Veritas at Touché: Input Enrichment or Label Leakage? Fine-tuning Encoder Models for Fallacy Detection

Touché at CLEF 2026

Farzaneh Bayan Memar[1,*], Janiek de Rijke[1], Niek Biesterbos[1] and Mark den Ouden[1]

[1]University of Groningen, Groningen, The Netherlands

Abstract

Encoder-only models fine-tuned on the Touché 2026 fallacy dataset reach very high scores, but only when they are given the enhanced version of each argument. We first reproduce this effect: RoBERTa-large and ModernBERT-large fine-tuned on the enhanced input variant reach a macro-F1 of 0.904 on fallacy detection and 0.956 on fallacy classification, far above the base variant (0.723 and 0.447 respectively). We then ask what the enhanced text actually contributes, and whether the gain reflects genuine reasoning or an artifact of how the data was constructed. Through a series of controlled experiments we show that the gain is largely label-conditioned lexical leakage introduced by the rewrite. A simple bag-of-words classifier matches the fine-tuned encoders on enhanced text but not on base text; the most predictive words are the fallacy scheme terms themselves; the enhancement injects these cues into true fallacies far more than into the hard negatives that resemble them; the cues do not transfer to base text; and a manual inspection finds that almost every enhanced rewrite states the fallacy schema explicitly. We conclude that making argument structure explicit does help, but that most of the measured benefit comes from cues that will not be available in realistic settings.

Keywords

Fallacy detection, Argument mining, Encoder models, Dataset artifacts, Label leakage

1. Introduction

Fallacies are defined as illogical and invalid steps in the formulation of arguments. They can be used intentionally, to influence, manipulate or persuade others, or unintentionally, as a result of linguistic limitations, cognitive biases or carelessness, and may even appear rather convincing at a first glance. As such, fallacies can contribute to faulty governing, the reinforcement of present-day biases, and the dispersal of misinformation. In the era where misinformation is more common than ever, the automatic detection and classification of fallacies is thus extremely important. However, both the detection and classification of fallacies are quite complex: the language used in arguments is often context-dependent, ambiguous, and complicated. In addition, few datasets use the same fallacy types, which results in a relatively low amount of labeled data. Fallacies can also evolve over time, which makes detection and classification harder when models are trained on older data. In this paper, we describe our participation (under the name *Veritas*) in the Touché 2026 shared task on fallacy detection and classification [1, 2], where systems are evaluated on TIRA [3]. We fine-tune encoder-only models for both sub-tasks and experiment with the two input variants that the dataset provides, a base text and an enhanced text, which differ in how explicitly the argument structure and the fallacy cues are surfaced. Our systems reach high scores, but almost all of that performance depends on a single design choice: using the enhanced rewrite of each argument rather than the base text. Switching from base to enhanced raises detection by roughly 18 percentage points and classification by up to 51 percentage points, while the model, the training data, and the labels stay the same. A naive reading is that the enhanced rewrite simply makes the argument structure clearer, so the model can do its job better. A less comfortable reading is that the rewrite makes the task artificially easy, by writing information that is correlated

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

*Corresponding author.

f.bayan.memar@student.rug.nl (F. Bayan Memar); j.de.rijke.1@student.rug.nl (J. de Rijke); n.a.biesterbos@student.rug.nl (N. Biesterbos); m.e.den.ouden@student.rug.nl (M. den Ouden)

with the label directly into the input. This leads to the central research question of this paper: what is it about the enhanced text that drives these gains, and does the gain show that the model has learned to recognise fallacious reasoning, or that the rewrite has made the task easier in a way that will not generalise? We answer this with a series of controlled experiments that isolate the contribution of the enhanced text. We find that the gain is largely label-conditioned lexical leakage. The enhanced rewrite tends to state the fallacy scheme explicitly, and it does so far more often for the true fallacies than for the hard negatives that resemble them, so even a simple bag-of-words model can solve the task on enhanced text while failing on base text. Our contribution is therefore twofold. First, we present a strong encoder-based system for the shared task and confirm that fine-tuned encoders clearly outperform zero-shot baselines. Second, and more importantly for this project, we provide a diagnostic analysis showing that the headline gains are largely an artifact of label-aware input enrichment, which has implications for how such datasets should be built and how scores on them should be interpreted.

2. Related Work

Automatic fallacy detection is a relatively young field within computational argumentation. The most relevant contributions to the field were done only in the last decade. Research has developed along two main lines: the construction of annotated datasets, and the development of classification models trained on them.

