performance, making LLM involvement unnecessary. Routing thresholds are learned from out-of-fold cross-validation predictions by sweeping candidate values and selecting those that maximise macro F1 under an oracle assumption for low-confidence cases. In five-fold stratified cross-validation on the training set, our RoBERTa baseline achieves macro F1 of 0.856 ± 0.026, 0.961 ± 0.015, 0.984 ± 0.010, and 0.928 ± 0.059 on Sub-Tasks 1, 2, 3 (goal), and 3 (basis), respectively. Reclassifying uncertain cases using GPT, average macro F1 scores across 2 trials stayed at 0.856 for Sub-Task 1 and increased to 0.971 for Sub-Task 2. On the held-out test set we achieved maximum F1 scores of 0.893, 0.991, and 0.858 on the three subtasks, ranking first on Sub-Task 2.

Keywords

fallacy detection, argument mining, confidence-based routing, large language models, cascade systems, RoBERTa

1. Introduction

Identifying fallacious arguments in natural language is a core challenge at the intersection of argumentation theory and natural language processing [2]. Unlike factual claims, which can be verified against external knowledge, fallacies are structural flaws in reasoning: the support offered does not legitimately justify the claim, even when it superficially appears to. This makes automatic fallacy detection substantially harder than surface-level misinformation classification, requiring models to assess the *quality* of an argument's inferential structure rather than merely its content [2, 3].

The Touché @ CLEF 2026 Fallacy Detection shared task [4, 5] poses this challenge in a particularly demanding form. The dataset [6], drawn from Reddit discussions, covers eight informal fallacy types grounded in pragma-dialectical theory. Crucially, every non-fallacious entry is selected precisely because it *structurally resembles* a fallacy: a legitimate appeal to a qualified expert looks like an appeal to authority; a well-founded generalisation looks like a hasty generalisation. Systems must therefore distinguish genuinely flawed reasoning from superficially similar but valid arguments, a boundary that prior work has found to challenge even strong pretrained language models [7, 8].

We address this challenge with a system built around two core observations. First, a fine-tuned RoBERTa classifier [1] is already highly accurate on clear-cut cases but produces systematically low-confidence predictions on genuinely ambiguous cases. Second, RoBERTa's prediction uncertainty is a *proxy for argument difficulty*: high confidence signals a case where direct prediction is reliable; low confidence signals a boundary case where deliberate LLM-based reasoning adds substantial value. This

Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2026), September 21–24, 2026, Jena, Germany

fatma-zohra.rezkellah@uzh.ch (F. Rezkellah); sophia.conrad@uzh.ch (S. Conrad)

motivates a **confidence-adaptive two-tier cascade**. In **Tier 1**, high-confidence predictions are issued directly from RoBERTa with no LLM involvement. In **Tier 2**, the argument is forwarded to GPT-5.4 [9], which reasons over the case with structured prompts.

A further contribution concerns the joint treatment of Sub-Tasks 1 and 2. For cases where RoBERTa falls below the threshold on both sub-tasks, a single GPT call first classifies the fallacy type (Sub-Task 2) and then determines whether the argument actually commits it (Sub-Task 1). This reflects the theoretical structure of the problem: detecting a fallacy and identifying its type are two aspects of the same inferential assessment, and resolving them jointly avoids error propagation across independent pipelines.

For Sub-Task 3 (argument scheme classification), the cascade is deliberately not applied. Five-fold cross-validation results show that RoBERTa achieves macro F1 of 0.984 ± 0.010 on `argument_goal` and 0.928 ± 0.059 on `argument_basis`, with nearly all predictions made at confidence above 0.99.

Our contributions are as follows:

- A fine-tuned `roberta-large` baseline for all three sub-tasks using structured claim–support inputs from `argument_enhanced`, validated by five-fold stratified cross-validation.
- A confidence-adaptive two-tier cascade for Sub-Tasks 1 and 2 in which high-confidence cases are resolved by RoBERTa alone, while uncertain cases are forwarded to GPT-5.4.
- A learned routing threshold derived from out-of-fold cross-validation predictions, calibrated to maximise macro F1 under an oracle assumption for low-confidence cases.
- A joint Sub-Task 1 and Sub-Task 2 treatment in which a single GPT call handles both classification and detection for doubly uncertain cases.

