UZHCL_for_the_win at Touché: Encoder Fine-Tuning, Synthetic Augmentation, and an Underpowered Encoder–LLM Comparison

Touché at CLEF 2026

Stylianos Psychias[1]

[1]University of Zurich, Zurich, Switzerland

Abstract

Online discussions on platforms like Reddit are a productive testbed for fallacy detection: short texts, broad topical range, and the full inventory of informal fallacies catalogued by classical argumentation theory long before social media. The Touché 2026 lab at CLEF formalises this as a shared task on a Reddit-derived corpus and provides a structured benchmark for systems that combine fallacy detection with argument-scheme classification. This paper describes our participation in the lab, covering all three subtasks (ST1 binary fallacy detection, ST2 eight-way fallacy classification, ST3 four-way argument-scheme classification) and both evaluation tracks. Our submitted systems are fine-tuned RoBERTa-large encoders with task-specific heads, augmented with a small batch of LLM-generated synthetic contrastive pairs; for ST2 and ST3 on the enhanced track we additionally extended training with pseudo-labelled examples, extra training items that the encoder labels itself from the published unlabelled pool, kept only where it is most confident and only after a verification check has passed. As an alternative we ran a wider backbone comparison spanning additional encoders (DeBERTa-v3, ModernBERT) and an LLM track (four Qwen variants from 3B to 32B, zero-shot through CoT-LoRA); at this corpus scale none showed a validated benefit over the RoBERTa-large encoder, with 5-fold variance wide enough that we call the comparison underpowered rather than settled. The enhanced track exposes organiser-rewritten input fields (scored separately from the base track); we characterise how much of its lift reflects the rewrite carrying label information rather than genuine argument clarification, and anchor cross-comparisons on base-track or 5-fold CV behaviour. On our held-out development partitions the submitted systems perform strongly on the enhanced and binary cells, with the two base-track classification cells underestimated by their small dev splits. On the official Touché 2026 test set they ranked first on both base-track classification cells (ST2 and ST3) and second on both binary-detection (ST1) tracks, with the two enhanced-track classification cells topped by cascaded and multitask systems. We close on the practical-external rare-class limitation on ST3, the limitations of the present work, and the most useful follow-ups. All code is publicly available at https://github.com/psychias/fallacy_detection_clef_2026.

Keywords

Fallacy detection, Argument scheme classification, Touché 2026, RoBERTa, Synthetic data augmentation, Pseudo-labelling

1. Introduction

Online discussions on platforms like Reddit are a productive testbed for fallacy detection: the texts are short, the topics range across politics, science, and lifestyle, and the rhetorical moves include the full inventory of informal fallacies that classical argumentation theory catalogued long before social media [1, 2]. The Touché 2026 lab at CLEF formalises this as a shared task on a Reddit-derived corpus and provides a structured benchmark for systems that combine fallacy detection with argument-scheme classification [3].

Our submission and this notebook contribute to the lab's overall record as documented in the Touché 2026 overview paper [3] and its extended companion [4].

We participated in all three subtasks of the 2026 edition [3]: binary fallacy detection (ST1), eight-way fallacy classification (ST2), and four-way argument-scheme classification (ST3) organised along a goal axis (practical vs. epistemic) crossed with a basis axis (internal vs. external) [5]. Each subtask is evaluated under two tracks. The *base* track uses only original Reddit fields. The *enhanced* track additionally

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21–24, 2026, Jena, Germany
stylianos.psychias@uzh.ch (S. Psychias)

exposes organiser-supplied paraphrases that were deliberately rewritten to surface fallacy and scheme structure as a designed difficulty setting; the two tracks are scored separately, so the enhanced rewrite is a sanctioned aid, not a contamination of base-track scoring. We submit one run per subtask and track, totalling six TIRA runs [6].

Contributions. The paper makes four contributions, which the methodology and results sections map onto one-to-one:

- **C1** A uniform RoBERTa-large [7] fine-tuning pipeline applied across all six (subtask, track) cells with minimal per-cell variation (§3.1, §3.2).
- **C2** An augmentation study combining LLM-generated synthetic contrastive pairs with a gated pseudo-labelling top-up of the published unlabelled pool, with the negative results (a pseudo-label loss on ST1 enhanced, a rare-class top-up that hurt ST3 base) reported alongside the gains (§3.3, §3.4).
- **C3** A backbone comparison covering DeBERTa-v3-base/large [8], ModernBERT-large [9], and four Qwen variants from 3B to 32B parameters [10, 11] fine-tuned with LoRA / QLoRA [12, 13] and chain-of-thought supervision [14], showing no validated benefit over the encoder at this corpus scale (§3.5, §5.1).
- **C4** A characterisation of two structural features of the 2026 evaluation: what the enhanced-track rewrite actually contributes to a classifier (a string-mask sanity check, a per-class distribution that is consistent with strong label-bearing signal, and a falsifiable follow-up; §5.2), and the practical-external (pe) rare-class limitation that survived targeted augmentation, threshold calibration, and a basis-axis meta-learner (§5.3).

A central posture across all four is careful reading of small-corpus comparisons, separating what dev numbers and 5-fold CV evidence can support from what they cannot. We keep the negative results in the body rather than in footnotes, because the design space they map out is the most useful contribution this notebook can offer a team considering the same shared task next year.

The rest of the paper is organised as follows. Section 2 situates the work inside the fallacy-detection and argument-scheme NLP lineage. Section 3 describes the pipeline, the backbone selection, and the LLM alternative we evaluated. Section 4 reports per-subtask numbers, the augmentation ablation, and the backbone comparison. Section 5 discusses the encoder–LLM comparison, the enhanced-track signal, the pe limitation, the limitations of the present work, and the most useful follow-ups. Section 6 summarises our conclusions.

2. Previous Work

2.1. Fallacy detection in NLP

The Reddit informal-fallacy dataset that the Touché 2026 lab redistributes was introduced by Sahai et al. [1], who labelled a long-tail eight-way taxonomy over comments mined from r/changemyview and adjacent subreddits and reported the first transformer baselines on the resulting corpus. Adjacent benchmarks treat closely related phenomena with similar fine-grained label structures. The Argument Reasoning Comprehension task introduced multi-class warrant identification on user-written argument pairs and surfaced how brittle large language models can be on argumentation tasks that require implicit-premise reasoning [15]. Da San Martino et al. [16] catalogued eighteen propaganda techniques in news articles. Several of those techniques (*appeal to authority*, *appeal to fear*, *black-and-white fallacy*) overlap directly with the ST2 inventory and indicate that the propaganda-detection literature is a useful cross-domain reference even though the source genre differs. Goffredo et al. [17] classified fallacies in political-debate transcripts; their taxonomy differs from the ST2 inventory, but their supervised setting is a close methodological neighbour for the classifier we build.

Two more recent papers anchor the picture. Alhindi et al. [18] reported strong multitask instruction-prompted results across a suite of fallacy-recognition datasets, demonstrating that multitask supervision over related fallacy datasets transfers usefully to eight-way tasks of this kind. MAFALDA [19] is the most-cited 2024 unified fallacy benchmark, we do not retrain on MAFALDA, but our ST2 label inventory is recognisable in its taxonomy and the MAFALDA error-analysis findings on rare-class behaviour echo what we report for the ST3 pe class in §5.3. Pan et al. [20] provide the published precedent for evaluating LLMs as zero-shot fallacy classifiers. They show that zero-shot LLM performance is real but uneven across fallacy types, which motivates the LLM track we describe in §3.5 as worth running rather than dismissing.

2.2. Argumentation schemes

The four ST3 labels are not ad hoc, but neither do they map one-to-one onto individual Walton schemes. The goal (practical / epistemic) × basis (internal / external) crossing is the lab's operationalisation [3] of the argumentation-profile framing of Macagno [5], itself rooted in the broader Walton–Reed–Macagno scheme tradition [2]. The crossing summarises, at a coarser grain than a full scheme catalogue, whether the speaker is arguing about what to do or what to believe, and whether the warrant comes from the speaker's own resources or from an external source. NLP work that classifies arguments by Walton scheme is comparatively thin: Feng and Hirst [21] gave the canonical treatment, assigning arguments to five common schemes, while Jo et al. [22] classify argumentative *relations* using scheme-based logical mechanisms rather than scheme labels. Neither targets the ST3 goal/basis label set directly, so a system designer cannot transfer a published scheme-classification recipe onto this task.

The Touché lab itself has progressively shifted over recent editions from argument retrieval [23, 24] to full argumentation systems [25, 26], with the 2026 edition the first to feature fallacy detection as a dedicated task. The encoder-vs-LLM trade-off for small-corpus fallacy classification is not, to our reading, settled by the published literature: encoder fine-tuning, LoRA-adapted mid-size LLMs, and zero/few-shot prompted frontier models have each been reported as competitive on neighbouring tasks, with no clean recipe-matched comparison at the corpus scale of the present shared task. That open question is the gap our contribution C3 addresses, and it motivates the systematic backbone comparison reported in §4.

