helium at Touché: The Case for Naturalistic Evaluation of Fallacy Detection

Touché at CLEF 2026

Diana Markova[1]

[1]*Sofia University, Bulgaria*

Abstract

Manipulative discourse need not be limited to false factual claims – arguments can change beliefs through implicit common-ground manipulation, fallacious reasoning, and emotive framing even when their surface factual claims are not false. We study Touché 2026 Subtask 1, binary fallacy detection, as a case study in naturalistic evaluation. We compare lexical baselines, embedding classifiers, fine-tuned transformer encoders, and instruction-tuned language models across naturalistic and enhanced inputs. Our results show that enhanced rewrites can substantially improve performance when both training and test inputs use the same label-aware enhanced-rewrite format, though these gains weaken when evaluation inputs switch back to base or raw text. In particular, a RoBERTa-large checkpoint trained on enhanced text loses much of its performance when evaluated on base text (Δ macro-F1 = -0.32), indicating reliance on input cues rather than robust fallacy recognition. This interpretation is also supported by the high performance of TF-IDF classifiers on enhanced inputs. By contrast, the general-purpose language model tested shows more consistent performance across prompting variants and datasets. We argue that realistic assessment of fallacy-detection capabilities should be wary of format overadaptation and style confounding.

Keywords

fallacy detection, Touché 2026, format overadaptation, style confounding, out-of-distribution evaluation

1. Introduction

Persuasion does not require false premises. Arguments can shift beliefs through implicit common-ground manipulation, fallacious reasoning, and emotive framing even when their surface claims are not factually wrong [1]. Detecting such informal fallacies automatically is thus distinct from standard fact-checking: the relevant question is not only whether the stated facts are true, but also whether the argument uses weak or misleading reasoning. This makes fallacy detection relevant for analysing online discourse and debate, especially in settings where persuasive text may be generated or amplified by language-model systems.

Touché 2026 frames this problem as a shared-task setting for evaluating these capabilities [2, 3], with evaluation hosted on the TIRA platform [4]. In this working note we focus on Subtask 1, binary fallacy detection over arguments drawn from Reddit discussions [5]: given an argument, decide whether it commits an informal fallacy.

The Subtask 1 data provides each argument in several parallel forms: a *raw* Reddit comment with optional conversational context, a normalised *base* rewrite, and an *enhanced* rewrite. The enhanced rewrite is produced with knowledge of the gold label. This makes it useful as a diagnostic condition, but it also means that it may contain cues that are not present in natural text. The main question this work asks is as follows: how much of the measured performance comes from detecting fallacious reasoning, and how much comes from the input representation?

We study this question by comparing four families of systems: lexical TF-IDF baselines, dense-embedding classifiers, fine-tuned transformer encoders, and an instruction-tuned language model. We evaluate them across base, raw, and enhanced input conditions, and across in-domain Touché validation, a public test proxy, and LOGIC-binary as an out-of-distribution (OOD) stress test. Our main

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

EMAIL: dianamarkovakn@gmail.com (D. Markova)

findings are:

- Enhanced rewrites raise in-domain scores for every model family, but the gain appears to be mostly tied to the enhanced format: a fixed encoder loses much of its advantage when evaluated on base inputs, and a TF-IDF classifier also performs well on enhanced text.
- Unlike Gemma 4 31B, the supervised lexical and encoder systems show weak OOD transfer on LOGIC-binary, often near or below chance.

We therefore consider base and raw inputs as the primary conditions for evaluating naturalistic fallacy detection, and treat enhanced-format results as diagnostics rather than as the main evidence of model ability. We base our submission choice on this generalization concern.

2. Task and Data

Touché 2026 Subtask 1 is binary fallacy detection: given an argument drawn from Reddit discussions, the task is to predict whether it commits an informal fallacy (`fallacy_exists` $\in \{0, 1\}$). The released data contains three text representations for each instance. Table 1 summarizes the class balance for the Touché set, the Sahai-aligned test proxy, and the LOGIC-binary diagnostic.

2.1. Input Fields

- **raw** (`text_raw`): the original Reddit comment, with optional thread title and parent-comment fields (`text_raw_title`, `text_raw_parent`);
- **base** (`text_base`): a normalised rewrite of the argument, including the relevant title/parent context;
- **enhanced** (`text_enhanced`): a label-aware rewrite of the argument, produced with access to the gold label.

Experiments are run under three corresponding input view conditions: *raw+ctx*, *base*, and *enhanced*. Here *raw+ctx* means concatenating `text_raw`, `text_raw_title`, and `text_raw_parent`; *base* uses the normalised `text_base` field, which already incorporates relevant context from the title and parent; and *enhanced* uses the label-aware `text_enhanced` field.

2.2. LOGIC-binary Transfer Diagnostic

For an additional transfer check, we use the LOGIC-binary diagnostic benchmark, constructed from the public data files released with the NL2FOL work of Lalwani et al. [6], following their LOGIC+SNLI construction. None of the models for the Touché task are trained, tuned, or selected on LOGIC-binary.

The fallacy side comes from the LOGIC corpus of Jin et al. [7], a 13-way collection of fallacious arguments. Since LOGIC does not provide genuine non-fallacious argumentative examples, the binary setup pairs 2,449 LOGIC fallacy examples with 2,449 valid entailment examples from the SNLI (Stanford Natural Language Inference) test split [8]. These SNLI examples are formed from premise and hypothesis sentences whose source sentences are Flickr image captions. The resulting diagnostic set is exactly balanced (2,449 fallacy and 2,449 valid examples).

While these negatives are valid entailment examples, they are not natural non-fallacious arguments. It is worth noting that this creates a strong style difference between the two classes: the negative examples are caption-based entailments, while the fallacious examples are longer argumentative texts. In our experiments, a bag-of-words model reaches macro-F1 near 0.98 in-domain on this setup, so LOGIC-binary itself is not a naturalistic benchmark of fallacy detection.