2. Related Work

2.1. Fallacy Detection: From Early Corpora to Unified Benchmarks

The automatic detection of argumentative fallacies in natural language has a relatively short but rapidly evolving history in NLP. Early efforts were constrained by the scarcity of annotated data and the diversity of fallacy taxonomies across rhetoric, logic, and argumentation theory, making cross-study comparisons difficult. A turning point came with Sahai et al. [6], who constructed the first large-scale Reddit corpus of informal fallacies grounded in pragma-dialectical theory, covering the eight types that form the backbone of the present shared task. Their neural baselines established that conversational context is critical: models with access to the parent comment systematically outperform those operating on isolated utterances, motivating our use of the `argument_enhanced` field of the data. Jin et al. [2] introduced LOGIC and LOGICClimate and showed that standard pretrained language models perform poorly without structure-aware supervision, making fallacy detection a genuine reasoning benchmark rather than a surface-level classification task.

Subsequent work has widened both scope and evaluation rigour. Helwe et al. [7] unified several prior corpora into the MAFALDA benchmark and provided the first systematic comparison of state-of-the-art models and human performance, finding that even the strongest systems lag behind human annotators on fine-grained types. Hong et al. [10] found that LLMs struggle to self-verify fallacious reasoning chains using a corpus of 232 reasoning-step fallacy types. Xu et al. [11] combined logic-programming oracles with LLM generation to produce naturally expressed fallacious statements with controlled type labels, enabling rigorous probing of LLM reasoning at scale. This progression from single-domain datasets to automatically generated benchmarks underscores the sustained difficulty of the task.

2.2. LLMs as Fallacy Classifiers

Systematic studies of LLM fallacy classification abilities have found mixed results. Pan et al. [8] evaluate GPT-4, GPT-3.5, LLaMA, Qwen, and Mistral and find that zero-shot LLMs can approach but rarely exceed

fine-tuned T5 baselines on in-domain data, while multi-round strategies with explicit type definitions provide the largest gains. Jeong et al. [12] enrich prompts with counterarguments, goal-aware context, and explanations, obtaining large F1 gains across five domains. Alhindi et al. [13] demonstrate that multitask instruction prompting generalises better across label sets, while Alhindi et al. [14] show that LLMs can generate synthetic training examples for rare fallacy classes.

Beyond prompt engineering, several works combine language models with explicit structural components. Sourati et al. [3] retrieve similar fallacy cases to enrich model inputs with goals, explanations, and counterarguments. Wang et al. [15] decompose classification into binary procedural questions drawn from a relational fallacy graph. Lalwani et al. [16] translate arguments into first-order logic and apply SMT solvers to decide validity, substantially outperforming end-to-end neural approaches. Collectively, these results show that explicit logical or structural reasoning consistently outperforms purely neural methods.

2.3. Argumentation Theory and Argument Schemes

The theoretical foundations for Sub-Task 3 lie in formal argumentation theory. Walton et al. [17] provide the foundational compendium of 96 argumentation schemes, stereotypical patterns of defeasible reasoning each associated with critical questions. Walton and Macagno [18] extend this into a systematic classification grouping schemes by shared high-level properties. Macagno [19] operationalises these dimensions empirically, instantiating `argument_goal` (practical vs. epistemic) and `argument_basis` (internal vs. external) as orthogonal axes for classifying argumentative moves in political discourse, the exact framework adopted by the shared task. Jo et al. [20] connect this tradition to NLP by demonstrating that explicit logical mechanisms can classify argumentative relations without direct supervision on relation labels.

3. Task and Data

The Touché @ CLEF 2026 Fallacy Detection shared task comprises three sub-tasks over a dataset of arguments extracted from Reddit discussions [6]. Sub-Task 1 is binary: determining whether an argument is `fallacious` or `non-fallacious`. Sub-Task 2 assumes the argument is fallacious and requires identifying which of eight types it commits: `authority`, `black-white`, `hasty_generalization`, `natural`, `population`, `slippery_slope`, `tradition`, or `worse_problems`. Sub-Task 3 assumes the argument is non-fallacious and classifies it along two orthogonal dimensions from Macagno's framework [19]: `argument_goal` (practical vs. epistemic) and `argument_basis` (internal vs. external), yielding four combined labels.