Datasets. Several datasets have been created to help detect fallacies, and they differ in topic, number of categories, and level of detail in the annotations. Sahai et al. [4] introduce the `informal_fallacies` dataset of Reddit comments annotated for eight informal fallacy types, which forms the basis of `touchefallacy_2026` used in this work. A key design choice in their dataset is the inclusion of *hard negatives*: non-fallacious arguments that structurally resemble a specific fallacy type, making the classification task substantially harder than distinguishing fallacious from randomly sampled non-fallacious text. Jin et al. [5] introduce LOGIC, a larger dataset of 2,449 examples covering 13 fallacy types, automatically crawled from educational quiz websites. While broader in taxonomy, the automated collection process raises questions about annotation consistency. Helwe et al. [6] present MAFALDA, a unified benchmark merging multiple existing datasets into a single resource of nearly 10,000 texts, organized under a three-tier taxonomy inspired by Aristotle's rhetorical categories (ethos, logos, pathos). A different angle is taken by Haddadan et al. [7], whose ElecDeb60to20 corpus annotates U.S. presidential debate transcripts from 1960 to 2020 for six fallacy types at the token level using BIO tagging, enabling span-level detection rather than document-level classification.

Models. Early work on fallacy classification showed that zero-shot approaches perform poorly on this task. Sahai et al. [4] report that both GPT-3 and a zero-shot RoBERTa-MNLI model achieve only around 12% macro-F1 on their dataset, barely above the random baseline, highlighting the difficulty of the task without task-specific training. Fine-tuned encoder models improve substantially: Sahai et al. [4] and Jin et al. [5] show that fine-tuned BERT achieves around 45 to 46% macro-F1, while fine-tuning Electra yields 53%. Incorporating explicit argument structure further helps: a BERT model augmented with contextual features reaches 58.41%, and a structure-aware Electra model achieves 58.77% [5]. These results suggest that understanding the logical structure of an argument, not just its surface form, is important for fallacy classification. Ruiz-Dolz and Lawrence [8] explore a cascaded pipeline for fallacy identification and classification using fine-tuned RoBERTa and zero-shot LLM prompting, finding that fine-tuned models outperform prompting approaches in low-resource settings.

Shortcut learning and dataset artifacts. A recurring finding in natural language processing is that strong benchmark scores do not always reflect the ability they appear to measure. Gururangan et al. [9] show that natural language inference models can classify examples without even reading the premise, because annotation artifacts leak the label into the hypothesis. Niven and Kao [10] find that the high

accuracy of BERT on an argument reasoning task is driven by spurious statistical cues rather than reasoning, and collapses once those cues are removed. McCoy et al. [11] show that inference models rely on shallow syntactic heuristics that fail on adversarial test sets. More generally, Geirhos et al. [12] describe shortcut learning as a pervasive failure mode in which models exploit superficial features that happen to correlate with labels in the training distribution. Our analysis applies this lens to fallacy detection, and asks whether the enhanced input variant introduces exactly this kind of shortcut.

Relation to our work. Our work builds directly on Sahai et al. [4]'s dataset and addresses the same classification taxonomy. Consistent with prior findings, we find that fine-tuned encoder models substantially outperform zero-shot baselines. We extend the comparison by evaluating both RoBERTa-large [13] and the more recently introduced ModernBERT-large [14], and by comparing the two input variants (`text_base` and `text_enhanced`) that differ in how explicitly the argument structure and fallacy cues are surfaced in the input text. Unlike a standard system description, our main goal is to diagnose why the enhanced variant works so well, connecting our results to the literature on dataset artifacts and shortcut learning described above.

3. Task and Data

3.1. Task Definition

We address two sub-tasks of the Touché 2026 shared task on fallacy detection and classification [1, 2]. **Sub-Task 1: Fallacy Detection** is a binary classification task. Given an argument, the system must determine whether it is fallacious or not. The input is a short argument text, and the expected output is one of two labels: *fallacy* or *non-fallacy*. A key challenge of this sub-task is that each non-fallacious argument was specifically selected because its reasoning pattern resembles a particular fallacy type. This means that systems cannot rely on surface-level features alone, but must distinguish genuinely fallacious arguments from superficially similar but valid ones. **Sub-Task 2: Fallacy Classification** is an eight-way classification task. Given an argument that is known to be fallacious, the system must identify which type of fallacy it contains. The input is again an argument text, and the expected output is one of eight fallacy type labels: *authority*, *black-white*, *hasty_generalization*, *natural*, *population*, *slippery_slope*, *tradition*, and *worse_problems*.