Nonetheless, this diagnostic is still useful for the limited purpose of an OOD evaluation-time generalization test.

2.3. Internal Split and Public-Test Proxy

The Touché Subtask 1 labelled training data contains 938 instances. For model development, we split this data into 738 training instances and a 200-instance held-out validation set. This split is used for hyperparameter, checkpoint, prompt, and configuration selection.

The official Touché public-test labels were withheld during our work on this task. For additional reporting, we reconstruct a proxy label set for the subset of public-test rows whose raw text can be aligned to the earlier Sahai et al. dataset [5]. This is not the official shared-task evaluation set – it covers only the aligned subset and uses labels recovered from the earlier source, while the official Touché labels remain hidden. These proxy labels are **not** used for training, hyperparameter tuning, checkpoint selection, prompt selection, or model selection. Results on this subset are reported only as an auxiliary diagnostic, and do not represent official leaderboard performance.

Table 1
Class balance by dataset and evaluation role. F denotes fallacious examples; NF denotes non-fallacious or valid examples. "–" means that the dataset is not used for that role.

Dataset	Train		Validation		Test/diagnostic	
	F	NF	F	NF	F	NF
Touché Subtask 1	365	373	100	100	–	–
Sahai-aligned public-test proxy	–	–	–	–	94	101
LOGIC-binary diagnostic	–	–	–	–	2449	2449

3. Models

We compare four families of systems, each evaluated under the base, *raw+ctx*, and enhanced input conditions defined above. The *raw+ctx* condition is the explicit-context raw Reddit view: the target comment is provided together with the thread title and parent comment when those fields are available.

Lexical baselines. Word unigram and bigram TF-IDF features (`ngram_range` $= (1, 2)$, `min_df` $= 2$, sublinear term frequency) paired with a linear support-vector classifier (LinearSVC) and with logistic regression.

Dense-embedding classifiers. OpenAI `text-embedding-3-large` embeddings [9] with a logistic-regression head.

Fine-tuned encoders. ELECTRA-small [10] and RoBERTa-large [11]. Base checkpoints, per-run hyperparameters, and hardware are listed in Appendix G (Table 7). We use the Hugging Face sequence-classification heads – the final-layer representation of the classification token (`[CLS]` for ELECTRA and `<s>` for RoBERTa) is passed through the model's classification head to produce two logits, with binary probabilities obtained by softmax.

We train or fine-tune the models listed above on the 738-instance Touché Subtask 1 training split under each input view condition.

Instruction-tuned language model. Gemma 4 31B (`gemma-4-31b-it`) [12], accessed through the Google GenAI API as a zero-shot freeform classifier, together with a four-shot variant (two non-fallacious and two fallacious examples). The model was asked to end with a JSON object, which we parsed for the binary label. Decoding and API settings are given in Appendix F.1.

4. Evaluation

We use macro-F1 as the primary reported score. Where useful, we also report fallacy-class F1, AUROC for scored binary systems, and class-specific recall.

For in-domain evaluation, all supervised systems are trained on the 738-instance Touché Subtask 1 training split and evaluated on the 200-instance held-out Touché validation split. For the public-test proxy, we evaluate the same selected systems on the Sahai-aligned subset described above; these labels are not used for model or prompt selection.

For OOD reporting, we evaluate the Touché-trained systems directly on the LOGIC-binary diagnostic set without additional training, tuning, or threshold selection. We report the transfer gap as

$$\Delta_{\text{OOD}} = \text{MacroF1}_{\text{LOGIC-binary}} - \text{MacroF1}_{\text{Touché val}}.$$

Negative values hence mean that a model performs worse on LOGIC-binary than on the Touché validation split. Because the LOGIC-binary valid class consists of the dataset's caption-based entailments rather than natural non-fallacious arguments, this OOD score should be read as an evaluation-time stress test of format and corpus transfer rather than as a naturalistic estimate of fallacy-detection ability on a balanced external benchmark.

We also report class-specific recall: for a class c, recall is the fraction of gold instances of class c that are predicted as c ($\text{TP}_c/(\text{TP}_c + \text{FN}_c)$). This distinguishes fallacy recovery from models that mainly recognise the caption-based entailment valid class. Appendix A, Appendix B, and Appendix C report these recalls for Touché validation, LOGIC-binary, and the Sahai-aligned test proxy, respectively.

5. Results

Figure 1 shows the fixed-checkpoint input control for the enhanced-trained RoBERTa-large model. Figure 4 later summarises Subtask 1 performance across input conditions, model families, and evaluation sets.