Each entry provides the original Reddit comment (`text_raw`), a self-contained context-aware rewrite (`text_base`), and an enhanced rewrite surfacing fallacy and scheme information (`text_enhanced`). Structured `{claim, supports}` representations are provided for both variants. A critical design property is the `resembles_fallacy` field: every non-fallacious entry is selected because it structurally resembles a specific fallacy type, ensuring systems cannot rely on surface heuristics to separate the classes in Sub-Task 1. The training set contains 938 entries (465 fallacious, 473 non-fallacious) and the held-out test set contains 233 entries. We use the enhanced fields (`tag = enhanced`) in all experiments described below.

4. System Description

We investigate two system configurations. First, we establish a RoBERTa-based baseline for all three sub-tasks. Second, we extend this baseline for Sub-Tasks 1 and 2 with a confidence-adaptive cascade in which uncertain RoBERTa predictions are forwarded to GPT-5.4 for re-classification. Sub-Task 3 is handled exclusively by the RoBERTa baseline, as the model already achieves near-ceiling performance without LLM involvement.

4.1. RoBERTa Baseline

We format each argument as a structured string:

`Claim: <claim> Support: <support_1> <support_2> ...`

where `claim` and `supports` are extracted from `argument_enhanced`.

We then fine-tune `roberta-large` Liu et al. [1] independently for each sub-task using the HuggingFace Transformers Trainer.

For Sub-Task 1, RoBERTa is trained as a binary classifier for fallacy detection. For Sub-Task 2, the model is trained only on entries labeled as fallacious in the training set (465 instances) to predict the fallacy type. For Sub-Task 3, we train two independent binary classifiers for `argument_goal` and `argument_basis`, both using only non-fallacious entries from the training set (473 instances), with predictions combined at inference time.

Optimisation uses AdamW with learning rate 2×10^{-5}, batch size 8, weight decay 0.01, and a linear warmup schedule with warmup ratio 0.1. Training runs for up to 10 epochs with early stopping based on validation macro F1, retaining the best checkpoint.

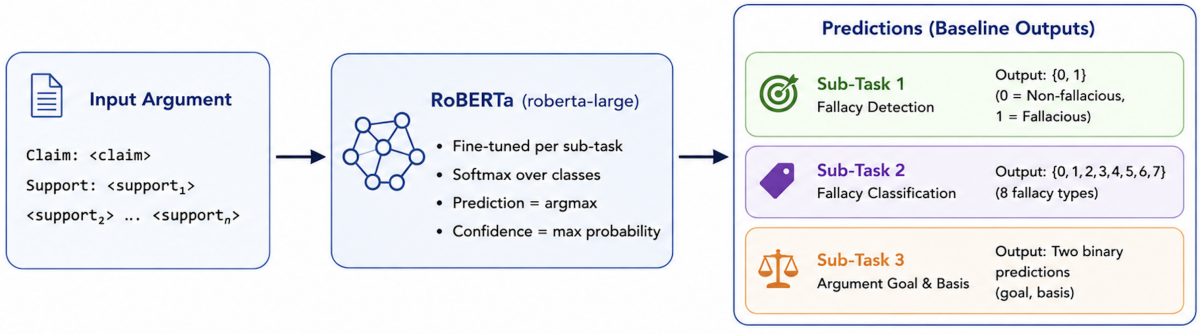

Figure 1: Overview of the standalone RoBERTa baseline used across all three shared-task sub-tasks. Separate task-specific classifiers are trained for fallacy detection, fallacy type classification, and argument scheme prediction.

4.2. Confidence-Adaptive Cascade (CAC)

Figure 2 provides an overview of the proposed confidence-adaptive cascade.

4.2.1. Threshold Learning

To determine when to trust RoBERTa directly and when to invoke the LLM, we learn a routing threshold from out-of-fold predictions generated during five-fold stratified cross-validation.

By using out-of-fold predictions, where each entry is always predicted by a model that never trained on it, the confidence scores reflect genuine model uncertainty on unseen arguments.

We initially considered learning a separate threshold per fallacy type, motivated by the observation that some types such as `slippery_slope` are highly distinctive while others such as `population` are more often confused. However, with only approximately 55 examples per class after train–validation splitting, per-class threshold estimates would be too noisy to be statistically reliable. We therefore use a single global threshold per sub-task.