3. Methodology

3.1. Pipeline overview

The pipeline has three input streams feeding one fine-tuned encoder: real examples, synthetic contrastive pairs, and a gated pseudo-labelled top-up — a small batch of extra training items that the encoder labels itself from the published unlabelled data, kept only for the examples it is most confident about and only after a verification check has passed. §3.2 fixes the backbone (supporting C1), §3.3–§3.4 build the two augmented streams (supporting C2), and §3.5 reports the LLM alternative we evaluated against this pipeline (supporting C3). Our main focus this year was on building a uniform fine-tuning pipeline that we could apply, with minimal cell-specific variation, across every (subtask, track) combination the lab requires. The Touché 2026 lab [3] (task description and labels at https://touche.webis.de/clef26/touche26-web/fallacy-detection.html) defines three subtasks over the Reddit informal-fallacy data of Sahai et al. [1], with evaluation hosted on TIRA [6]. Subtask ST1 is a binary decision on whether a given comment, presented together with its parent comment and the title of the originating thread, commits any informal fallacy. Subtask ST2 conditions on a positive ST1 decision and asks for the specific fallacy type from an eight-way inventory (using the official label strings from the task schema): authority, black-white, hasty-generalization, natural, population, slippery-slope, tradition, worse-problems. Subtask ST3 conditions on a negative ST1 decision and asks for the argument scheme of the comment from a four-way inventory [2, 5]: practical-internal, practical-external, epistemic-internal, epistemic-external, which we abbreviate pi, pe, ei, ee respec-

tively in the rest of the paper. Evaluation is by macro-averaged F1 in all three subtasks, which puts roughly equal weight on each label and turns rare-class behaviour into a binding constraint on the headline number, a point we return to in §5.3. Each subtask is evaluated in two tracks. The *base* track exposes only original Reddit fields (in our setup, `text_raw`, `text_raw_parent`, `text_raw_title`, `text_base`, `argument_base`). The *enhanced* track additionally exposes two organiser-supplied paraphrases, `text_enhanced` and `argument_enhanced`, that were rewritten with knowledge of the gold labels in order to make the argumentative structure of the comment more explicit, and is scored as a separate track from the base submissions. The rewrite is a designed feature; we defer the characterisation of how much label-bearing signal it carries to §5.2, where the analysis sits in one place. The lab permits up to three runs per (subtask, track), for a maximum of nine submissions. We used six, populating one slot per subtask per track. Appendix E traces which prior work motivated each component of the pipeline described below.

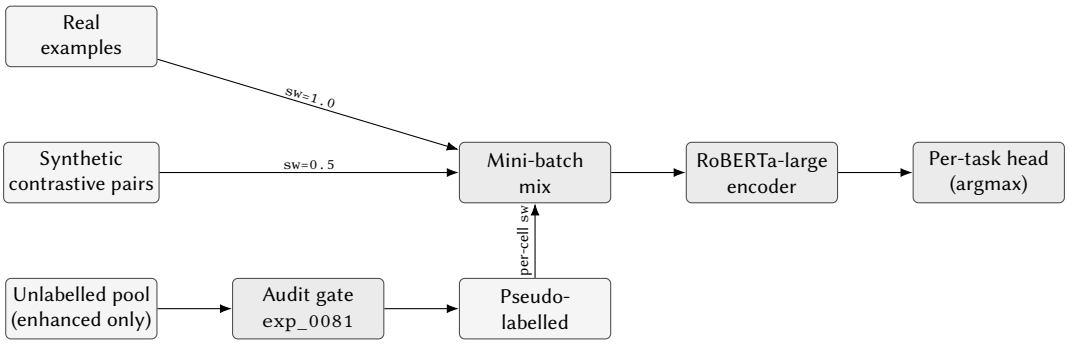

Figure 1: Training and inference pipeline for the submitted RoBERTa-large systems. Real, synthetic, and pseudo-labelled examples are mixed at per-stream loss weights (real at 1.0; the synthetic and pseudo-labelled streams down-weighted per cell, see Appendix B) into a single fine-tuning loop. The audit gate sits between pseudo-label generation and pseudo-label training (§3.4). At inference time the trained classifier head is decoded with per-class thresholds for ST3 and default argmax for ST1/ST2.

The same pipeline produces all six submitted runs, with per-cell hyperparameters summarised in Appendix B. Real examples are concatenated from the published Reddit fields, augmented with a small batch of LLM-generated synthetic contrastive pairs (§3.3), and for ST2 and ST3 enhanced extended with pseudo-labelled unlabelled examples (§3.4) once the dev-overlap audit has passed. The three input streams are interleaved in each training mini-batch under the per-stream loss weights given in the caption. The mixed corpus is fed into a RoBERTa-large encoder [7] with a task-specific classification head. Loss is class-weighted cross-entropy for ST2 and ST3 (with class weights set to inverse-frequency on the training partition) and unweighted cross-entropy for ST1. We use the HuggingFace `transformers` library throughout [27], with the AdamW optimiser, a linear warmup over the first 10% of steps, and a peak learning rate of 1e-5. Inference uses default argmax over class logits for all three subtasks, including the submitted ST3 runs. We additionally explored a per-class threshold sweep on the ST3 development partition (§3.4) as a diagnostic, but did not apply it to the uploaded predictions. The same training loop is used for all six (subtask, track) cells. Only the input fields, the loss weights, the truncation length, and the presence or absence of pseudo-labelled examples vary across cells. Inputs are concatenated using the standard RoBERTa special tokens `<s>...</s></s>...</s>`, with tokeniser truncation set to `max_len=384` for ST3, `max_len=256` for ST1 enhanced and both ST2 cells, and `max_len=512` for the submitted ST1-base run (`exp_0161`), which concatenates `text_raw` with `text_base`. The choice between these windows was driven by a token-length diagnostic showing that ST3 inputs benefited measurably from the longer window once `argument_enhanced` was concatenated, while the shorter ST1-enhanced and ST2 inputs fit comfortably inside 256 with only a small tail of truncations.

3.2. Backbone selection

We compared RoBERTa-base, RoBERTa-large, DeBERTa-v3-base, DeBERTa-v3-large, and ModernBERT-large under matched data and training budgets. RoBERTa-large beat RoBERTa-base by 0.027 on ST1 base, 0.007 on ST1 enhanced, and 0.055 on ST2 base under matched recipes. The one cell where RoBERTa-base was marginally ahead, ST2 enhanced (0.925 vs. 0.923), sits inside seed-variance for these models and we treat it as a tie rather than evidence against the larger backbone. We therefore standardised on RoBERTa-large for the six submitted runs.

DeBERTa-v3-large did not survive the fine-tuning recipes we tried. Plain dev runs collapsed to near-chance F1 across all three subtasks (ST1 base 0.351, ST2 base 0.030, ST3 base 0.205 across two attempts). Mixed-precision training NaN-exploded across four subtask×sw combinations. Switching to bf16 with a smaller learning rate stopped the NaNs but produced near-zero F1. After four NaN-explosion incidents and seven collapse-or-fail runs, we removed DeBERTa-v3-large from the backbone shortlist.

ModernBERT-large [9] ran cleanly but did not beat RoBERTa-large on ST3 enhanced, reaching 0.719 against RoBERTa-large's 0.735. A 0.016 F1 gap on a 4-way task is real but not enough to override the simpler, well-understood RoBERTa pipeline for a notebook-paper submission. For a corpus of roughly 473 ST3 examples, RoBERTa-large is the rational choice. We revisit the LLM alternatives in §3.5 and §5.1.

3.3. Synthetic data augmentation

We generated synthetic contrastive pairs using a frontier LLM via API. The lab data additionally ships a `resembles_fallacy` field that pairs each non-fallacious comment with a structurally similar fallacy class, exactly the near-miss contrast our synthetic pairs are designed to manufacture; we did not consume that field as an explicit training signal, and treating it as a built-in contrastive supervision channel for ST1 is a natural follow-up we leave for a future submission [3]. For each target label, we prompted the model with three real in-domain examples and asked it to produce a paired contrast, a minimally edited rewrite that flips the label while preserving topic, register, and Reddit-conversational style. Generations were filtered by a length sanity check and a string-overlap heuristic against the source pair to discourage trivial copies. Three batches were produced: an initial `synth_v002` (130 pairs for ST1/ST2 and 110 pairs for ST3); a rare-class top-up `synth_v003` for ST3 (100 epistemic-external + 50 practical-external pairs); and a separate pe-targeted batch (100 practical-external + practical-anchor pairs).[1]

Synth at `sw=0.5` added 0.014–0.019 F1 across the four ST1 and ST2 cells (Table 3). On ST3 enhanced, however, synth alone did not move the needle: the real-only baseline and the +synth configuration tied at 0.694, and the eventual gain on this cell only appeared once pseudo-labelling was added (§3.4).