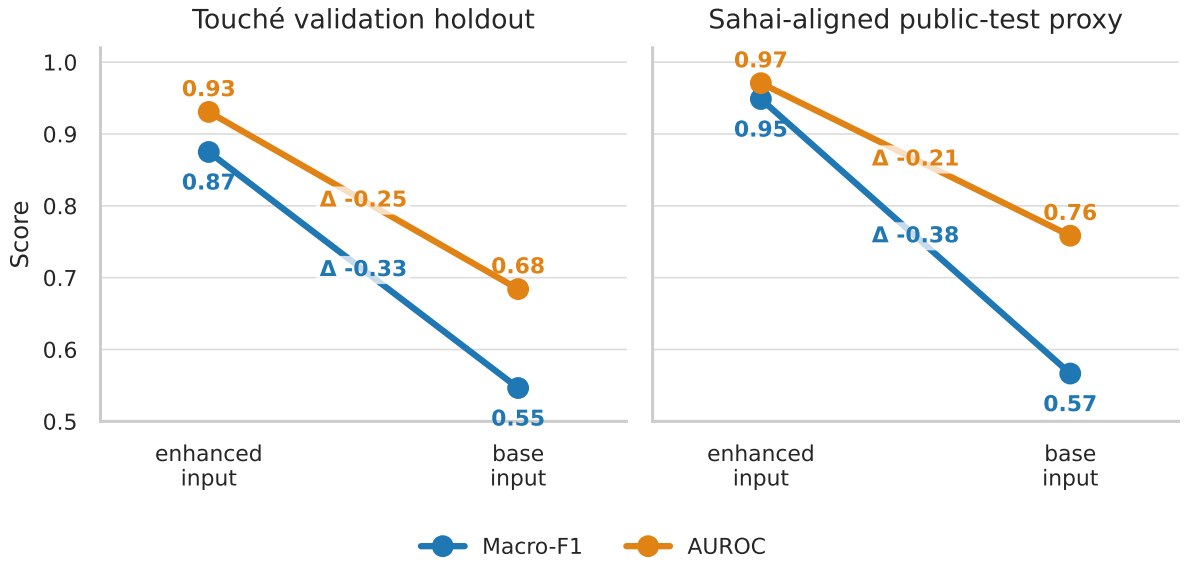

Figure 1: Input-control result for the final RoBERTa-large checkpoint trained on enhanced text. The same checkpoint is evaluated twice: first with the label-aware enhanced input view and then with the base input view. Lines report macro-F1 and AUROC on our Touché validation split and on the Sahai-aligned public-test proxy. The drop under base inputs shows that the enhanced view gain is tied to the enhanced rewrite format rather than to a representation-independent fallacy detector.

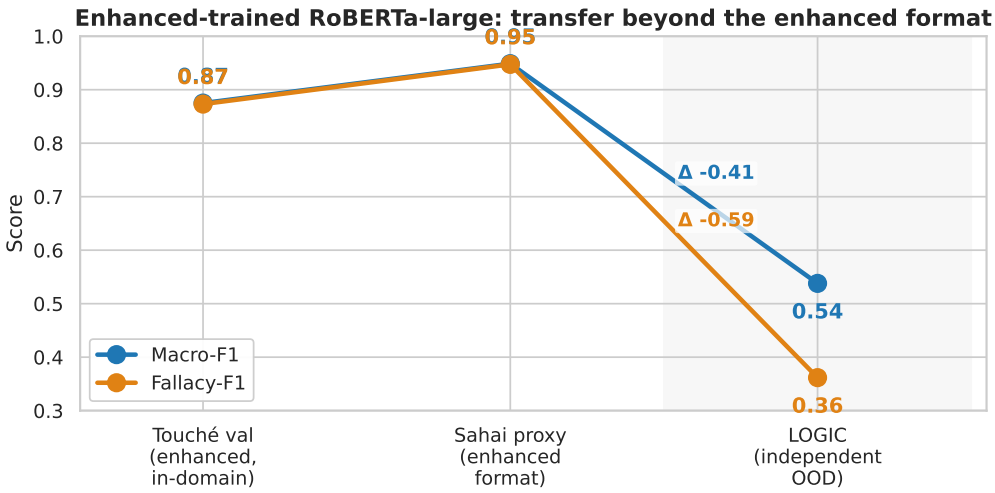

Figure 2: Transfer of the enhanced-trained RoBERTa-large checkpoint. The high Sahai-proxy score is again reported on the enhanced view; the LOGIC-binary drop shows that the enhanced view score is format-bound. The LOGIC-binary set is itself style-confounded and used only as an independent-corpus stress test.

5.1. Enhanced Training Inputs

Moving from base to enhanced inputs for training increases in-domain Touché validation macro-F1 for every model family. For example, RoBERTa-large rises from 0.65 to 0.87, ELECTRA-small from 0.66 to 0.84, and TF-IDF+SVC from 0.61 to 0.83. Holding the enhanced RoBERTa checkpoint fixed and feeding it base inputs removes most of the advantage (Figure 1), and the same checkpoint obtains much lower scores on an independent corpus (Figure 2).

To be precise, the cross-view effect is asymmetric for the encoders: feeding enhanced inputs to base-trained checkpoints slightly improves validation macro-F1 (RoBERTa 0.65 → 0.69; ELECTRA 0.66 → 0.70), whereas feeding base inputs to enhanced-trained checkpoints causes much larger drops (RoBERTa 0.87 → 0.55; ELECTRA 0.84 → 0.57). This suggests that enhanced training induces dependence on rewrite-specific cues. Appendix D (Table 6) reports the full base/enhanced train-view × eval-view crossing for every scored system. The enhanced-on-enhanced condition is the peak in each case and drops sharply under enhanced-on-base inputs, while base-trained models do *not* show such collapse.

To test whether this enhanced-input signal is concentrated in a small set of surface lexical cues, we ran a diagnostic TF-IDF logistic-regression model on the same Subtask 1 train/validation split and then restricted the model to the highest-absolute-coefficient n-grams selected on the training split. With the enhanced view, the top 100 n-grams recover most of the full TF-IDF score (macro-F1 0.785 vs. 0.840); with raw or base views, the same restriction remains near the corresponding weaker full vocabulary baseline. The strongest enhanced cues include fallacy-template n-grams such as *problems*, *nature*, *there are*, and *proves*, and non-fallacy rewrite markers such as *reasoning*, *actually*, *actual*, and *consequences*. Figure 3 visualizes the corresponding coefficient-ranked terms across raw+ctx, base, and enhanced inputs.

An auxiliary attention probe on validation rows containing such cue spans points the same way. More concretely, the enhanced RoBERTa checkpoint assigns elevated classification-token attention to these spans (mean cue-attention lift 1.32 on fallacy rows and 1.68 on non-fallacy rows; Appendix E), though note that this is diagnostic rather than causal.

Table 2
TF-IDF logistic-regression diagnostic for label-correlated n-grams. "Top" models are retrained using only the highest-absolute-coefficient features selected from the training split. The reported scores are on the Touché validation holdout split.