For each candidate threshold $\tau \in [0.50, 0.99]$ in steps of 0.01, we simulate the cascade outcome on the out-of-fold predictions: entries with confidence $\geq \tau$ retain their RoBERTa predictions, while entries with confidence $< \tau$ are assumed to be corrected by the LLM (oracle assumption). The threshold maximising macro F1 under this simulation is selected. This oracle assumption is optimistic, it represents an upper bound on cascade performance, but it provides a principled, data-driven way to identify the optimal operating point before running any LLM calls.

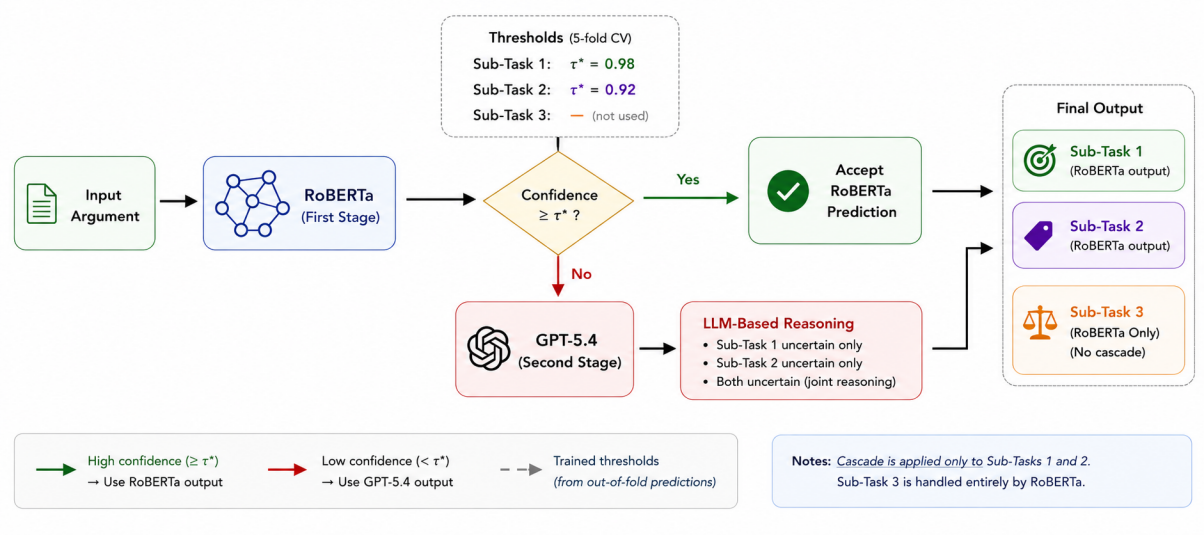

Figure 2: Overview of the confidence-adaptive cascade for Sub-Tasks 1 and 2. RoBERTa serves as the first-stage classifier. High-confidence predictions are accepted directly, while low-confidence cases are forwarded to GPT-5.4 for additional reasoning. Sub-Task 3 does not use the cascade.

The analysis reveals a clear signal: RoBERTa's accuracy on high-confidence entries (confidence $\geq \tau^*$) is substantially higher than on low-confidence entries, confirming that uncertainty predicts error. For Sub-Task 1, the learned threshold is $\tau^* = 0.98$, routing 19.4% of entries to the LLM; RoBERTa accuracy on high-confidence entries is 0.899 versus 0.667 on uncertain ones. For Sub-Task 2, $\tau^* = 0.92$, routing only 11% of entries; accuracy is 0.975 on high-confidence entries versus 0.765 on uncertain ones. For Sub-Task 3, the model is confidently correct on virtually all entries, and no entries are identified as benefiting from LLM involvement; consequently, the cascade is not applied.

4.2.2. LLM-Based Reasoning

For entries below the routing threshold, we call GPT-5.4 [9] via the OpenAI API. For arguments forwarded to the LLM, we extract `claim`, `supports`, and `text_enhanced` for context, embedded directly in the prompt.