The largest single gain came on ST3 base, where the `synth_v002` recipe at `max_len=384` lifted F1 from 0.436 to 0.523, an improvement of +0.087. Synth is not a free lunch: adding the `synth_v003` rare-class top-up on top of `synth_v002` dropped ST3-base F1 by 0.045 relative to `synth_v002` alone, and a focal-loss recipe [28] lost 0.018 over the same baseline. We read this as confirmation of the general caveat documented by Li et al. [29]: synthetic data improves the average and harms the worst-case, and which side you land on depends on the recipe.

3.4. Pseudo-labelling

For ST2 and ST3 enhanced we additionally added pseudo-labelled examples drawn from the published unlabelled pool. Predictions from the best dev-F1 checkpoint at the matching subtask × track were filtered by confidence at a nominal threshold $\tau = 0.9$ (relaxed for under-populated classes so that rare schemes were not dropped entirely) and added to the training mix as an additional down-weighted stream; the per-cell loss weights are listed in Appendix B.[2] Following the modern transformer-era

[1] The exact prompt template and the generated pairs are released alongside this notebook in the TIRA submission package.

[2] The pseudo-label confidence threshold τ and the source checkpoint are released in the TIRA reproducibility package alongside the prompt template.

treatment by Mukherjee and Awadallah [30] and the original pseudo-label idea of Lee [31], we treated this step with caution: pseudo examples are drawn from the unlabelled pool and could in principle be near-duplicates of dev instances, inflating the dev score, so we ran a dedicated dev-overlap audit (`exp_0081`) before letting them into training. It checked id-level, exact-text, and token-Jaccard overlap between the pseudo-labelled entries and the dev split, and found no id or exact-text overlap and no token-Jaccard pair above 0.5 (maximum 0.30); we proceeded. The recipe paid off most clearly on ST2 enhanced, where adding pseudo lifted F1 from 0.937 to 0.970, a gain of +0.033 over the synth-only baseline. On ST3 enhanced the same recipe at `max_len=384` lifted F1 from 0.694 to 0.735, a gain of +0.041 over the closest matching synth-only baseline. The pseudo recipe at `max_len=256` underperformed its synth-only counterpart on ST1 enhanced (0.906 against 0.914, a loss of 0.008), which is one of the reasons we did not adopt pseudo for ST1 enhanced in the final submission.

A first diagnostic on ST3 used an NLI-style basis probe whose per-class breakdown surfaced a clear weakness: pe F1 was 0.25, well below the other three classes. That number was what motivated the pe-targeted synth batch above. After training the final ST3-enhanced classifier, we ran a per-class threshold sweep on the dev partition as a diagnostic; the best configuration $\{\text{ee} = 0.1, \text{ei} = 0.1, \text{pe} = 0.05, \text{pi} = 0.4\}$ added +0.026 F1 over default argmax on that partition. The submitted ST3 predictions use argmax; we did not apply these thresholds to the uploaded runs. A subsequent attempt to learn a basis-axis meta-learner from LLM signals gave no improvement over the encoder-only weighted average and was not used in the submission. The exact synthetic-pair generation prompts can be seen in Appendix A.

3.5. The LLM alternative

Encoders did not look like the obvious choice at the start of the project. Following the published precedent of Pan et al. [20], whose zero-shot LLM results are reported on different fallacy corpora and are therefore not directly numerically comparable, we ran a zero-shot Qwen2.5-3B baseline on ST3 enhanced, which reached F1 0.47, well below the RoBERTa baselines of §3.2. A kNN few-shot variant retrieving over a `bge-large-en-v1.5` sentence embedding index (1876×1024) recovered some ground at 0.53, but still trailed RoBERTa-large by a wide margin.

We then trained CoT-LoRA recipes [12, 13, 14] on Qwen2.5-7B and Qwen2.5-32B. The 7B 5-fold CV on ST3 enhanced reached 0.85 ± 0.11; the 32B single-run reached 0.84. We also evaluated Qwen3-4B-Thinking [11]: its 5-fold CV reached 0.96 on ST2 enhanced, which is competitive with the submitted encoder, but on ST3 base the same model reached only 0.45, below the 0.52 we already had with RoBERTa-large under the `synth_v002` recipe.

Despite the apparent competitiveness of the larger configurations on individual cells, the CoT-LoRA configurations we evaluated did not show a validated benefit over the encoder within our compute envelope. Our final submission (`exp_0220`) populates all six slots with RoBERTa-large systems.

4. Results

We report both our held-out development results, on which all system-selection decisions were made, and the official Touché 2026 test-set leaderboard results (§4.2, Table 2). Development numbers are macro-averaged F1 on the held-out partition or, where flagged, the mean of a 5-fold cross-validation on that partition; all six submitted runs use recipes fixed before the test labels were released. Unless a number is explicitly labelled a test result, it is a development number.

4.1. Validation Results

Following the official Touché 2026 release, we work from the published labelled pool for each subtask: 938 examples for ST1, the 465-example fallacious subset for ST2, and 473 non-fallacious examples for ST3. From each pool we hold out a stratified, roughly 15% development split (ST1 $n = 139$, ST2 $n = 67$, ST3 $n = 69$) on which the single-run headline numbers in Table 1 are computed, training

on the remainder; the 5-fold cross-validation runs instead cross-validate over the full ST3 pool. This distinction is load-bearing for ST3: the single dev split holds exactly one pe example (Appendix C), so its macro-F1 is dragged down by that one always-hard cell, whereas 5-fold CV distributes pe across folds. This is the main reason the ST3-enhanced CV mean (0.875, §4.6) sits well above the single-split dev and five-seed estimates (0.735 and 0.720 ± 0.015). Class distributions follow the underlying Sahai et al. data and are long-tailed; the ST3 pe class in particular is small enough to be the binding constraint on macro-F1, which we revisit in §5.3.

Table 1 reports precision, recall, and F1-macro on our held-out development partition for the six submitted runs. Each row carries the precision, recall, and F1-macro for one (subtask, track) cell, with macro-averaging over the 8- and 4-class label sets for ST2 and ST3 respectively; the two enhanced-track values should be read in light of the rewriter-signal discussion in §5.2. Across the six cells the easier binary task and the hardest scheme task differ by roughly 0.2 F1 on each track, and the base-to-enhanced gap varies markedly across subtasks.

Table 1
Headline dev results for the six submitted runs. P, R, F1 are macro-averaged for ST2 and ST3. Precision and recall are omitted for ST3 base (exp_0069), whose saved evaluation row has a corrupted precision column; its F1-macro is computed independently and is reliable. The two enhanced-track values should be read with the caveat of §5.2 on the enhanced-track signal.

Subtask	Track	Precision	Recall	F1	exp_id
ST1	base	0.769	0.762	0.761	exp_0161
ST1	enhanced	0.917	0.914	0.914	exp_0035
ST2	base	0.746	0.728	0.730	exp_0032
ST2	enhanced	0.970	0.972	0.970	exp_0072
ST3	base	—	—	0.523	exp_0069
ST3	enhanced	0.732	0.740	0.735	exp_0073

On ST3 we additionally ran 5-fold cross-validation under our submitted recipe (fold seeds 42–46), which gave mean F1 0.875 ± 0.083 (per-fold 0.77–0.99) with the same configuration. The Qwen2.5-7B CoT-LoRA alternative for ST3 enhanced gave 0.851 ± 0.114 over the same fold split. The standard deviation overlaps the RoBERTa CV mean, which is the decision-relevant fact behind the final-submission rationale we discuss below. Of the six numbers, the ST2-enhanced cell is simultaneously the numerically strongest result in this paper and the one in which we have the least interpretive confidence. We address this directly in §5.2.

4.2. Official test-set results

Table 2 reports the official Touché 2026 test-set leaderboard scores for our six submitted runs, together with our placement among participating teams on each (subtask, track) cell. These are the final evaluation numbers; the recipes were fixed on the development evidence above before the test labels were released.

Table 2
Official Touché 2026 test-set results for the six submitted runs (macro-averaged precision, recall, F1). *Rank* is our placement among participating teams on that (subtask, track) cell. Bold marks the two cells we ranked first on.