Input view	Vocab. size	Full vocab.	Top 25	Top 100
raw+ctx	8,629	0.609	0.530	0.585
base	6,344	0.615	0.593	0.549
enhanced	11,158	0.840	0.689	0.785

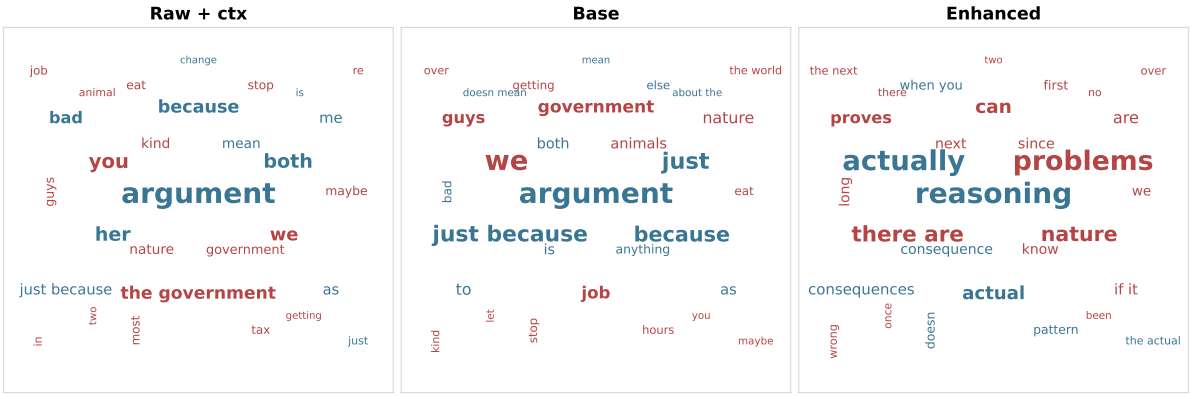

Figure 3
Coefficient-weighted n-gram clouds for the TF-IDF leakage diagnostic. Font size scales with the absolute logistic-regression coefficient. Red terms push toward the fallacy class; blue terms push toward the non-fallacy class.

5.2. Out-of-Distribution Generalization

The supervised lexical and encoder models transfer poorly to LOGIC-binary. Relative to Touché validation, the largest macro-F1 gaps are −0.42 for ELECTRA-small base and −0.40 for RoBERTa-large base. Enhanced training does not mitigate this gap for the lexical models: TF-IDF+SVC drops by −0.07 in the base condition but by −0.29 in the enhanced condition. Gemma 4 31B is the only positive-gap exception, reaching macro-F1 ≈ 0.93 on LOGIC-binary.

These gaps should not be read as a fair transfer ranking. Several systems collapse toward one class on LOGIC-binary (Appendix B, Table 4). Enhanced encoders score better than their base/raw+ctx counterparts by Δ_{OOD}, but this is likely achieved by recovering the caption-based valid class: enhanced RoBERTa has valid/fallacy recall 0.99/0.22, and enhanced ELECTRA has 1.00/0.24. Base and raw+ctx encoder models show the opposite skew, recovering LOGIC fallacies above chance while nearly failing on the caption-based valid class. We treat LOGIC-binary as a stress test for format and corpus transfer rather than as a naturalistic external benchmark.

The Subtask 1 submission uses the base-track Gemma 4 31B *raw+ctx* configuration, with `text_raw` plus the explicit parent comment and submission title. The *base* view configuration scores higher on the Sahai-aligned public-test proxy (macro-F1 0.80 vs. 0.73), but *raw+ctx* scores higher on the held-out Touché validation split (macro-F1 0.77 vs. 0.73). Since model selection is based on the validation split, the proxy result is reported only as an auxiliary diagnostic and is not used as a submission benchmark.

On the official Task 1 base-track leaderboard, this submitted configuration received precision 0.745, recall 0.737, and F1 0.736, ranking fourth among ranked base-track submissions.