3.2. Data

The dataset used in this work is `touchefallacy_2026`, which is built on the informal fallacies dataset introduced by Sahai et al. [4]. It consists of 938 labeled training examples and 233 test examples for which labels are withheld and predictions are submitted to TIRA for evaluation. The dataset, including the test gold labels, is publicly available [15]. Each entry in the dataset contains two versions of the argument: `text_base` and `text_enhanced`. The base version is a self-contained rewrite of the original Reddit comment that integrates the conversational context (the parent comment and the title) into a single coherent text. The enhanced version is described by the organizers as a rewrite of the base text that explicitly surfaces fallacy and scheme information. Both versions are provided by the organizers for the training set and for the test set, so the enhanced text is available at test time as well. The organizers document that the `text_enhanced` input variant was produced by rewriting each base text with a large language model (Claude Opus 4.6) with knowledge of the corresponding scheme and fallacy labels, so that the scheme dimensions become clearly identifiable and, when the argument is fallacious, the fallacious reasoning pattern is made more explicit and central.[1] For Sub-Task 1, all 938 training examples are used, with labels indicating whether the argument is fallacious or not. For Sub-Task 2, only the 465 fallacious entries are used, each labeled with one of the eight fallacy types. Each entry also records a `resembles_fallacy` field, which for both fallacies and hard negatives names the fallacy type that

[1] https://touche.webis.de/clef26/touche26-web/fallacy-detection-paper-guidelines.html

the argument resembles. We make use of this field in our analysis. We split the training data into 80% train, 10% validation, and 10% test using stratified sampling to preserve the label distribution. As can be seen in Table 1 and Table 2, the labels for both detection and classification are roughly balanced.

Table 1
Dataset statistics for fallacy detection.

Label	Train	Val	Test	Total
Non-fallacy	379	47	47	473
Fallacy	371	47	47	465
Total Examples	750	94	94	938

Table 2
Dataset statistics for fallacy classification.

Label	Train	Val	Test	Total
Tradition	51	7	7	65
Slippery Slope	50	6	6	62
Natural	49	6	6	61
Worse Problems	49	6	6	61
Authority	47	6	6	59
Hasty Generalization	43	6	6	55
Black-White	44	5	5	54
Population	38	5	5	48
Total Examples	371	47	47	465

3.3. An illustrative example

To make the difference between the two variants concrete, consider one training argument that is labeled *authority*. The base version reads:

> [...] Nathaniel Hawthorne, Walt Whitman, and Mark Twain were Anti-Stratfordians, if that's the kind of evidence you're into. [...] it's a certainty that if you think it was William from Stratford then you haven't done any research on it at all.

The enhanced version of the same argument reads:

> [...] And consider the credibility of those who've reached this conclusion, Nathaniel Hawthorne, Walt Whitman, and Mark Twain were all Anti-Stratfordians. These are some of the greatest literary minds in history, and they all believed Shakespeare wasn't the true author, so that should tell you something about where the evidence really points. [...]

The base version mentions three names as evidence. The enhanced version adds an explicit appeal to their credibility and standing as great minds, and frames their agreement as proof. In other words, it spells out the appeal to authority, which is exactly the gold label for this example. This pattern, where the rewrite states the fallacy scheme that corresponds to the label, recurs throughout the dataset and motivates our analysis.

4. Method

We approach both sub-tasks as text classification problems and fine-tune encoder-only language models on the training data. For Sub-Task 1, the model learns to distinguish fallacious from non-fallacious arguments. For Sub-Task 2, only the fallacious arguments are used, and the model learns to assign one of eight fallacy type labels. For both input variants, the argument text is the only input to the model, with no additional features.

4.1. Fine-tuning Encoder Models

For our fallacy detection baseline, we fine-tune RoBERTa-base without tuning the hyperparameters. We fine-tune two encoder-only models for our main approach: RoBERTa-large [13] and ModernBERT-large [14]. Both models are fine-tuned separately for each sub-task and for each input variant, resulting in several model configurations in total. Our internal validation split, extracted from the training data, was used for early stopping and for selecting the checkpoint with the highest macro-F1 score. The held-out test split was only used for the final evaluation and was never seen during training or model selection.

4.1.1. Ensemble

We combine the predictions of ModernBERT-large and RoBERTa-large (both enhanced variant) for Sub-Task 2 using majority voting. In case of a tie, we defer to ModernBERT-large, as it achieved the higher macro-F1 on the internal validation split. Since the ensemble consists of only two models, any disagreement is resolved by deferring to ModernBERT-large, making the ensemble score identical to ModernBERT-large alone (0.956).