Subtask	Track	Precision	Recall	F1	Rank
ST1	base	0.770	0.754	0.751	2
ST1	enhanced	0.946	0.945	0.944	2
ST2	base	0.876	0.860	**0.859**	**1**
ST2	enhanced	0.958	0.954	0.954	6
ST3	base	0.710	0.724	**0.716**	**1**
ST3	enhanced	0.716	0.933	0.748	5

Our uniform encoder pipeline ranked *first* on both base-track classification cells — ST2 base (0.859) and ST3 base (0.716) — and *second* on both binary-detection tracks (ST1 base 0.751, ST1 enhanced 0.944). On the two enhanced-track classification cells it placed lower (ST2 enhanced 0.954, sixth; ST3 enhanced 0.748, fifth), where the leading systems were a cascaded pipeline (uzh-tt, 0.991 on ST2 enhanced) and a multitask DeBERTa (sevitha, 0.976 on ST3 enhanced).

Two patterns are worth drawing out, both specific to the classification cells (ST2, ST3); the binary ST1 cells ranked second on both tracks and moved little between dev and test (0.761 vs. 0.751 base, 0.914 vs. 0.944 enhanced). First, the two base-track classification cells came in well above their development estimates: ST3 base rose from a dev 0.523 to a test 0.716, and ST2 base from 0.730 to 0.859, while the enhanced-track classification cells tracked their dev numbers closely (ST3 enhanced 0.735 dev vs. 0.748 test; ST2 enhanced 0.970 vs. 0.954). This is consistent with the small held-out dev splits (§4.1) underestimating base-track classification performance rather than with any test-time surprise. Second, the single-encoder recipe was most competitive on the base-track classification cells, where the enhanced rewrite is unavailable, while the enhanced-track classification cells were topped by systems built to exploit the rewritten fields, by cascading or multitasking over them; this is the same label-bearing signal we characterise in §5.2, seen from the leaderboard side.

4.3. Binary Fallacy Detection (ST1)

The binary task was the easiest of the three. The submitted ST1-base run (`exp_0161`) concatenates `text_raw` with `text_base` at `max_len=512` and reached F1 0.761 on the dev partition. The synthetic-data contribution is isolated separately in Table 3 at `max_len=256`, where `+synth_v002` at `sw=0.5` lifts F1 from a real-only baseline of 0.716 to 0.734 (`exp_0033`). On the enhanced track, the same synth recipe reached 0.914 against a real-only enhanced baseline of 0.899, with the +synth lift contributing +0.015 over that baseline. A pseudo-labelling extension at `max_len=256` and `sw=0.5` underperformed its synth-only counterpart by 0.008 (Table 3, row −0.008), which is the reason ST1 enhanced ships under synth only rather than synth + pseudo. The 0.15 base-to-enhanced gap on ST1 previews the enhanced-track signal discussion of §5.2.

4.4. Eight-Way Fallacy Classification (ST2)

Fallacy classification (ST2) is comparably easy on the enhanced track (0.97) and substantially harder on the base track (0.73), the largest base-to-enhanced gap in Table 1. On the base track the submitted +synth configuration reached 0.730 over a real-only baseline of 0.711 (+0.019). On the enhanced track, the +synth configuration alone reached 0.937 against a real-only enhanced baseline of 0.923 (+0.014), and the +synth+pseudo recipe at `max_len=384` and `sw=0.3` added another +0.033 on top of that to reach 0.970. Per-class precision and recall for ST2-enhanced are reported in Appendix C; on the $n = 67$ dev split the eight classes range from 0.889 to 1.000 F1, with five of the eight at 1.000 — a high, tight distribution consistent with the rewriter carrying strong label-bearing signal as discussed in §5.2.

4.5. Four-Way Argument Schemes (ST3)

Argument-scheme classification (ST3) was the hardest on both tracks, at 0.523 base and 0.735 enhanced; the four-way scheme task absorbed the bulk of our experimental effort and is the cell where the per-class behaviour discussed in §5.3 matters most. On the base track, the `+synth_v002` recipe at `max_len=384` lifted F1 from a real-only baseline of 0.436 to 0.523 (+0.087, the largest single ablation gain in the paper). On the enhanced track, the +synth+pseudo recipe at the same window (with the augmented streams left at full weight, `sw=1.0`) reached 0.735 against a real-only enhanced baseline of 0.694 (+0.041); +synth alone tied the real-only baseline on this cell, so the gain only appeared once pseudo-labelling was added. Two ST3-base recipes lost ground in the ablation: the `synth_v002 + v003` stack at `max_len=384` dropped to 0.478 (−0.045 against `synth_v002` alone), and the focal-loss variant reached 0.505 (−0.018 against the same baseline).

Table 3
Per-subtask augmentation ablation, F1-macro on dev. Δ is the change relative to the immediately preceding row within the same (subtask, track) block. The shorthand `@N` denotes `max_len=N` (tokeniser truncation length). Negative deltas are not hidden.[a]Single-seed unless noted; see §5.4 for the seed-variance caveat.

Subtask	Track	Recipe	exp_id	F1	Δ
ST1	base	real only	exp_0013	0.716	—
ST1	base	+ synth (sw=0.5)	exp_0033	0.734	+0.018
ST1	enhanced	real only	exp_0022	0.899	—
ST1	enhanced	+ synth (sw=0.5)	exp_0035	0.914	+0.015
ST1	enhanced	+ synth + pseudo @256 (sw=0.5)	exp_0071	0.906	−0.008
ST2	base	real only	exp_0014	0.711	—
ST2	base	+ synth (sw=0.5)	exp_0032	0.730	+0.019
ST2	enhanced	real only	exp_0023	0.923	—
ST2	enhanced	+ synth (sw=0.5)	exp_0036	0.937	+0.014
ST2	enhanced	+ synth + pseudo @384 (sw=0.3)	exp_0072	0.970	+0.033
ST3	base	real only	exp_0058	0.436	—
ST3	base	+ synth_v002 @384	exp_0069	0.523	+0.087
ST3	base	+ synth_v002 + v003 @384	exp_0068	0.478	−0.045
ST3	base	+ synth_v002 + focal loss	exp_0053	0.505	−0.018
ST3	enhanced	real only	exp_0024	0.694	—
ST3	enhanced	+ synth @384	exp_0067	0.694	0.000
ST3	enhanced	+ synth + pseudo @384 (sw=1.0)	exp_0073	0.735	+0.041

Augmentation strategies are recipe-sensitive: synth alone gives small gains on ST1/ST2 (the smallest within the measured seed variance) and the largest single gain on ST3 base, while pseudo-labelling unlocks the enhanced track for ST2 and ST3 but costs on ST1 enhanced under the pseudo recipe. The `synth_v003` top-up on ST3 base and the focal-loss recipe on ST3 base illustrate the same lesson at smaller scale. We keep the losses in the same table rather than relegating them to a footnote.

Table 4 reports the matched ST3-enhanced backbone comparison, and Table 5 the off-track context rows (DeBERTa on ST3 base, and the Qwen3-4B-Thinking comparator on ST2 enhanced and ST3 base) that are *not* directly comparable to it. Each row is a single dev-partition F1 unless flagged as a 5-fold mean. The cross-row variability is wide enough that the comparison should be read alongside the cross-validation evidence in the prose below, not as a clean ranking.

Table 4
Matched backbone comparison on ST3 enhanced. Single-run dev F1 unless marked as a 5-fold cross-validation mean±sd. The submitted row is marked † (its 5-fold CV mean is given alongside the single-run number).

Backbone	exp_id	F1 (ST3 enh)
RoBERTa-base	exp_0018	0.675
RoBERTa-large†	exp_0073	0.735 (CV 0.875 ± 0.083)
ModernBERT-large	exp_0086	0.719
Qwen2.5-3B zero-shot	exp_0091	0.467
Qwen2.5-3B kNN few-shot	exp_0143	0.529
Qwen2.5-7B CoT-LoRA (5-fold CV)	exp_0180	0.851 ± 0.114
Qwen2.5-32B CoT-LoRA (single)	exp_0207	0.841

Within the matched ST3-enhanced comparison (Table 4), RoBERTa-large is the submitted system and RoBERTa-base sits 0.06 F1 below it, which is the empirical justification for paying the larger backbone's compute cost. ModernBERT-large at 0.719 is the next-best encoder route but loses 0.016 F1 on this cell. The Qwen rows tell their own story: the 3B zero-shot configuration trails RoBERTa-large by

Table 5
Off-track context rows, reported for completeness and *not* comparable to the ST3-enhanced column of Table 4: DeBERTa on ST3 base (the large collapsed under NaN, the base merely underperformed; §3.2) and the Qwen3-4B-Thinking comparator on ST2 enhanced and ST3 base.