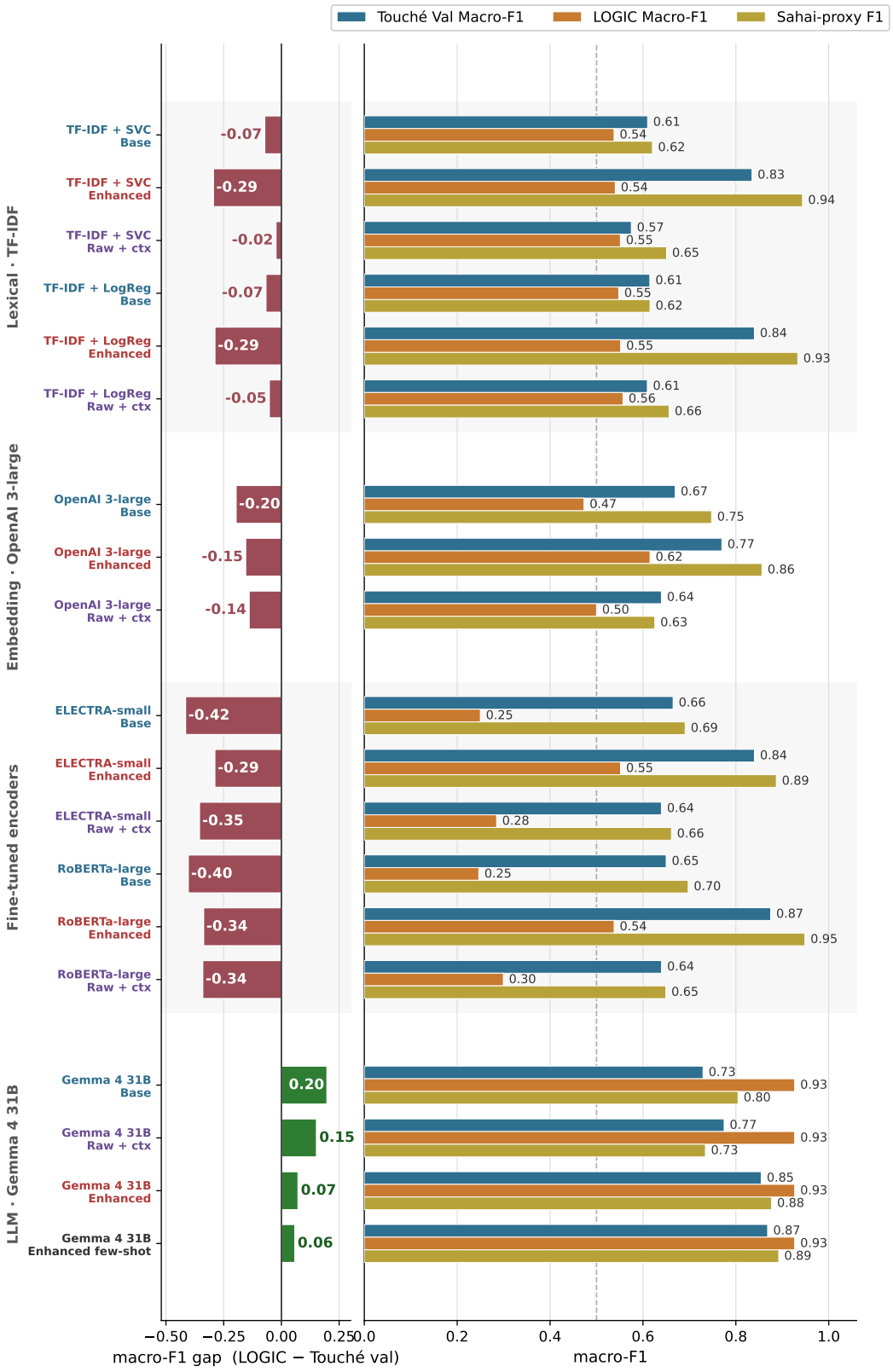

Figure 4: Subtask 1 macro-F1 on Touché validation, LOGIC-binary, and the Sahai-aligned proxy. Negative gaps indicate lower LOGIC-binary than Touché validation macro-F1.

6. Conclusion

In this working notes paper, we evaluated Touché 2026 Subtask 1 under several input conditions and model types. The main pattern is that enhanced rewrites lead to higher in-domain scores across model types, but these scores are sensitive to the input representation. For example, the best-performing checkpoint on the Touché validation split – RoBERTa-large, trained on enhanced inputs – loses much of its performance when evaluated on base text. A similar drop appears when the checkpoint is evaluated on an independent corpus.

For future work, we plan to extend the study to the remaining Touché subtasks and to run further cross- and mixed-benchmark generalization analyses. We also plan to test whether fine-tuning or otherwise adapting language models on the task data improves performance beyond zero-shot and few-shot prompting, especially under base and raw input conditions. White-box analysis and activation probing on language models for detecting and identifying fallacious reasoning and rhetoric are further research directions we intend to pursue.

Declaration on Generative AI

The author was assisted by coding agents in the execution of this work. AI assistants were also used to help polish the text. The author reviewed and edited the generated content as needed and takes full responsibility for the publication's content.

References

[1] 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.

[2] 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. https://touche.webis.de/clef26/touche26-web/index.html.

[3] 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. https://touche.webis.de/clef26/touche26-web/index.html.

[4] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous integration for reproducible shared tasks with tira.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.

[5] 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 (ACL-IJCNLP), 2021, pp. 644–657. doi:10.18653/v1/2021.acl-long.53.

[6] A. Lalwani, T. Kim, L. Chopra, C. Hahn, Z. Jin, M. Sachan, Autoformalizing natural language to first-order logic: A case study in logical fallacy detection, in: Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific

Chapter of the Association for Computational Linguistics, 2025, pp. 132–147. doi:10.18653/v1/2025.findings-ijcnlp.8.

[7] 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, 2022, pp. 7180–7198. doi:10.18653/v1/2022.findings-emnlp.532.

[8] S. R. Bowman, G. Angeli, C. Potts, C. D. Manning, A large annotated corpus for learning natural language inference, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, pp. 632–642. doi:10.18653/v1/D15-1075.

[9] OpenAI, New embedding models and API updates, 2024. https://openai.com/index/new-embedding-models-and-api-updates/.

[10] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, ELECTRA: Pre-training text encoders as discriminators rather than generators, in: International Conference on Learning Representations (ICLR), 2020. https://arxiv.org/abs/2003.10555.

[11] 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, arXiv preprint arXiv:1907.11692 (2019). https://arxiv.org/abs/1907.11692.

[12] Google DeepMind, Gemma 4 model card, 2026. https://ai.google.dev/gemma/docs/core/model_card_4.

A. Touché Validation Class-Recall Diagnostics

Table 3 reports class-specific recall on the 200-instance internal Touché Subtask 1 validation split. This is the balanced held-out split used for model and configuration selection.

Table 3
Per-class recall on the internal Touché validation split. Δ is fallacy recall minus non-fallacy recall. Color highlights the absolute size of the class-recall imbalance; AUROC is shown for scored classifiers and "–" for hard-label Gemma outputs.

Model	Input view	Macro-F1	AUROC	Non-fallacy recall	Fallacy recall	Δ recall
TF-IDF + SVC	base	0.610	0.657	0.620	0.600	-0.020
TF-IDF + SVC	enhanced	0.835	0.914	0.800	0.870	0.070
TF-IDF + SVC	raw+ctx	0.575	0.609	0.560	0.590	0.030
TF-IDF + LogReg	base	0.615	0.663	0.640	0.590	-0.050
TF-IDF + LogReg	enhanced	0.840	0.911	0.810	0.870	0.060
TF-IDF + LogReg	raw+ctx	0.609	0.630	0.570	0.650	0.080
OpenAI 3-large + LogReg	base	0.669	0.737	0.620	0.720	0.100
OpenAI 3-large + LogReg	enhanced	0.770	0.840	0.790	0.750	-0.040
OpenAI 3-large + LogReg	raw+ctx	0.640	0.705	0.670	0.610	-0.060
ELECTRA-small	base	0.665	0.726	0.650	0.680	0.030
ELECTRA-small	enhanced	0.840	0.910	0.820	0.860	0.040
ELECTRA-small	raw+ctx	0.640	0.678	0.610	0.670	0.060
RoBERTa-large	base	0.650	0.713	0.660	0.640	-0.020
RoBERTa-large	enhanced	0.875	0.931	0.890	0.860	-0.030
RoBERTa-large	raw+ctx	0.640	0.706	0.650	0.630	-0.020
Gemma 4 31B	base	0.729	–	0.780	0.680	-0.100
Gemma 4 31B	raw+ctx	0.775	–	0.810	0.740	-0.070
Gemma 4 31B	enhanced	0.855	–	0.800	0.910	0.110
Gemma 4 31B	enhanced few-shot	0.868	–	0.760	0.980	0.220