Sub-Task 1 (fallacy detection) and Sub-Task 2 (fallacy classification) are handled jointly or independently depending on which sub-task falls below threshold:

Sub-Task 1 uncertain only. GPT is prompted to determine whether the argument commits a fallacy. The prompt explicitly provides the `resembles_fallacy` annotation and the corresponding fallacy definition, then asks GPT to reason through whether the argument's support fails to justify its claim in the specific way that definition requires. This targets the core challenge of Sub-Task 1: the argument looks like a fallacy, but does it actually commit one? The full prompt is shown in A.1.

Sub-Task 2 uncertain only. GPT is prompted to classify the fallacy type given the argument and all eight fallacy definitions. The Sub-Task 1 prediction from RoBERTa is passed as context. The full prompt is shown in A.2.

Both uncertain. A single GPT call handles both sub-tasks: GPT first identifies which fallacy type the argument most closely resembles (Sub-Task 2), then determines whether the argument actually commits that fallacy (Sub-Task 1). This joint treatment exploits the theoretical dependency between detection and classification: classifying the type first provides a grounded basis for the binary detection decision.

All prompts use structured JSON output via the OpenAI structured outputs API, ensuring valid label predictions.

4.3. Sub-Task 3: RoBERTa Only

Sub-Task 3 is handled entirely by the RoBERTa baseline without cascade involvement. Cross-validation yields macro F1 of 0.984 ± 0.010 on `argument_goal` and 0.928 ± 0.059 on `argument_basis`, with the model operating at very high confidence on virtually all entries.

Given this, invoking the LLM for Sub-Task 3 would incur API cost without meaningful accuracy gain.

5. Experiments

5.1. Experimental Setup

All RoBERTa models are initialised from `roberta-large` and fine-tuned on a single NVIDIA A100 GPU with the hyperparameters described in Section 4.1. For reporting, we use five-fold stratified cross-validation on the 938 training entries, which serves two purposes: obtaining reliable performance estimates unaffected by a single train–validation split, and generating out-of-fold predictions for threshold learning (Section 4.2.1).

5.2. Baseline Results

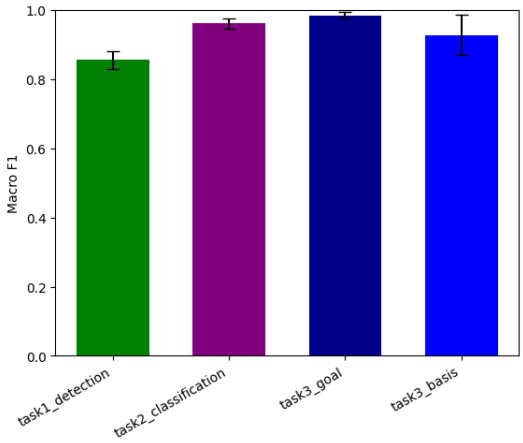

Figure 3: Cross-validation performance of the RoBERTa baseline across all sub-tasks of the shared task. Bars show mean macro F1 across five stratified folds, while error bars indicate standard deviation.

Cross-Validation Results Figure 3 reports five-fold cross-validation macro F1 mean and standard deviation. Results are strong across all sub-tasks. Sub-Task 2 achieves the highest performance (0.961 ± 0.015), with `slippery_slope` perfectly classified across all folds (F1 = 1.00 ± 0.00) and `population` being the most challenging type (F1 = 0.910 ± 0.034). Sub-Task 1 is the hardest task as expected, given that every non-fallacious entry structurally resembles a fallacy type (0.856 ± 0.026). Sub-Task 3 goal classification is nearly perfect (0.984 ± 0.010), while basis classification shows higher variance (0.928 ± 0.059) driven by the minority `external` class.

Per-Class Analysis Table 1 reports per-class F1 for Sub-Task 2 across the five cross-validation folds. `slippery_slope` is the most reliably classified type (F1 = 1.00±0.00), confirming that the cascade threshold is high for cases where RoBERTa is consistently confident. `population` and `hasty_generalization` show the highest variance, consistent with their structural similarity to each other and to `authority`.

Fallacy Type	F1 Mean	F1 Std
authority	0.957	0.040
black-white	0.981	0.023
hasty_generalization	0.931	0.055
natural	0.975	0.034
population	0.910	0.034
slippery_slope	1.000	0.000
tradition	0.968	0.039
worse_problems	0.970	0.027

Table 1
Per-class macro F1 (mean ± std) for Sub-Task 2.