4.1.2. Implementation Details

Each argument is tokenized and truncated to a maximum length of 512 tokens. For our baseline we train for 3 epochs with a batch size of 16, a learning rate of 2e-5, a weight decay of 0.01. Our main approach used the same hyperparameters, except for the amount of training epochs (max 12 instead of 3), the use of early stopping with a patience of 3 epochs, and a warmup ratio of 0.1. All experiments were conducted using the HuggingFace Transformers library in python [16], utilizing the bf16 precision format and running on a single A100 GPU. Link to our github repository: https://github.com/JDRijke/LTP/

4.2. Prompting LLMs

Apart from testing fine-tuned encoder models, we also prompted four different decoder-only LLMs: Llama3.2-3B-it [17], Llama3.1-8B-it [17], Gemma3-12B-it [18], and Gemma4-E4B-it. All models were tested using zero-shot and few-shot prompting (2 examples per class) on both input variants across fallacy detection and classification, resulting in 32 experiments. We used task-specific system prompts to instruct the model to act as an expert in informal logic and argumentation theory. For the classification task, definitions of all eight fallacy categories were embedded directly in the prompt. User prompts presented the argument text and specified the required output format.

4.3. Diagnostic Analysis

The second part of our method is a set of experiments designed to isolate why the enhanced input variant helps. Across these experiments we use a deliberately simple model as a probe: a TF-IDF bag-of-words representation (word unigrams and bigrams) followed by a logistic regression classifier, implemented with scikit-learn [19]. We use this proxy for two reasons. First, it captures only shallow lexical information, so if it can reproduce the enhanced effect, that effect must be largely lexical. Second, as we report below, the proxy reaches the level of the fine-tuned encoders on enhanced text, which makes it a faithful and inexpensive stand-in for the analysis. We use the same data splits as for the encoders, and we additionally report five-fold cross-validation to account for the small size of the held-out split. We run the following experiments. (1) A bag-of-words comparison of base against enhanced input variants for both sub-tasks. (2) An inspection of the most predictive tokens per fallacy class. (3) A hard-negative analysis that measures, separately for true fallacies and for the hard negatives that resemble the same type, how often the text contains a cue word for that type. (4) A length control, in which the enhanced text is truncated to the word count of the corresponding base text. (5) A transfer experiment, in which the proxy is trained on one variant and tested on the other. (6) A structure-only experiment, in which the model is given the structured claim and supports (`argument_base` and

`argument_enhanced`) instead of the prose. (7) A masking experiment, in which fallacy cue words are removed from the enhanced text. (8) A manual annotation of a sample of enhanced rewrites.

4.4. Evaluation

All systems are evaluated using macro-averaged F1, the official metric of Touché 2026 [1, 2]. To assess performance before the official test release, we reserve 10% of the training data as an internal test split, using stratified sampling to maintain class balance. We additionally examine per-class behaviour to better understand the model across the eight fallacy types.

5. Results

5.1. Encoder-only results

Table 3 shows the macro-F1 scores we obtained on our internal test split (10% of the training data, held out before training).

Table 3
Macro-F1 scores of the fine-tuned encoders on the internal test split. ST1 = Fallacy Detection, ST2 = Fallacy Classification.

Sub-task	Model	Var.	F1
ST1	RoBERTa-base	base	0.626
ST1	RoBERTa-large	base	0.723
ST1	RoBERTa-large	enhanced	0.904
ST2	ModernBERT-large	base	0.447
ST2	RoBERTa-large	base	0.667
ST2	RoBERTa-large	enhanced	0.928
ST2	ModernBERT-large	enhanced	**0.956**
ST2	Ensemble	enhanced	0.956

Note that ModernBERT-large was only evaluated on Sub-Task 2, and RoBERTa-base only on Sub-Task 1, due to time and compute constraints. The clearest finding is that for the models and sub-tasks we evaluated, the enhanced text variant consistently outperforms the base variant. For example, RoBERTa-large jumps from 0.723 to 0.904 on Sub-Task 1, and ModernBERT-large jumps from 0.447 to 0.956 on Sub-Task 2 when switching from base to enhanced text. For Sub-Task 2, ModernBERT-large with the enhanced variant gave the overall best result (0.956). For the base variant, RoBERTa-large (0.667) performed better than ModernBERT-large (0.447), which shows that ModernBERT benefits more from the richer input. For the official leaderboard, we selected ModernBERT-large enhanced, RoBERTa-large enhanced, and the Ensemble as our three runs, as these are the three highest-scoring submissions on our internal evaluation. These scores are based on our internal evaluation split and may differ from the official TIRA scores. The size of the base-to-enhanced gap, and the fact that it appears for every model and both sub-tasks, is what motivates the rest of the paper. In the next section we take the enhanced effect apart.