B. LOGIC-binary Class-Recall Diagnostics

Table 4 reports the class-specific LOGIC-binary recalls used to interpret Figure 4. The valid class contains the dataset's caption-based entailments; the fallacy class contains LOGIC fallacy examples. The repeated Gemma rows use the same LOGIC-binary predictions because the LOGIC prompt is independent of the Touché input view.

Table 4
Per-class recall on the fixed LOGIC-binary diagnostic set. Δ is valid recall minus fallacy recall. Color highlights the absolute size of the class-recall imbalance; AUROC is shown for scored classifiers and "–" for hard-label Gemma outputs.

Model	Input view	Macro-F1	AUROC	Valid recall	Fallacy recall	Δ recall
TF-IDF + SVC	base	0.538	0.544	0.590	0.488	0.103
TF-IDF + SVC	enhanced	0.540	0.609	0.863	0.292	**0.571**
TF-IDF + SVC	raw+ctx	0.551	0.564	0.603	0.502	0.101
TF-IDF + LogReg	base	0.548	0.555	0.610	0.488	0.122
TF-IDF + LogReg	enhanced	0.552	0.629	0.866	0.307	**0.559**
TF-IDF + LogReg	raw+ctx	0.557	0.570	0.619	0.498	0.121
OpenAI 3-large + LogReg	base	0.473	0.461	0.478	0.468	0.009
OpenAI 3-large + LogReg	enhanced	0.615	0.624	0.820	0.438	**0.382**
OpenAI 3-large + LogReg	raw+ctx	0.500	0.487	0.501	0.499	0.001
ELECTRA-small	base	0.250	0.348	0.016	0.622	**-0.607**
ELECTRA-small	enhanced	0.552	0.597	0.995	0.237	**0.757**
ELECTRA-small	raw+ctx	0.285	0.320	0.004	0.783	**-0.780**
RoBERTa-large	base	0.247	0.274	0.013	0.618	**-0.605**
RoBERTa-large	enhanced	0.538	0.656	0.987	0.224	**0.763**
RoBERTa-large	raw+ctx	0.299	0.336	0.068	0.648	**-0.580**
Gemma 4 31B	base	0.927	–	0.907	0.947	-0.040
Gemma 4 31B	raw+ctx	0.927	–	0.907	0.947	-0.040
Gemma 4 31B	enhanced	0.927	–	0.907	0.947	-0.040
Gemma 4 31B	enhanced few-shot	0.927	–	0.907	0.947	-0.040

C. Sahai-Aligned Public-Test Class-Recall Diagnostics

Table 5 reports the same class-specific recall diagnostic on the 195 public-test rows aligned to Sahai et al. labels. Unlike LOGIC-binary, both classes come from Reddit-style argument data. The proxy is reported only as an auxiliary diagnostic; it is not an official Touché test score and was not used for model or checkpoint selection. Rows without a corresponding Sahai-proxy value in Figure 4 are omitted.

Table 5
Per-class recall on the Sahai-aligned public-test proxy. Δ is fallacy recall minus non-fallacy recall. Color highlights the absolute size of the class-recall imbalance; AUROC is shown for scored classifiers and "–" for hard-label Gemma outputs.

Model	Input view	Macro-F1	AUROC	Non-fallacy recall	Fallacy recall	Δ recall
TF-IDF + SVC	base	0.620	0.706	0.614	0.628	0.014
TF-IDF + SVC	enhanced	0.944	0.968	0.941	0.947	0.006
TF-IDF + SVC	raw+ctx	0.651	0.679	0.663	0.638	-0.025
TF-IDF + LogReg	base	0.615	0.721	0.614	0.617	0.003
TF-IDF + LogReg	enhanced	0.933	0.967	0.931	0.936	0.005
TF-IDF + LogReg	raw+ctx	0.656	0.691	0.644	0.670	0.027
OpenAI 3-large + LogReg	base	0.748	0.801	0.782	0.713	-0.069
OpenAI 3-large + LogReg	enhanced	0.856	0.929	0.851	0.862	0.010
OpenAI 3-large + LogReg	raw+ctx	0.625	0.714	0.624	0.628	0.004
ELECTRA-small	base	0.691	0.776	0.733	0.649	-0.084
ELECTRA-small	enhanced	0.887	0.932	0.871	0.904	0.033
ELECTRA-small	raw+ctx	0.661	0.704	0.663	0.660	-0.004
RoBERTa-large	base	0.697	0.753	0.683	0.713	0.030
RoBERTa-large	enhanced	0.949	0.971	0.941	0.957	0.017
RoBERTa-large	raw+ctx	0.649	0.724	0.703	0.596	-0.107
Gemma 4 31B	base	0.805	–	0.822	0.787	-0.035
Gemma 4 31B	raw+ctx	0.734	–	0.832	0.638	-0.193
Gemma 4 31B	enhanced	0.877	–	0.871	0.883	0.012
Gemma 4 31B	enhanced few-shot	0.892	–	0.851	0.936	0.085