5.3. CAC Results

Threshold Analysis and Routing Statistics Table 2 reports the routing thresholds learned from out-of-fold predictions and the resulting accuracy split between high- and low-confidence cases. The gap between high- and low-confidence accuracy validates the uncertainty-as-difficulty-proxy hypothesis: RoBERTa is substantially less accurate precisely on the cases it is uncertain about.

Sub-Task	τ^*	High-conf acc	Low-conf acc
Sub-Task 1	0.98	0.899	0.667
Sub-Task 2	0.92	0.975	0.765

Table 2
Routing thresholds and accuracy split. High-conf: entries with confidence $\geq \tau^*$; Low-conf: entries below threshold forwarded to GPT.

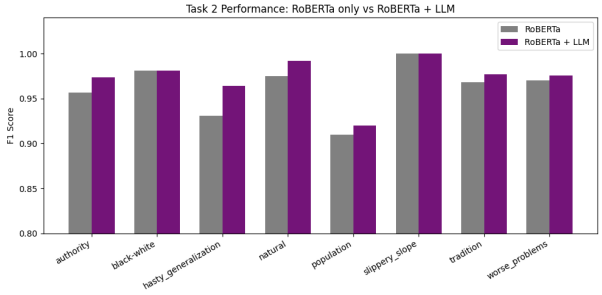

Figure 4: Per-class F1 scores for Sub-Task 2 on the training set using the cascaded approach (purple) compared to the RoBERTa-only approach (grey).

LLMs Results Forwarding the difficult (RoBERTa confidence below threshold) cases to GPT-5.4 resulted in the same Macro F1 score (0.856) for Sub-Task 1, but slight gains in performance on Sub-Task 2 on the training set. Every class benefit from being re-classified by GPT-5.4 as shown in Fig. 4, with one exception – the `black-white` class remained at 0.981 Macro F1. The biggest increase in performance is seen in `hasty_generalisation`, from 0.931 to 0.964 Macro F1.

Why the Oracle Assumption Does Not Fully Hold In Practice Qualitative inspection of , specifically, the low-confidence Sub-Task 2 cases reveals two systematic confusion patterns that explain why GPT-5.4 cannot fully close the gap between the oracle upper bound and actual performance.

First, three fallacy types `authority`, `population`, and `tradition`, share a common surface structure: all appeal to an external collective (expert figures, crowd size, or historical practice) as justification. Arguments that mix these signals simultaneously confuse both RoBERTa and GPT. For instance, the

argument *"A billion people across every era have testified to the reality of God; the speaker, a legally trained sceptic, studied the matter and found it convincing"* is labelled `population` (the flaw is the crowd-size appeal) but RoBERTa predicts `authority` (fixating on the speaker's credentials), and GPT faces the same ambiguity because both readings are locally coherent. Similarly, *"Literally everyone held those views 200 years ago; they reflected the natural order of things"* is labelled `tradition` but predicted as `authority`, as references to historical figures and universal past consensus are difficult to disentangle without author intent.

More interestingly, some cases require identifying the *primary* flaw among several simultaneously present. The argument *"Personal experience shows the worst police were bored officers in small rural towns"* (true label: `hasty_generalization`) is predicted as `worse_problems` because the argument pivots to a reform proposal, but the underlying flaw is drawing a sweeping conclusion about police brutality from a single anecdote. GPT, like RoBERTa, tends to classify the most salient surface feature rather than the deepest structural flaw.

These observations suggest that further gains on Sub-Task 2 require prompts that explicitly ask the model to identify the *primary* reasoning flaw rather than any present flaw, and that provide contrastive definitions for the easily confused triple of `authority`, `population`, and `tradition`.

Table 3
Official test set results on TIRA (macro F1). Sub-Task 3 was only done using RoBERTa, therefore there is no value for the Cascade Model on that task. Our best systems ranked eleventh on Sub-Task 1, first on Sub-Task 2 and third on Sub-Task 3.