Backbone	Subtask	Track	F1
DeBERTa-v3-base	ST3	base	0.277
DeBERTa-v3-large (collapsed)	ST3	base	0.205
Qwen3-4B-Thinking (5-fold CV)	ST2	enh	0.963
Qwen3-4B-Thinking (5-fold CV)	ST3	base	0.448

0.27 F1, and kNN few-shot recovers some ground but still trails by 0.21. The fine-tuned 7B and 32B CoT-LoRA configurations close most of the gap on a single run, but under cross-validation the encoder and the LLM are not separable: the encoder's 5-fold mean of 0.875 ± 0.083 and the Qwen2.5-7B mean of 0.851 ± 0.114 have heavily overlapping standard-deviation bands, so the comparison is statistically inconclusive at our sample size. The off-track rows (Table 5) are context only: DeBERTa-v3 did not train usefully under any recipe we tried (0.277 on ST3 base, and 0.205 with the NaN-explosion history we detail in §3.2), so its low numbers reflect a training failure rather than a track-matched loss, while the Qwen3-4B-Thinking comparator is competitive on ST2 enhanced under cross-validation (0.963) but below the RoBERTa baseline on ST3 base (0.448), which complicates a clean encoder-beats-LLM headline.

The per-class F1 distribution for the submitted ST3-enhanced system is reported in Appendix C (ee 1.000, ei 0.979, pi 0.963, pe 0.000), computed under argmax on the $n = 69$ dev split, and discussed in §5.3. The pe cell rests on a single dev example and is therefore uninformative; we rely on the 5-fold cross-validation (§5.3) for any pe characterisation. The submitted run's dev logits were not checkpointed, so its exact confusion matrix cannot be recovered. We have since added dev-logit checkpointing to the training script and re-ran the `exp_0073` recipe under five seeds on the same fixed dev split, both to characterise run-to-run variance and to recover a representative confusion matrix (Table 8). Across the five re-runs dev macro-F1 is 0.720 ± 0.015 (range 0.700–0.741, all fifteen epochs completed), which places the submitted 0.735 near the top of the observed range and is consistent with the single-run caveat of §5.4. The per-class pattern is stable across seeds: ei (0.98–1.00) and pi (0.963 in every run) are recovered almost perfectly, the variance concentrates in the seven-example ee class (0.86–1.00), and pe has recall 0 in all five re-runs. We turn to the interpretation of the enhanced-track numbers and the rare-class behaviour we observed on ST3 in §5.

4.6. Final submission rationale

We did not use the additional two-runs-per-track budget available under the Touché submission rules. We report on the single CV-validated configuration per slot. The non-obvious choice was ST3 enhanced, where the Qwen2.5-32B single-run F1 of 0.841 sat above the RoBERTa-large dev number of 0.735 but just below the RoBERTa-large 5-fold CV mean of 0.875. Two reasons drove the encoder choice, and we keep them separate. The first is operational: the Qwen2.5-32B 5-fold cross-validation was still running at the deadline, leaving the RoBERTa-large CV as the only completed cross-validation we had in hand at submission time. The second is the absence of a statistical reason to prefer Qwen even if the comparison had been completed: a 5-fold standard deviation of ± 0.114 on Qwen2.5-7B is evidence the encoder-vs-LLM comparison is underpowered at this sample size, not evidence for the system whose CV mean is higher. We therefore do not claim the available CV evidence rules out the Qwen alternatives; we only claim that at deadline the encoder pipeline was the only fully-validated configuration. A post-hoc ensemble of the available encoder and LLM systems gave no gain over the single-source encoder, which is one further reason we did not retroactively complicate the submission. A submission dry-run verified that the TIRA-format outputs matched the lab's expected schema before the final upload.

5. Discussion and Future Work

This year we reflect on the structural shape of the Touché 2026 evaluation and on what our submissions can and cannot say about it. The first three subsections below close the four contributions stated in §1: §5.1 closes C3 (encoder vs. LLM at this scale), §5.2 closes the first half of C4 (the enhanced-track signal), and §5.3 closes the second half of C4 (the pe rare-class limitation). §5.4 states the limitations attached to the numbers behind those subsections, and §5.5 lists the most useful follow-ups the present work allows.

5.1. Encoder vs. LLM at this scale

Across the cells we were able to validate under cross-validation, encoder fine-tuning matched or edged out every LLM track we evaluated on point estimates. The most informative comparison is on ST3 enhanced, where the encoder's 5-fold CV mean of 0.875 ± 0.083 and the Qwen2.5-7B CoT-LoRA comparator's 0.851 ± 0.114 have standard-deviation bands that overlap almost entirely. The single-run Qwen2.5-32B reached 0.841 on the same task. With dispersion that wide on both sides, the apparent 0.024 gap is not statistically meaningful from a single 5-fold split. The strongest claim the evidence supports is that the LLM track did not show a validated benefit at this dataset scale within our compute envelope, not that LLMs are categorically worse. We scope the claim to this dataset (roughly 473 ST3 examples), this label structure, and our compute budget; we would not extrapolate to larger labelled corpora or to richer reasoning targets. The test outcome is consistent with this reading: the encoder pipeline ranked first on both base-track classification cells (ST2 and ST3, §4.2), where no rewritten fields are available, while the enhanced-track cells were topped by cascaded and multitask systems built to exploit the rewrite — an architecture axis, not a backbone-family verdict.

5.2. What the enhanced-track signal contains

The 0.970 ST2-enhanced and 0.914 ST1-enhanced numbers are the strongest in the paper and warrant a careful read of what the rewritten fields contribute. The lab provides the enhanced track as a designed difficulty setting: `text_enhanced` and `argument_enhanced` are organiser-supplied paraphrases produced with knowledge of the gold labels, and base-only systems are scored in a separate track [3]. The legitimate scientific question is therefore not whether something illicit happened (it did not) but how much of the enhanced-track lift reflects the rewrite carrying label-bearing signal versus genuine argument clarification. We address the question in four steps.

(a) The setup. Any classifier trained on `text_enhanced` or `argument_enhanced` has access to features that a base-only pipeline would have to recover from raw text. The interesting question is not whether such features exist (they are designed in) but whether the classifier is exploiting them as label strings or as cleaner argumentative structure.

(b) A sanity check, and what it does not establish. We ran a regex-mask sanity check that replaced known fallacy-name strings and obvious cue words in the enhanced fields and re-trained the ST2-enhanced classifier on the masked inputs. F1 was retained at 0.937 after masking, against the unmasked enhanced-track *baseline* (no pseudo-label) of 0.937. The mask was applied to this synth-only configuration, not to the submitted `+synth+pseudo` system (0.970), so it does not bear on any signal the pseudo-label stage might add. The mask deliberately covers the eight ST2 fallacy-class names and their morphological variants only, not the full semantic content of the rewritten fields. A null delta is therefore the expected outcome *if* the classifier was not relying on those literal strings as a shortcut, and is the specific hypothesis the check rules out, not a general claim that the rewritten fields contain no label-bearing signal at all. We call this a sanity check rather than a diagnostic for exactly this reason: it tests a hypothesis the rewriter's design made a priori unlikely, and therefore says little about the softer semantic channel.

(c) The interpretation. A retained F1 of 0.937 under aggressive string masking means that trivial string-matching of fallacy labels in `text_enhanced` is not the dominant signal the classifier relies

on. That rules out the most superficial story, but the rewriter could still encode the label in word choice, sentence structure, hedging, or topical framing without ever using a fallacy-name string. We note that the ST2-enhanced per-class F1 values (Appendix C) range from 0.889 to 1.000 on the $n = 67$ dev split, with five of the eight classes at 1.000 — a high, tight distribution still consistent with strong label-bearing signal in the rewritten fields, though the wider spread than we first reported tempers that reading slightly.

(d) The reporting commitment. We present base-track and enhanced-track numbers separately throughout this notebook and do not aggregate them into a single headline; where the paper takes a position e.g., on the ablation pattern, on the encoder vs. LLM comparison, on the pe limitation, we anchor that position on base-track behaviour or 5-fold CV numbers rather than on enhanced-track point estimates. The headline enhanced-track numbers stand as ceiling results for the designed track they were produced under.

The dev-overlap audit of §3.4 addresses a different worry: that the pseudo-labelled examples, drawn from the unlabelled pool, might be near-duplicates of dev instances and so inflate the dev score directly; it found no such overlap. It does not, however, speak to whether the rewritten fields carry label-bearing signal — that is what the masking check above probes, and neither check clears the softer semantic channel. A falsifiable extension, training a classifier on base-only fields and evaluating it on enhanced fields, would isolate the marginal contribution of the rewriter directly; we flag it in §5.5.