D. Train-View × Eval-View Crossing

Table 6 reports the base/enhanced train-view by eval-view crossing: each scored system is trained on either base or enhanced input and evaluated on either base or enhanced input, on both the Touché validation split and the Sahai-aligned proxy. Two broad patterns hold. First, the enhanced diagonal (e→e) is the peak for every system, but the same model collapses when evaluated off that view. Second, evaluating base-trained models on enhanced input (b→e) does not produce the same collapse as evaluating enhanced-trained models on base input (e→b). Instead, it usually gives a small improvement over base-on-base evaluation (b→b); this holds for all Touché validation rows and all Sahai-proxy rows except RoBERTa-large (0.686 vs. 0.697). Thus, we treat the crossing table primarily as evidence of asymmetric format dependence, with the TF-IDF and cue-span diagnostics providing the stronger evidence for answer-correlated lexical cues. The macro-F1 collapse exceeds the AUROC collapse because the enhanced-trained threshold is miscalibrated for base input; AUROC, being threshold-free, isolates the residual loss of ranking.

Table 6
Train-view → eval-view crossing (macro-F1 and AUROC) for the scored non-Gemma systems. Views: b=base, e=enhanced; x→y trains on x and evaluates on y. The enhanced diagonal (e→e, bold) is the peak for every system. All classifiers are trained on the 738-row train split only. Gemma is omitted (a prompted generator has no train view); the fine-tuned encoders cover the base↔enhanced sub-grid only, as raw+ctx off-diagonals would require re-running each checkpoint.

	Macro-F1				AUROC			
Model	b→b	b→e	e→e	e→b	b→b	b→e	e→e	e→b
Touché validation								
TF-IDF + SVC	0.610	0.640	**0.835**	0.555	0.657	0.708	**0.914**	0.604
TF-IDF + LogReg	0.615	0.644	**0.840**	0.560	0.663	0.711	**0.911**	0.605
OpenAI 3-large + LogReg	0.669	0.710	**0.770**	0.676	0.737	0.763	**0.840**	0.724
ELECTRA-small	0.665	0.700	**0.840**	0.569	0.726	0.784	**0.910**	0.657
RoBERTa-large	0.650	0.694	**0.875**	0.546	0.713	0.732	**0.931**	0.684
Sahai-aligned proxy								
TF-IDF + SVC	0.620	0.666	**0.944**	0.606	0.706	0.753	**0.968**	0.722
TF-IDF + LogReg	0.615	0.681	**0.933**	0.612	0.721	0.766	**0.967**	0.732
OpenAI 3-large + LogReg	0.748	0.784	**0.856**	0.704	0.801	0.845	**0.929**	0.807
ELECTRA-small	0.691	0.758	**0.887**	0.595	0.777	0.826	**0.932**	0.732
RoBERTa-large	0.697	0.686	**0.949**	0.567	0.753	0.775	**0.971**	0.758

E. Encoder Attention-Cue Diagnostic

As a qualitative check on the lexical leakage diagnostic, we measured whether fine-tuned encoders route disproportionate classification-token attention to the same high-coefficient TF-IDF cue spans. For each input view v, let G_v be the cue list formed from the top 30 highest-absolute-coefficient unigram and bigram features from the TF-IDF logistic-regression diagnostic. We drop very short or generic cue strings (length below four characters, and the tokens `there`, `first`, `long`, and `over`) before span matching. For an example x_i, let S_i be the set of character spans in x_i that exactly match an n-gram in G_v, with bigrams matched as full phrases rather than split into independent word cues.

After tokenization, let T_i be the set of non-special token indices and let $C_i \subseteq T_i$ be the cue-token set:

$$C_i = \{t \in T_i \,:\, \text{offset}(t) \cap s \neq \emptyset \text{ for some } s \in S_i\}.$$

We kept validation rows with $|S_i| \geq 2$ and selected a small balanced probe set by taking the fallacy and non-fallacy rows with the most cue spans. This formed a 12-example probe set for each input view: six fallacy and six non-fallacy validation rows. For a model with L layers and H heads, let $a_{i,t}^{\ell,h}$ denote the attention weight from the classification token to token t in layer ℓ and head h. We average attention over all heads and the final four layers $\mathscr{L}$:

$$\bar{a}_{i,t} = \frac{1}{|\mathscr{L}|H} \sum_{\ell \in \mathscr{L}} \sum_{h=1}^{H} a_{i,t}^{\ell,h}.$$

We then renormalize over non-special tokens,

$$\tilde{a}_{i,t} = \frac{\bar{a}_{i,t}}{\sum_{u \in T_i} \bar{a}_{i,u}}, \quad t \in T_i,$$

so cue-attention lift is

$$\text{lift}(i) = \frac{\sum_{t \in C_i} \tilde{a}_{i,t}}{|C_i|/|T_i|}.$$

A value of 1 is the uniform-attention baseline over non-special tokens; values above 1 mean that cue tokens receive more classification-token attention than their token share would predict. This is a routing diagnostic, not a causal attribution method.