Sub-Task	Baseline (RoBERTa)	Cascade
Sub-Task 1 — Fallacy Detection	0.878	0.893
Sub-Task 2 — Fallacy Classification	0.689	0.991
Sub-Task 3 — Argument Goal and Basis	0.858	N/A

6. Conclusion

We have presented a confidence-adaptive two-tier cascade for the Touché @ CLEF 2026 Fallacy Detection shared task. The core insight, that a fine-tuned RoBERTa classifier's prediction confidence is a reliable proxy for argument difficulty, allows reasoning compute to be allocated adaptively: direct classification for confident cases and GPT-based reasoning only for genuinely ambiguous boundary cases. We would like to highlight RoBERTa's performance on the task, which was particularly strong for certain classes in Sub-Task 2 (i.e. perfect classification on the slippery_slope class in the training set) despite rather few (ca. 55) examples per class, eliminating the need to include computationally expensive LLMs. For the same reason of saving resources, we abstain from fine-tuning an LLM on the task.

Future work will investigate whether incorporating structured multi-step reasoning such as Tree of Thoughts [21] within the LLM tier further improves performance on the hardest boundary cases, and whether the per-class confidence profile can be used more directly: for instance, by weighting LLM call rates differently across fallacy types as more training data becomes available.

Acknowledgments

This work was supported by the DIZH-supported DSI PostDoc Project *AI-R* at the University of Zurich (UZH), as well as by the Swiss National Science Foundation (SNSF) under Grant No. 218836[1].

[1] https://data.snf.ch/grants/grant/218836

Generative AI Statement

During the preparation of this manuscript, we used Claude by Anthropic to assist with refining the language and improving the clarity and readability of the text. All scientific content was developed, verified, and approved by us, and we take full responsibility for the content of the paper.

References

[1] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: https://arxiv.org/abs/1907.11692.

[2] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schölkopf, Logical fallacy detection, in: Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 7180–7198. URL: https://aclanthology.org/2022.findings-emnlp.532. doi:10.18653/v1/2022.findings-emnlp.532.

[3] Z. Sourati, V. Yadav, D. Deshpande, H. Rawlani, F. Ilievski, H.-A. Sandlin, A. Mermoud, Case-based reasoning with language models for classification of logical fallacies, CoRR abs/2301.11879 (2023). URL: https://arxiv.org/abs/2301.11879.

[4] J. Kiesel, M. Feger, T. Hagen, S. Heineking, M. Heinrich, M. Fröbe, K. Boland, C. Mathews, W. Pertsch, J. Romberg, I. Zelch, S. Dietze, M. Hagen, M. Potthast, B. Stein, Overview of touché 2026 — argumentation systems, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. S. Salido, A. Barrón-Cedeño, A. G. S. Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. 17th International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2026.

[5] J. Kiesel, M. Feger, T. Hagen, S. Heineking, M. Heinrich, M. Fröbe, K. Boland, C. Mathews, W. Pertsch, J. Romberg, I. Zelch, S. Dietze, M. Hagen, M. Potthast, B. Stein, Overview of touché 2026 — argumentation systems — extended version, in: E. S. Salido, A. Barrón-Cedeño, A. G. S. Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[6] S. Sahai, O. Balalau, R. Horincar, Breaking down the invisible wall of informal fallacies in online discussions, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 644–657. URL: https://aclanthology.org/2021.acl-long.53. doi:10.18653/v1/2021.acl-long.53.

[7] C. Helwe, T. Calamai, P. Paris, C. Clavel, F. M. Suchanek, MAFALDA: A benchmark and comprehensive study of fallacy detection and classification, CoRR abs/2311.09761 (2023). URL: https://arxiv.org/abs/2311.09761.

[8] F. Pan, X. Wu, Z. Li, A. T. Luu, Are LLMs good zero-shot fallacy classifiers?, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024), Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 14256–14272.

[9] A. Singh, A. Fry, A. Perelman, A. Tart, A. Ganesh, A. El-Kishky, A. McLaughlin, A. Low, A. Ostrow, A. Ananthram, et al., Openai gpt-5 system card, CoRR abs/2601.03267 (2025). URL: arXiv:2601.03267.

[10] R. Hong, H. Zhang, X. Pang, D. Yu, C. Zhang, A closer look at the self-verification abilities of large language models in logical reasoning, CoRR abs/2311.07954 (2023). URL: https://arxiv.org/abs/2311.07954.