5.2. Decoder-only results

Table 4 shows the fallacy detection scores of all decoder-only LLMs we evaluated on our internal test split. One thing that can be noticed is the fact that a relatively small model performs the best on fallacy detection: Gemma 4 E4B, with the highest scores (tied for enhanced zero-shot) for all input and prompt variants. Another remarkable result is that enhanced zero-shot outperforms all other variants across all four models, similar to the results of the encoder-only models. However, there is no big difference in performance between different model sizes for enhanced zero shot, meaning that the worst models

benefit the most from the enhanced input. Few-shot prompting, however, rarely helps and in most cases actively hurts performance. Prompting thus does not seem to be a very good approach for fallacy detection. Based on Table 5, prompting seems to be a decent approach for fallacy classification, with most models scoring close to or above 90% macro-F1. Here the gap between base and enhanced text is even greater than for fallacy detection. Few-shot enhanced is the best approach for 75% models, but it drops 35% in performance for Llama 3.2 3B, which could be a result of the size of the model. Gemma 4 E4B few-shot enhanced nearly matches our fine-tuned model ModernBERT-large.

Table 4
Macro F1 scores for Subtask 1 (Fallacy Detection).

Model	Base Zero-shot	Base Few-shot	Enhanced Zero-shot	Enhanced Few-shot
Llama 3.2 3B	0.416	0.395	0.678	0.379
Llama 3.1 8B	0.550	0.495	**0.691**	0.518
Gemma 4 E4B	**0.552**	**0.560**	**0.691**	**0.683**
Gemma 3 12B	0.470	0.474	0.681	0.551

Table 5
Macro F1 scores for Subtask 2 (Fallacy Classification).

Model	Base Zero-shot	Base Few-shot	Enhanced Zero-shot	Enhanced Few-shot
Llama 3.2 3B	0.381	0.222	0.867	0.511
Llama 3.1 8B	0.640	0.357	0.911	0.914
Gemma 4 E4B	0.652	**0.691**	**0.933**	**0.954**
Gemma 3 12B	**0.697**	0.640	0.893	0.931

5.3. Official test-set results

Table 6 reports our runs on the official held-out test set of 233 examples, which the organizers designate as the authoritative evaluation, superseding the preliminary TIRA scores.[2] On Sub-Task 1, our RoBERTa-large enhanced run reaches a macro-F1 of 0.927, which places it fourth of the thirteen enhanced runs, while the base run reaches 0.737 and places third of the twelve base runs. The large gap between the base and enhanced variants that we observed on our internal split therefore reappears on the official test set. On Sub-Task 2, our strongest run is RoBERTa-large enhanced with a macro-F1 of 0.947, ahead of both the ensemble and ModernBERT-large, which both reach 0.927. This reverses the ordering we found on our internal split, where ModernBERT-large was the strongest model. On the official test set RoBERTa-large is the better classifier, and because our two-model ensemble reduces to ModernBERT-large it offers no improvement over either single model. Our two base runs for Sub-Task 2 exceeded the three-run submission limit and are reported by the organizers for reference only.

6. Analysis

This section reports the diagnostic experiments described in subsection 4.3. Together they show that the enhanced effect is driven by shallow, label-aligned lexical cues that the rewrite adds to the input.

6.1. Results for bag-of-words model on multiple input variants

Table 7 reports five-fold cross-validated macro-F1 for the bag-of-words proxy on a range of input variants. The pattern of the fine-tuned encoders is reproduced almost exactly. On base text the proxy is weak (0.482 on classification), but on enhanced text it reaches 0.965 on classification, which is at

[2] https://touche.webis.de/clef26/touche26-web/fallacy-detection-results.html

Table 6
Official Touché 2026 test-set results for our runs (macro-averaged precision, recall, and F1). Runs are ranked separately within the base and enhanced tracks; "inf." marks runs reported for reference only because they exceeded the three-run submission limit. ST1 = Fallacy Detection, ST2 = Fallacy Classification.

Sub-task	Model	Var.	P	R	F1	Rank
ST1	RoBERTa-large	base	0.742	0.737	0.737	3
ST1	RoBERTa-large	enhanced	0.927	0.927	0.927	4
ST2	RoBERTa-large	enhanced	0.950	0.947	0.947	7
ST2	Ensemble	enhanced	0.936	0.925	0.927	9
ST2	ModernBERT-large	enhanced	0.936	0.925	0.927	10
ST2	RoBERTa-large	base	0.685	0.656	0.661	inf.
ST2	ModernBERT-large	base	0.576	0.523	0.523	inf.

the level of ModernBERT-large (0.956) and above RoBERTa-large (0.928), and 0.850 on detection. A model that has no access to word order beyond bigrams, and no pretrained knowledge, should not be able to reason about argument validity. The fact that it nonetheless matches the encoders on enhanced text indicates that the enhanced signal is overwhelmingly lexical. This is consistent with the official leaderboard, where simple TF-IDF linear models submitted by other teams rank among the top systems on the enhanced variant, reaching up to 0.954 on Sub-Task 2 and around 0.88 on Sub-Task 1, while falling well behind on the base variant, which is exactly the pattern our analysis predicts.