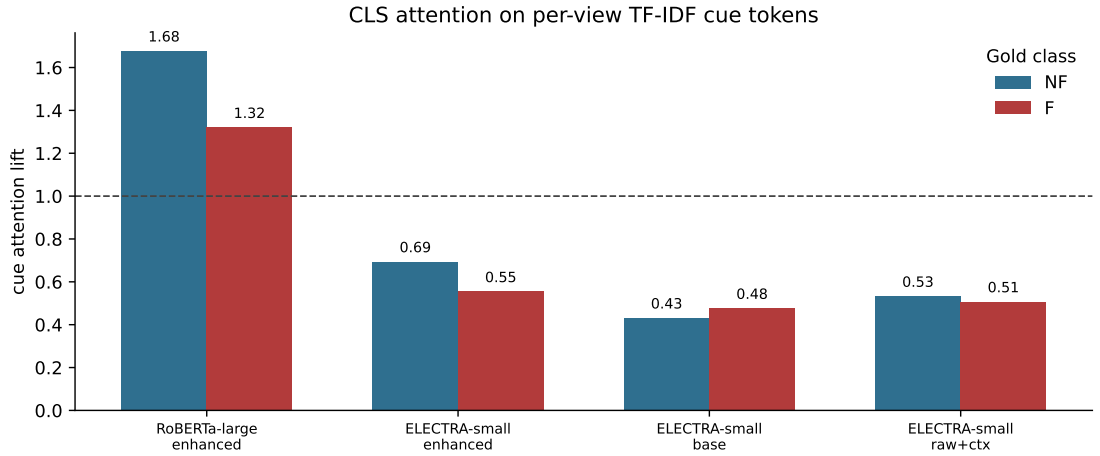

Figure 5: Classification-token attention lift on TF-IDF cue spans for the selected Touché validation probe rows. The enhanced RoBERTa checkpoint assigns elevated attention to these spans (mean lift 1.32 for fallacy rows and 1.68 for non-fallacy rows), whereas the ELECTRA probes remain below the uniform-attention baseline, which is not evidence that ELECTRA ignores the cues; this diagnostic may simply miss its signal.

F. Gemma Prompt Templates

This appendix records the prompt templates used for the Gemma Subtask 1 binary fallacy-detection runs. For Gemma 4 models, the system prompt was sent as a GenAI `system_instruction`.

F.1. Decoding and API Settings

Gemma 4 31B (`gemma-4-31b-it`) was queried through the Google GenAI API with greedy decoding (`temperature` = 0.0) and a single sample per instance. For the no-schema Gemma Subtask 1 runs reported here, we did not pass a response schema constraint to the API. Instead, we used freeform decoding with `max_output_tokens` set to 4096 and parsed the final JSON object from the model response. The four-shot enhanced condition instead used schema-constrained decoding (a response schema with `max_output_tokens` = 512) and did not request step-by-step reasoning.

F.2. Subtask 1 System Prompt

```
You are an expert in informal logic and argumentation theory annotating
Reddit-derived arguments for the Touché 2026 fallacy-detection benchmark.

Your task: decide whether the given argument commits an informal fallacy.

FALLACY CATEGORIES YOU SHOULD RECOGNIZE (Macagno-style, for context —
you will not be asked to name the type, only whether one is present):
- authority — appeal to authority: citing someone's standing as evidence
  rather than the evidence itself.
- blackwhite — false dilemma: presenting only two options when more exist.
- hasty_generalization — drawing broad conclusions from too few or
  unrepresentative cases.
- natural — appeal to nature: assuming "natural" implies good or right.
- population — appeal to popularity / bandwagon: X is true because many
  believe it.
- slippery_slope — claiming a small step inevitably leads to a larger
  negative outcome without justifying the chain.
- tradition — appeal to tradition: X is right because it has always been done.
- worse_problems — whataboutism / red herring: deflecting via comparison
  to worse issues.

A non-fallacious argument may still resemble a fallacy template by surface
form but does not actually commit one.

Output: fallacy_exists, an integer 0 or 1.
```

F.3. User Message Templates

Without explicit context, the user message was:

```
Argument:
"""
{text}
"""
```

For the explicit raw-context condition, title and parent comment were included before the argument when available:

```
Submission title:
"""
{text_raw_title}
"""

Parent comment (the comment this argument replies to):
"""
{text_raw_parent}
"""

Argument:
```

```
"""
{text}
"""
```

The `text` slot was filled with the selected input representation, for example `text_base`, `text_enhanced`, or raw text plus context fields depending on the run.

F.4. Freeform Output

The expected final JSON object contained a single required integer field:

```
{"fallacy_exists": 0 | 1}
```

For the no-schema Gemma runs, we appended:

```
Reason step-by-step about whether the argument commits a fallacy, then
output your final label as a JSON object inside a ```json code fence with
this exact key: fallacy_exists (0 or 1).
```

F.5. Few-Shot Enhanced Prompt

For the enhanced few-shot Gemma condition, the same system prompt was used and the user message began with a balanced set of four labeled examples (two non-fallacious, two fallacious) from the Touché training split:

````
Here are some examples of Reddit arguments with their correct fallacy
classification using text_enhanced:

Argument:
"""
{example_text_enhanced}
"""
```json
{
  "fallacy_exists": {0_or_1}
}
```

...

Now classify the following argument:
Argument:
"""
{target_text_enhanced}
"""
Output your final label as a JSON object with this exact key:
fallacy_exists.
````

G. Training Configuration Summary

Table 7 summarizes the fine-tuned encoder runs used in the Subtask 1 analysis. Both encoder families are 512-token-class models (ELECTRA config `max_position_embeddings=512`; RoBERTa config `max_position_embeddings=514` because of its positional-offset convention), but all runs used a stricter tokenizer cap of 384 tokens, including special tokens. Inputs longer than this cap were truncated; this was rare in our Subtask 1 splits, affecting at most 0.5% of rows for any input view. The binary heads were trained as two-class single-label classifiers with cross-entropy loss. Common settings were weight decay 0.01, warmup ratio 0.1, random seed 42, and fp16 training on CUDA.

Table 7
Encoder fine-tuning configurations. Batch denotes per-device train batch size; accumulation is gradient accumulation.

Model	Input	Base model	Device	Max len.	Epochs	LR	Batch	Accum.
ELECTRA-small	base	google/electra-small-discriminator	T4	384	5	3e-5	16	1
ELECTRA-small	raw+ctx	google/electra-small-discriminator	T4	384	5	3e-5	16	1
ELECTRA-small	enhanced	google/electra-small-discriminator	T4	384	5	3e-5	16	1
RoBERTa-large	base	FacebookAI/roberta-large	T4	384	3	2e-5	4	2
RoBERTa-large	raw+ctx	FacebookAI/roberta-large	T4	384	3	2e-5	4	2
RoBERTa-large	enhanced	FacebookAI/roberta-large	T4	384	3	2e-5	4	2