5.3. The pe rare-class limitation

The practical-external (pe) class on ST3 was the hardest scheme class in every evaluation that contained enough pe examples to measure it. An NLI-style basis probe flagged it early. We responded with three interventions: a pe-targeted synthetic batch, a per-class threshold sweep that pushed the pe threshold to 0.05, and an LLM basis-axis meta-learner; none produced a measurable lift on the dev split, which holds too few pe examples to detect one.

Quantifying the gap is constrained by how few pe examples the dev partition holds. Our single held-out dev split contains exactly one pe example, so the submitted-run pe numbers — F1 $= 0.000$ under argmax (`exp_0073`) and 0.25 under the dev-only threshold sweep (`exp_0145`), both computed on that same one-example split — reflect a single instance rather than a distribution and cannot support a ceiling estimate. The only evaluation with meaningful pe support is the 5-fold cross-validation under the submitted recipe (`exp_0153`), where pe-F1 averages 0.713 across folds against a 0.875 macro-F1, but with wide fold-to-fold variance (per-fold 0.40–1.00, standard deviation 0.25). Even there, pe is consistently the class that most depresses macro-F1, and the variance underlines how few pe instances the corpus provides; the thin pe representation in the dev partition is itself a limitation of the evaluation (§5.4). The substantive finding stands: pe is the rare class that the targeted synthetic augmentation, the per-class threshold sweep, and the basis-axis meta-learner all failed to lift, and targeted pe-class data collection (§5.5) is the intervention most likely to help. A five-seed re-run of the submitted recipe (Table 8, §4.5) reinforces the point from the other direction: ei and pi are recovered almost perfectly in every run while the single pe dev example is misclassified in all five, so its dev F1 of 0 reflects the split's inability to measure the class rather than a stable property of the system.

5.4. Limitations

Several limitations should accompany the numbers in this notebook.

System selection was made entirely on our held-out dev partition and 5-fold cross-validation; the official test results (Table 2, §4.2) became available only afterwards and informed no decision in this paper. Where a single 5-fold split was available (ST3 enhanced under our submitted recipe, the Qwen2.5-7B CoT-LoRA comparator, and the Qwen3-4B-Thinking comparators) we use the CV mean as the decision-relevant estimate; the other dev numbers in Tables 1–4 are single-run point estimates, and the ranking of close-together rows should be read with appropriate caution.

We ran a five-seed variance check only on the submitted ST3-enhanced cell (§4.5, dev macro-F1 0.720 ± 0.015); we did not repeat it for the other five cells, whose reported single-run dev numbers should be read as point estimates. For runs whose individual seeds are not recorded, we used the lab-default `seed=42`; CV runs used fold seeds 42–46.

The enhanced-track headline numbers are upper bounds on what the designed track can yield, in the sense developed in §5.2. The regex-mask sanity check rules out the trivial-string channel; it does not clear the softer semantic channel, and we have flagged this everywhere the enhanced numbers appear.

The per-class F1 breakdown in Appendix C is computed under argmax decoding for both ST2 and ST3, matching the decoding used for the submitted runs and the headline numbers in Table 1. The ST3 pe cell in particular rests on a single dev example and should not be read as a per-class estimate; the 5-fold cross-validation (§5.3) is the only evaluation with meaningful pe support.

The per-class analysis in this version of the notebook (Appendix C and §5.3) differs from the initially submitted version. The submitted version reported a per-class breakdown whose values and supports did not trace to any saved evaluation artefact; during code–paper auditing we reconciled the analysis against the actually-submitted checkpoints (`exp_0072`, `exp_0073`) and now report the reproducible argmax numbers on the real dev partitions ($n = 67$ and $n = 69$).

Overall, these limitations do not invalidate the above numbers, but the enhanced-track results in particular should be read as upper bounds on what a fair pipeline would extract from the same inputs.

5.5. Future work

The most useful follow-ups the present work allows are concrete. A falsifiable extension of the leakage diagnostic, training a classifier on base-only fields and evaluating it on enhanced fields, would isolate the marginal contribution of the rewriter and let us put a number on the residual semantic-leakage channel; we did not run this within our compute envelope and flag it as the most useful follow-up the C4 thread allows. A leakage-controlled re-evaluation on a future Touché edition, one that ships a separate decoder for the rewritten fields, or that withholds the label-aware rewrite from training entirely, would let us decide more cleanly between the encoder and LLM tracks at this dataset scale. A larger compute budget on the Qwen2.5-32B fold split would let us close the comparison we left underpowered, addressing the open question behind C3. A per-class threshold sweep on ST3 base, paralleling the sweep we ran on ST3 enhanced, would let us test whether the pe limitation we observed is a property of the enhanced track or of the four-way label structure more broadly. And targeted pe-class data collection, rather than further synthetic augmentation, looks like the most useful intervention if the rare-class limitation is to be overcome at all.

6. Conclusion

We close on the four contributions in the same order they appear in §1. **C1:** A single RoBERTa-large fine-tuning pipeline, with only per-cell variation in input fields and loss weights, populated all six (subtask, track) cells, ranking first on the official test set for both base-track classification cells (ST2 and ST3) and second on both binary-detection tracks (§4.2). **C2:** Synthetic contrastive augmentation gave small gains (+0.014 to +0.087 across cells, the smallest of which fall within the ± 0.015 seed variance we measured), and pseudo-labelling unlocked the enhanced track for ST2 and ST3, but a pseudo recipe lost ground on ST1 enhanced and a rare-class synthetic top-up lost ground on ST3 base; we reported both kinds of result in the same tables. **C3:** A grid covering DeBERTa-v3, ModernBERT, and four Qwen variants up to 32B parameters showed no validated benefit over the encoder at this corpus scale; the 5-fold standard deviation on the strongest LLM comparator leaves that comparison underpowered rather than closed. **C4:** The enhanced-track headline numbers should be read as upper bounds because the rewritten fields carry signal that our string-mask sanity check could not rule out at the semantic level, and the practical-external (pe) ST3 class continued to limit macro-F1 after targeted augmentation, threshold calibration, and a basis-axis meta-learner. With its 2026 addition of fallacy detection, Touché

gives argument-aware NLP a benchmark we expect to return to, and the negative results collected here are offered to save the next team the detours we took.

Acknowledgments

The author thanks the Touché 2026 lab organizers for providing the dataset, evaluation infrastructure, and submission platform.

AI Usage Declaration

This text was proofread and improved with the assistance of Claude (Anthropic). The following prompt was used:

> "Imagine you're an AI master student. Evaluate my text based on the task description and check if it correct or not, if not tell me which parts I should improve."

Claude was used to evaluate the text against the Touché 2026 Fallacy Detection task description, identify missing elements (results coverage, leakage handling, methodology and ablation details) and suggest phrasing improvements. The author conceived and executed all experiments, made all submission decisions, and takes full responsibility for the publication's content.

References

[1] 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, 2021, pp. 644–657. URL: https://aclanthology.org/2021.acl-long.53. doi:10.18653/v1/2021.acl-long.53.

[2] D. Walton, C. Reed, F. Macagno, Argumentation Schemes, Cambridge University Press, 2008. doi:10.1017/CBO9780511802034.

[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, 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.

[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 — 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.

[5] 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.

[6] 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: Advances in Information Retrieval (ECIR 2023), volume 13981 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.

[7] 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, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[8] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: Proceedings of the 11th International Conference on Learning Representations (ICLR), 2023. URL: https://arxiv.org/abs/2111.09543. arXiv:2111.09543.

[9] 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, 2024. URL: https://arxiv.org/abs/2412.13663. arXiv:2412.13663.

[10] Qwen Team, Qwen2.5 technical report, 2024. URL: https://arxiv.org/abs/2412.15115. arXiv:2412.15115.

[11] Qwen Team, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: Proceedings of the 10th International Conference on Learning Representations (ICLR), 2022. URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.

[13] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, in: Advances in Neural Information Processing Systems (NeurIPS), 2023. URL: https://arxiv.org/abs/2305.14314. arXiv:2305.14314.

[14] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: Advances in Neural Information Processing Systems (NeurIPS), 2022. URL: https://arxiv.org/abs/2201.11903. arXiv:2201.11903.

[15] I. Habernal, H. Wachsmuth, I. Gurevych, B. Stein, The argument reasoning comprehension task: Identification and reconstruction of implicit warrants, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Association for Computational Linguistics, 2018, pp. 1930–1940. URL: https://aclanthology.org/N18-1175. doi:10.18653/v1/N18-1175.

[16] G. Da San Martino, S. Yu, A. Barrón-Cedeño, R. Petrov, P. Nakov, Fine-grained analysis of propaganda in news articles, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019, pp. 5636–5646. URL: https://aclanthology.org/D19-1565. doi:10.18653/v1/D19-1565.