Table 7
Five-fold cross-validated macro-F1 of the bag-of-words proxy (standard deviation in parentheses).

Input variant	ST1	ST2
`text_base`	0.599 (0.031)	0.482 (0.062)
`text_enhanced`	0.850 (0.025)	0.965 (0.024)
enhanced, length-matched	0.743 (0.023)	0.793 (0.049)
`argument_base`	0.652 (0.039)	0.578 (0.011)
`argument_enhanced`	0.818 (0.021)	0.913 (0.026)

6.2. Inspection of most predictive tokens per class

If the signal is lexical, the natural question is which words carry it. Table 8 lists the most predictive tokens per class, taken from the weights of the proxy trained on enhanced classification text. For almost every class the strongest features are the name of the fallacy scheme or its close paraphrases: *authority* is signalled by "authority", "credentials", and "experts"; *tradition* by "tradition" and "always been"; *population* by "majority" and "many people"; *natural* by "natural" and "naturally". On base text, by contrast, the most predictive tokens are topical content words such as "government", "gun", or named entities, which carry little information about the fallacy scheme. The enhanced rewrite therefore does not just clarify the argument, it injects the vocabulary of the scheme that matches the label into the text.

6.3. Analysis of hard-negatives

The most important experiment concerns the hard negatives. Each argument records the fallacy type it resembles, so we can compare, within the same resembled type, the true fallacies against the non-fallacious hard negatives. For each group we measure how often the text contains a cue word for that fallacy type, using a small hand-built lexicon per type.

Table 9 reports the results. In the base text, true fallacies and hard negatives contain cue words at almost the same rate (pooled 28% against 22%). This is what we would expect for a well-designed

Table 8
Most predictive tokens per fallacy class for the proxy on enhanced text.

Fallacy type	Top tokens (enhanced)
authority	authority, credentials, experts
black-white	either, only two, the only
hasty_generalization	every single, my experience, single
natural	natural, nature, naturally
population	majority, many people, sheer
slippery_slope	leads to, inevitably, each step
tradition	tradition, always been, for years
worse_problems	worse, far worse, bigger

dataset, since the hard negatives were chosen precisely because they look like fallacies. In the enhanced text, however, the gap explodes: true fallacies contain cue words 90% of the time, against only 36% for the hard negatives that resemble the same type. The enhancement adds the scheme vocabulary far more often when the argument is a true fallacy than when it is a look-alike negative. Since the only thing that distinguishes the two groups is the gold label, the rewrite must be conditioned on that label. This single result explains why detection becomes easy on enhanced text: the rewrite effectively writes the binary label into the input.

Table 9
Percentage of arguments that contain a cue word for their resembled fallacy type, split by whether the argument is a true fallacy (F) or a hard negative (N).

	base		enhanced	
Resembles	**F**	**N**	**F**	**N**
authority	25	29	92	55
black-white	15	20	100	27
hasty_generalization	25	10	69	21
natural	67	50	100	60
population	31	31	92	47
slippery_slope	24	19	94	34
tradition	29	18	94	33
worse_problems	8	3	77	8
Pooled	28	22	90	36

6.4. Controlling for length

The enhanced text is on average about twice as long as the base text (110 against 55 words), so one might worry that the model simply benefits from more text. We control for this by truncating each enhanced text to the word count of its base counterpart. As shown in the length-matched row of Table 7, performance on classification drops from 0.965 to 0.793, but remains far above the base text (0.482). Length therefore accounts for only a small part of the gain.

6.5. Transfer of cues

If the enhanced text taught the model something general about fallacies, a model trained on enhanced text should still work on base text. Table 10 shows that it does not. A proxy trained on enhanced classification text collapses from 0.976 to 0.480 when tested on base text, which is essentially the base level. The same happens for detection. The model trained on enhanced text has learned cues that

are specific to the enhanced rewrite and that are absent from realistic, non-enhanced text. This is the behaviour of a shortcut, not of a transferable skill.