[11] Z. Xu, J. Ding, Y. Lou, K. Zhang, D. Gong, Y. Li, Socrates or smartypants: Testing logic reasoning capabilities of large language models with logic programming-based test oracles, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, 2026, pp. 19433–19440.

[12] J. Jeong, H. Jang, H. Park, Large language models are better logical fallacy reasoners with

counterargument, explanation, and goal-aware prompt formulation, in: Findings of the Association for Computational Linguistics: NAACL 2025, 2025, pp. 6918–6937.
[13] T. Alhindi, T. Chakrabarty, E. Musi, S. Muresan, Multitask instruction-based prompting for fallacy recognition, CoRR abs/2301.09992 (2023). URL: https://arxiv.org/abs/2301.09992.
[14] T. Alhindi, S. Muresan, P. Nakov, Large language models are few-shot training example generators: A case study in fallacy recognition, CoRR abs/2311.09552 (2023). URL: https://arxiv.org/abs/2311.09552.
[15] O. P. Wang, T. Bansal, R. Bai, E. M. Chui, L. Gilpin, Follow my lead: Logical fallacy classification with knowledge-augmented LLMs, CoRR abs/2510.09970 (2025). URL: https://arxiv.org/abs/2510.09970.
[16] A. Lalwani, T. Kim, L. Chopra, C. Hahn, C. Trippel, Z. Jin, M. Sachan, NL2FOL: translating natural language to first-order logic for logical fallacy detection, CoRR abs/2404.12177 (2024). URL: https://arxiv.org/abs/2404.12177.
[17] D. Walton, C. Reed, F. Macagno, Argumentation Schemes, Cambridge University Press, Cambridge, UK, 2008. doi:10.1017/CBO9780511802034.
[18] D. Walton, F. Macagno, A classification system for argumentation schemes, Argument & Computation 6 (2015) 219–245.
[19] F. Macagno, Argumentation profiles and the manipulation of common ground: The arguments of populist leaders on Twitter, Journal of Pragmatics 191 (2022) 67–82. doi:10.1016/j.pragma.2022.01.022.
[20] Y. Jo, S. Bang, C. Reed, E. Hovy, Classifying argumentative relations using logical mechanisms and argumentation schemes, Transactions of the Association for Computational Linguistics 9 (2021) 721–739.
[21] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models, in: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), 2023.

A. Appendix

A.1. Prompt Task 1

You are an expert in logical fallacy detection. Analyze the following argument and determine if it contains a logical fallacy. A logical fallacy is a mistake in reasoning that makes an argument invalid or unreliable, even if it appears persuasive.
This is the argument: claim
This is how the argument is supported: supports
Here is some more context to the argument: text
Respond with valid JSON containing the label.

A.2. Prompt Task 2

You are an expert in logical fallacy analysis. Analyze the following argument in two steps:

STEP 1: Classify which fallacy type this argument most closely resembles: 1. authority - Wrong expert cited (source isn't qualified). Example: "This diet works, a celebrity said so"
2. blackwhite - Only 2 options given (hides other possibilities). Example: "You're either with us or against us"
3. hasty_generalization - Few cases ⟶ big rule (too little evidence). Example: "I met 2 rude French people, so all French people are rude" 4. natural - Natural = good (confuses natural with safe). Example: "This herbal remedy has no chemicals, so it's safer"
5. population - Everyone believes it (popularity ≠ truth). Example: "Millions believe it, so there must be something to it"
6. slippery_slope - X leads to catastrophe (unjustified chain reaction). Example: "If we

allow X, eventually society will collapse”
7. tradition - We’ve always done it (old = correct). Example: ”We shouldnt change, families always had men as breadwinners”
8. worse_problems - Bigger issues exist (dismisses valid concern). Example: ”Why care about straws when forests are burning?”

STEP 2: Based on that classification, determine if this is actually a logical fallacy or valid reasoning that just resembles that pattern. Remember:
- Citing a qualified expert in their field is NOT an authority fallacy
- A legitimate binary choice is NOT a black-white
- Sufficient evidence for a generalization is NOT hasty generalization
This is the argument: claim
This is how the argument is supported: supports
Here is some more context to the argument: text
Respond with valid JSON containing fallacy_type and is_fallacy.