[17] P. Goffredo, S. Haddadan, V. Vorakitphan, E. Cabrio, S. Villata, Fallacious argument classification in political debates, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI), 2022, pp. 4143–4149. URL: https://www.ijcai.org/proceedings/2022/575. doi:10.24963/ijcai.2022/575.

[18] T. Alhindi, T. Chakrabarty, E. Musi, S. Muresan, Multitask instruction-based prompting for fallacy recognition, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2022, pp. 8172–8187. URL: https://aclanthology.org/2022.emnlp-main.560. doi:10.18653/v1/2022.emnlp-main.560. arXiv:2301.09992.

[19] C. Helwe, T. Calamai, P.-H. Paris, C. Clavel, F. Suchanek, MAFALDA: A benchmark and comprehensive study of fallacy detection and classification, in: 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, 2024, pp. 4810–4845. URL: https://aclanthology.org/2024.naacl-long.270. doi:10.18653/v1/2024.naacl-long.270.

[20] 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), Association for Computational Linguistics, 2024. URL: https://aclanthology.org/2024.emnlp-main.794.

[21] V. W. Feng, G. Hirst, Classifying arguments by scheme, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011, pp. 987–996. URL: https://aclanthology.org/P11-1099/.

[22] 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. doi:10.1162/tacl_a_00394. arXiv:2105.07571.

[23] A. Bondarenko, M. Fröbe, J. Kiesel, S. Syed, T. Gurcke, M. Beloucif, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2022: Argument retrieval, in: Working Notes Papers of the CLEF 2022 Evaluation Labs, volume 3180 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3180/paper-247.pdf.

[24] A. Bondarenko, M. Fröbe, J. Kiesel, S. Syed, T. Gurcke, M. Stahl, P. Anh Vu, M. P. Carnot, L. Heinrich, N. Merker, A. Schmidt, M. Wolska, M. Beloucif, J. Bevendorff, C. Biemann, A. Panchenko, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2023: Argument and causal retrieval, in: Working Notes Papers of the CLEF 2023 Evaluation Labs, volume 3497 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 3065–3089. URL: https://ceur-ws.org/Vol-3497/paper-259.pdf.

[25] J. Kiesel, M. Fröbe, M. Gohsen, M. Heinrich, M. Hagen, B. Stein, M. Potthast, Overview of Touché 2024: Argumentation systems, in: Working Notes Papers of the CLEF 2024 Evaluation Labs, volume 3740 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 3341–3366. URL: https://ceur-ws.org/Vol-3740/paper-322.pdf.

[26] J. Kiesel, Ç. Çöltekin, M. Gohsen, S. Heineking, M. Heinrich, M. Fröbe, T. Hagen, M. Aliannejadi, S. Anand, T. Erjavec, M. Hagen, M. Kopp, N. Ljubešić, K. Meden, N. Mirzakhmedova, V. Morkevičius, H. Scells, M. Wolter, I. Zelch, M. Potthast, B. Stein, Overview of Touché 2025: Argumentation systems, in: Working Notes Papers of the CLEF 2025 Evaluation Labs, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 4536–4562. URL: https://ceur-ws.org/Vol-4038/paper_378.pdf.

[27] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. URL: https://arxiv.org/abs/1708.02002. arXiv:1708.02002.

[29] Z. Li, H. Zhu, Z. Lu, M. Yin, Synthetic data generation with large language models for text classification: Potential and limitations, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2023, pp. 10443–10461. URL: https://aclanthology.org/2023.emnlp-main.647. doi:10.18653/v1/2023.emnlp-main.647.

[30] S. Mukherjee, A. H. Awadallah, Uncertainty-aware self-training for text classification with few labels, in: Advances in Neural Information Processing Systems (NeurIPS), 2020. URL: https://arxiv.org/abs/2006.15315. arXiv:2006.15315.

[31] D.-H. Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: ICML 2013 Workshop on Challenges in Representation Learning (WREPL), 2013. Semantic Scholar paperId: 798d9840d2439a0e5d47bcf5d164aa46d5e7dc26. No DOI or arXiv id exists for this workshop manuscript.

A. Prompts

Synthetic contrastive pairs and pseudo-labelled examples contributed to the submitted systems (§3.3, §3.4). The system prompts that drove each generation batch are reproduced verbatim below; the matching user prompts are built per call from a topic seed, a subreddit, a small block of real exemplars from the train split, a sampled target length and number of supports, and (for ST1/ST2) a fallacy card or (for ST3) a scheme pair. Generated batches are released alongside this notebook in the TIRA submission package.

Prompt 1: ST1/ST2 contrastive-pair generation. System message.

You generate training data for a fallacy detection model. The data must read as authentic Reddit replies: informal register, hedges (“idk”, “tbh”, “iirc”), occasional typos, no debate-club phrasing, no signnposting. Never name the fallacy inside the generated text.
The ‘load-bearing signals’ you’ll be shown describe the LOGICAL PATTERN of the fallacy, not phrases to use verbatim. Express the same reasoning move with fresh wording. Do NOT use any of these stock formulations (fresh wording only): ‘millions of people can’t be wrong’, ‘people can’t be wrong’, ‘everyone knows’, ‘no middle ground’, ‘no in between’, ‘before you know it’, ‘next thing you know’, ‘where does it end’, ‘it’s a slippery slope’, ‘we’ve always done it’, ‘natural is better’, ‘chemicals are bad’, ‘yeah i agree’, ‘i agree it’s’. If your draft contains any of these phrases, rewrite the draft.
You will produce a CONTRASTIVE PAIR: one reply that commits the target fallacy, and one reply using structurally similar reasoning that does NOT commit it. Both replies must address the same topic with claims of similar polarity, so the only meaningful difference is the soundness of the inference.
CRITICAL: Do NOT cite specific percentages, year-tagged studies (e.g. ‘2023 study’), polls, named institutions (Gallup, Pew, CDC, etc.), or made-up statistics in EITHER reply. Both replies must reason from lived experience, anecdote, observation, or general claims. The difference between them is the soundness of the inference, not the presence of evidence.

The user prompt supplies a fallacy card (definition, load-bearing signals, legitimate counterpart), the topic and subreddit, a target length and per-reply support count, and a small block of real exemplars. The model returns a JSON object carrying `thread_title`, `parent`, and two replies (`fallacious` and `legitimate`), each with `text_raw`, `text_base`, `claim`, and `supports`.

Prompt 2 has the same shape as Prompt 1, but produces two non-fallacious replies using different argumentation schemes.

Prompt 2: ST3 contrastive-pair generation (`synth_v002` ST3). System message.

You generate training data for an argumentation scheme classifier. The data must read as authentic Reddit replies: informal register, hedges, occasional typos, natural conversation. No academic phrasing.
[blocklist as in Prompt 1]
You will produce a CONTRASTIVE PAIR: two non-fallacious replies to the same parent comment, on the same topic, with similar conclusions – but using DIFFERENT argumentation schemes. The structural difference should be clear but natural.

The user prompt specifies Scheme A (target) and Scheme B (contrast), each with goal axis (practical / epistemic), basis axis (internal / external), definition, and load-bearing signals, together with the topic, subreddit, target length, and exemplar block. JSON output is `scheme_a` / `scheme_b`, each with `scheme`, `argument_goal`, `argument_basis`, `text_raw`, `text_base`, `claim`, and `supports`.

A separate, simpler prompt generated the pe-targeted batch (50 examples per label, `openai/gpt-4o-mini`, temperature 0.9).

Prompt 3: pe-targeted generation (`exp_0103`). System message.

You are an expert in argumentation theory and propaganda techniques. Generate realistic argumentative text snippets that contain propaganda or emotional manipulation techniques (loaded language, fear appeals, bandwagon, false dilemma, etc.) rather than logical reasoning.
The text should be 2-4 sentences, resemble real-world political discourse or persuasive media, and

clearly exemplify propaganda / emotional fallacies.

Two user prompts, one per generated label:

Prompt 3: pe-targeted generation (`exp_0103`). User prompts.

[pe] Generate a short argumentative text (2-4 sentences) that uses propaganda or emotional manipulation. Include at least one of: loaded language, appeal to fear, bandwagon appeal, false dilemma, or emotional blackmail. Make it sound like real political/media discourse.
[pa] Generate a short argumentative text (2-4 sentences) that primarily appeals to emotion (pathos) rather than logic. Use emotional language, fear, pity, or desire to persuade. Make it realistic.

Pseudo-label confidence selection (`exp_0072`, `exp_0073`). The pseudo-labelling step filtered the encoder classifier's own predictions at a nominal $\tau = 0.9$, relaxed for under-populated classes so that rare schemes were retained (see §3.4); it is not LLM-API driven and uses no prompt. We list it here for completeness rather than because there is a prompt to record.