Table 10
Variant transfer for Sub-Task 2: macro-F1 of the proxy trained on one variant and tested on the other (internal test split).

	test base	test enhanced
train base	0.459	0.777
train enhanced	0.480	0.976

6.6. Structure-only experiment

The enhanced text bundles two changes: it makes the argument structure explicit, and it states the fallacy scheme. We can partly separate these using the structured fields. Feeding only the claim and supports of `argument_base`, which is a structured decomposition without the enhanced prose, raises classification from 0.482 to 0.578, a gain of about 10 points (Table 7). The enhanced structured field `argument_enhanced` reaches 0.913, and the full enhanced prose reaches 0.965. The legitimate benefit of making structure explicit is therefore modest, while the large remaining jump comes from the label-aware rewriting of the content. We note that `argument_base` is itself produced by the organizers and may already contain some of the same signal, so this decomposition is a conservative upper bound on the structural benefit rather than a clean separation.

6.7. Masking experiment

Finally, we ask whether the leakage could be removed by deleting a few obvious keywords. We remove the fallacy names and their closest variants from the enhanced text, and separately the ten most predictive content tokens per class, and then retrain the proxy. Neither intervention restores the difficulty of the base text: classification stays at or above 0.98. After the obvious cues are removed, the classifier simply relies on the next most predictive words, because the rewrite saturates the text with label-aligned signal. The leakage is not localised in a handful of keywords, it is spread across the whole rewrite, which is consistent with a rewrite that was produced with knowledge of the label.

6.8. Manual annotation

To check that these quantitative findings reflect what is actually in the text, we manually inspected a sample of 24 fallacious arguments, three per class, comparing the base and enhanced versions. For each pair we asked what the enhancement adds: the literal fallacy name, an explicit paraphrase of the fallacy scheme, intensifying assertion phrases such as “the fact is” or “that proves”, or nothing label-relevant. In all 24 cases the enhanced version states the fallacy scheme explicitly, often together with the literal scheme word, and in many cases it completes a reasoning move that the base version only hints at. None of the 24 enhanced rewrites added nothing label-relevant. The qualitative picture matches the quantitative one: the enhancement turns an implicit or ambiguous argument into one that openly commits the labeled fallacy.

7. Discussion

Taken together, the experiments give a consistent answer to our research question. The enhanced variant helps mainly because it surfaces the fallacy scheme that corresponds to the gold label, in a way that a simple lexical model can exploit. The benefit of making argument structure explicit, which is the framing one would naturally give to such a rewrite, is real but modest. The larger part of the gain is label-conditioned lexical leakage. The hard-negative experiment is the clearest evidence for

this, because it shows that the rewrite treats true fallacies and their look-alike negatives differently, which is only possible if the rewrite has access to the label. This reframes the strong scores in Table 3. Because the enhanced text is provided for the test set as well, the high leaderboard scores are valid within the benchmark, and our system is a competitive entry to the shared task. What the scores do not measure, however, is the ability to recognise fallacious reasoning from the kind of text one would actually encounter. On base text, which is closer to a real argument, the same models drop to between 0.45 and 0.72. A system built on the enhanced variant would not be usable in a setting where no label-aware rewrite is available, which is precisely the setting fallacy detection is meant for.

7.1. Error analysis

The errors of the fine-tuned models are consistent with this picture. For fallacy detection, our best model (RoBERTa-large enhanced) made only nine mistakes on the 94 example internal test split. Five were false negatives, of which three were of the hasty-generalization type, and four were false positives. Most of the false positives involved arguments that use formal or credential-based language, which the model associates with the appeal to authority. For fallacy classification, on the internal test split ModernBERT-large made only two mistakes out of 47 examples, both of them confusions between fallacy types that justify a claim by appealing to an external reference (authority confused with worse_problems, and population confused with authority). In other words, the few remaining errors occur exactly where the surface cue is ambiguous between two schemes, which is what we would expect from a model that keys on those cues.

7.2. Implications

Our findings carry two practical implications. For dataset design, providing an enhanced rewrite that is conditioned on the label, however helpful it looks, can turn a hard task into one that is solvable by lexical matching, and can erase the careful work that went into constructing hard negatives. If enhanced fields are provided, it would be valuable to document how they were generated and to keep an evaluation track that uses only label-blind input. For model evaluation, our results are a reminder that a high score on an enriched input is not by itself evidence of reasoning ability, and that simple lexical baselines and transfer tests are cheap and informative checks.

8. Limitations

A first limitation is the small size of the internal test set, with only 94 examples for Sub-Task 1 and 47 for Sub-Task 2. We mitigate this by reporting five-fold cross-validation for the analysis, which gives more stable estimates with small standard deviations, but the encoder numbers in Table 3 should still be read with some caution. A second limitation is that most of our analysis uses the bag-of-words proxy rather than the fine-tuned encoders. We use the proxy because it tracks the encoders closely on this data and isolates the lexical signal, but the headline ablations could be repeated with the encoders to confirm that the conclusions are identical. A third limitation is that we did not establish the source of the enhanced gain causally. The cleanest test would be to regenerate the enhanced text in two conditions, one with access to the gold label and one without, and to compare models trained on each. We did not run this experiment. The organizers document that the enhanced text was produced by a label-aware language-model rewrite, which is consistent with our findings, but a controlled label-blind regeneration would still be needed to turn our strong observational evidence into a causal claim. Finally, the cue lexicon used in the hard-negative analysis is hand-built, and for a few classes the scheme word overlaps with ordinary topic vocabulary (for example “natural”), which makes the test conservative for those classes.

9. Conclusion and Future Work

We participated in the Touché 2026 shared task on fallacy detection and classification, and showed that fine-tuned encoder models reach strong scores when given the enhanced input variant. The main contribution of this paper, however, is a diagnostic analysis of why the enhanced variant helps so much. Across eight experiments we found that the gain is largely label-conditioned lexical leakage: a bag-of-words model matches the encoders on enhanced text, the most predictive words are the fallacy scheme terms, the enhancement injects these cues into true fallacies far more than into the hard negatives that resemble them, the cues do not transfer to base text, and a manual inspection finds the scheme stated explicitly in almost every enhanced rewrite. Making argument structure explicit does help, but the bulk of the measured benefit comes from cues that will not be present in realistic input. Future work follows directly from our limitations. The label-blind regeneration experiment would establish the effect causally, and contacting the organizers would settle how the enhanced text was produced. Beyond this dataset, training and evaluating on broader fallacy resources such as MAFALDA [6], and exploring decoder-based models with few-shot or chain-of-thought prompting, would test whether genuine fallacy reasoning can be learned from input that has not been enriched with the label.

Declaration on Generative AI

AI tools were mainly used for writing and debugging code and for generating LaTeX tables. All code was manually reviewed and thoroughly tested.

References

[1] 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.

[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 — 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.

[3] 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.

[4] S. Y. 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, Association for Computational Linguistics, 2021, pp. 644–657.

[5] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schölkopf, Logical fallacy detection, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), 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.

[6] C. Helwe, T. Calamai, P.-H. Paris, C. Clavel, F. Suchanek, MAFALDA: A benchmark and comprehensive study of fallacy detection and classification, in: K. Duh, H. Gomez, S. Bethard

(Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 4810–4845. URL: https://aclanthology.org/2024.naacl-long.270/. doi:10.18653/v1/2024.naacl-long.270.

[7] S. Haddadan, E. Cabrio, S. Villata, Yes, we can! mining arguments in 50 years of US presidential campaign debates, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4684–4690. URL: https://aclanthology.org/P19-1463/. doi:10.18653/v1/P19-1463.

[8] R. Ruiz-Dolz, J. Lawrence, Detecting argumentative fallacies in the wild: Problems and limitations of large language models, in: M. Alshomary, C.-C. Chen, S. Muresan, J. Park, J. Romberg (Eds.), Proceedings of the 10th Workshop on Argument Mining, Association for Computational Linguistics, Singapore, 2023, pp. 1–10. URL: https://aclanthology.org/2023.argmining-1.1/. doi:10.18653/v1/2023.argmining-1.1.

[9] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, N. A. Smith, Annotation artifacts in natural language inference data, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, 2018, pp. 107–112.

[10] T. Niven, H.-Y. Kao, Probing neural network comprehension of natural language arguments, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, pp. 4658–4664.

[11] R. T. McCoy, E. Pavlick, T. Linzen, Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, pp. 3428–3448.

[12] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F. A. Wichmann, Shortcut learning in deep neural networks, Nature Machine Intelligence 2 (2020) 665–673.

[13] 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).

[14] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, I. Poli, Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2025, pp. 2526–2547.

[15] M. Heinrich, C. Mathews, B. Stein, Touché26-Fallacy-Detection, 2026. URL: https://zenodo.org/records/19925569. doi:10.5281/zenodo.19925569.

[16] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, 2020, pp. 38–45.

[17] A. Grattafiori, et al., The llama 3 herd of models, 2024. arXiv:2407.21783.

[18] Gemma Team, Gemma 3 technical report, 2025. arXiv:2503.19786.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, Journal of Machine Learning Research 12 (2011) 2825–2830.