B. Hyperparameter Grid

Table 6 summarises the configurations used for the six submitted runs. All runs share the RoBERTa-large backbone [7] and use the `transformers` library [27]. Optimiser is AdamW with linear warmup over the first 10% of steps. Loss is class-weighted cross-entropy for ST2 and ST3; ST1 uses unweighted cross-entropy. Where the experiment log does not separately record a seed, we used the lab default `seed=42`; 5-fold CV runs used fold seeds 42–46.

Table 6
Submitted-run hyperparameter grid. `sw` is the loss weight applied to synthetic (and pseudo-labelled) examples relative to real examples. `max_len` is the tokeniser truncation length.

Subtask	Track	exp_id	max_len	batch	lr	epochs	extras
ST1	base	exp_0161	512	8	1e-5	12	synth_v002 sw=0.5, raw_concat
ST1	enhanced	exp_0035	256	8	1e-5	12	synth_v002 sw=0.5
ST2	base	exp_0032	256	8	1e-5	15	synth_v002 sw=0.5
ST2	enhanced	exp_0072	384	8	1e-5	12	synth + pseudo sw=0.3
ST3	base	exp_0069	384	8	1e-5	15	synth_v002 sw=0.5
ST3	enhanced	exp_0073	384	8	1e-5	15	synth + pseudo sw=1.0

Synthetic generation used a frontier LLM via API (see §3); the prompt template and the generated pairs are released alongside this notebook in the TIRA submission package. Per-class decision thresholds for ST3 were swept on the dev partition; the best configuration was $\{\text{ee}{=}0.1, \text{ei}{=}0.1, \text{pe}{=}0.05, \text{pi}{=}0.4\}$ (`exp_0145`, $+0.026$ F1 over default argmax on the dev partition; this sweep was a dev-partition diagnostic and was not applied to the submitted predictions, which use argmax).

C. Per-Class F1 on the Dev Partition

Table 7 reports per-class precision, recall, F1, and support for the submitted enhanced-track systems on the held-out development partition. Both ST2 (`exp_0072`) and ST3 (`exp_0073`) values are computed from default argmax predictions — the decoding actually used for the submitted runs — and are read directly from each checkpoint's saved metrics ($n = 67$ for the ST2 dev split, $n = 69$ for ST3). Both macro-F1 values reconcile with Table 1 (§4) within ± 0.001 rounding. This breakdown was reconciled

against the submitted checkpoints during code–paper auditing, replacing an earlier table whose per-class values and supports did not correspond to any saved evaluation artefact.

Table 7
Per-class precision, recall, F1, and support on the dev partition for ST2-enhanced (`exp_0072`, $n = 67$) and ST3-enhanced (`exp_0073`, $n = 69$), both under argmax decoding. Values are read from the submitted checkpoints' saved metrics. The ST3 pe cell rests on a single dev example and is not an informative per-class estimate (see §5.3).

Label	Precision	Recall	F1	Support
ST2 (8-way; exp_0072; argmax; $n = 67$)				
authority	1.000	1.000	1.000	8
black-white	1.000	1.000	1.000	8
hasty-generalization	1.000	1.000	1.000	8
natural	1.000	0.889	0.941	9
population	0.875	1.000	0.933	7
slippery-slope	1.000	1.000	1.000	9
tradition	0.889	0.889	0.889	9
worse-problems	1.000	1.000	1.000	9
ST3 (4-way; exp_0073; argmax; $n = 69$)				
ee (epistemic-external)	1.000	1.000	1.000	7
ei (epistemic-internal)	1.000	0.958	0.979	48
pi (practical-internal)	0.929	1.000	0.963	13
pe (practical-external)	0.000	0.000	0.000	1

D. ST3-Enhanced Confusion Matrix

Table 8 gives a representative dev confusion matrix for the ST3-enhanced system. The submitted run's logits were not checkpointed, so its exact matrix is unrecoverable; the training script now saves per-example dev logits (`dev_logits.npz`), and we re-ran the `exp_0073` recipe under five seeds on the same fixed dev split (§4.5). The matrix shown is the run nearest the five-seed mean (macro-F1 0.722). The qualitative structure is stable across seeds: ei and pi are recovered almost perfectly, the residual errors sit in the small ee class, and the single pe example is misclassified in every run (§5.3).

Table 8
Representative dev confusion matrix for ST3-enhanced (`exp_0073` recipe, seed nearest the five-seed mean; macro-F1 0.722, $n = 69$). Rows are the gold label, columns the argmax prediction. Reconstructed from the released `dev_logits.npz`.

	Predicted			
True	ee	ei	pi	pe
ee (epistemic-external)	6	0	0	1
ei (epistemic-internal)	0	48	0	0
pi (practical-internal)	0	0	13	0
pe (practical-external)	0	0	1	0

E. Citation Graph

Figure 2 traces the prior work that fed each component of our pipeline. The horizontal axis is left-to-right: citations on the left, the pipeline stages they motivated on the right. Solid arrows mark routes that ended up in the submitted system; dashed arrows mark routes we tried and abandoned (the

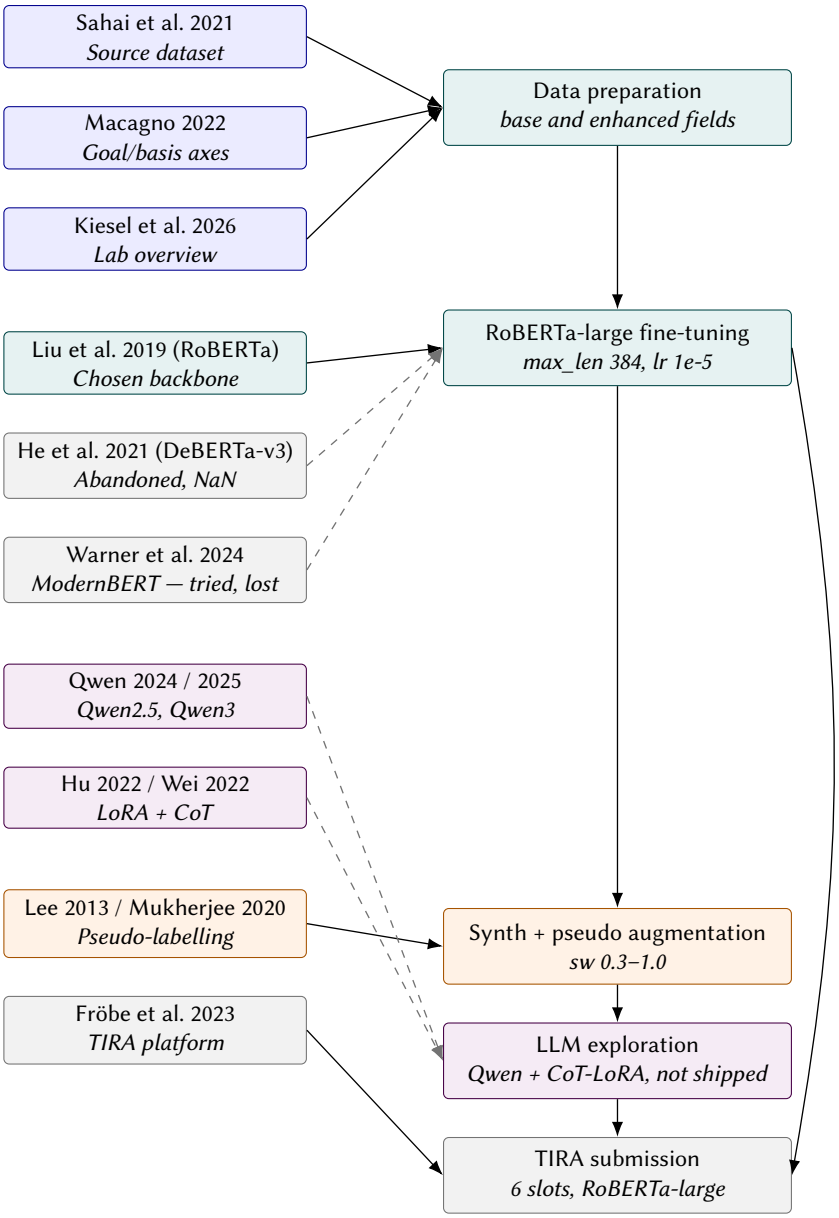

Figure 2: Which prior work fed which component of our pipeline. Solid arrows are routes that shipped; dashed arrows are routes we tried and dropped (DeBERTa-v3-large under NaN, ModernBERT-large underperforming RoBERTa-large, and the LLM alternative that did not out-perform RoBERTa-large under cross-validation within our compute envelope).

DeBERTa-v3-large NaN failures, ModernBERT-large underperformance, and the LLM alternative that did not validate under CV).