arginvariant at Touché: Error-Driven Prompt Refinement for Argument Identification

Touché at CLEF 2026

Midhun Kanadan[1,*], Maximilian Heinrich[1] and Benno Stein[1]

[1]Bauhaus-Universität Weimar, Weimar, Germany

Abstract

Argument mining aims to automatically identify and analyze argumentative structures in natural language text, with argument identification (determining whether a sentence expresses an argument at all) as its most fundamental subtask. Whether a sentence counts as an argument, however, is not a property of the sentence in isolation: it depends on the annotation guideline applied. The Touché 2026 shared task on *Generalizability of Argument Identification in Context* makes this dependency its primary evaluation target: the same tweets are annotated under three different annotation guidelines, and a system that ignores the active guideline will produce the same prediction regardless of which definition applies. The *arginvariant* system described in this paper frames this as a guideline-following problem: the expert annotation criteria for each corpus are distilled into a natural-language prompt, and a large language model applies those criteria at inference time without any parameter updates. Thirteen dataset-specific prompts are constructed through iterative error-driven refinement: common failure patterns identified across multiple LLMs on the training set drive inner-loop prompt updates, and development-set Macro F_1 serves as the outer-loop stopping criterion. The best submitted run achieves a mean Macro F_1 of 0.7834 on the primary evaluation set, placing second on the official leaderboard.

Keywords

Argument Identification, Iterative Prompt Refinement, Annotation Guideline Encoding, Cross-Domain Generalization, Large Language Models

1. Introduction

Argumentation is central to human reasoning and communication. From political debates and courtroom proceedings to scientific peer review and online discussion forums, the ability to identify, construct, and evaluate arguments underlies informed decision-making and critical thinking. As the volume of text-based discourse has grown, the automatic analysis of argumentative content has become increasingly relevant for applications such as debate assistance, fact-checking, opinion mining, and educational tools. This has driven a growing body of research in argument mining, the automatic identification and analysis of argumentative structures in natural language text [1, 2].

Its most fundamental subtask, argument identification, asks whether a given sentence expresses an argument at all. The task appears deceptively simple but conceals a deep difficulty in cross-domain settings: different corpora operationalize "argument" incompatibly, ranging from any expression of inference or opinion to strictly claim-plus-evidence structures. A classifier trained on one corpus's labels learns one specific definition, and that learned boundary becomes incorrect whenever the guideline changes.

The root cause is that whether a sentence counts as an argument depends not only on its content but on the annotation guideline applied to it. Feger et al. [3] demonstrated this failure at scale across 17 argument mining benchmarks, showing that state-of-the-art models exploit corpus-specific surface shortcuts rather than genuine argumentative structure. Supervised training teaches a model one fixed definition and provides no mechanism for switching to another.

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

*Corresponding author.

midhun.kanadan@uni-weimar.de (M. Kanadan); maximilian.heinrich@uni-weimar.de (M. Heinrich); benno.stein@uni-weimar.de (B. Stein)

0009-0008-7030-9878 (M. Kanadan); 0000-0001-5450-8203 (M. Heinrich); 0000-0001-9033-2217 (B. Stein)

The Touché 2026 shared task on *Generalizability of Argument Identification in Context* [4, 5][1] makes this failure mode the primary evaluation target. Its official ranking metric is computed over tweets annotated under three incompatible guidelines (TACO, TAPE, TAUS). Any system that ignores the active guideline will assign the same label to the same tweet under all three definitions, forfeiting any advantage from guideline-specific criteria.

This paper addresses this challenge by treating argument identification as a *guideline-following problem*. Rather than training on argument labels, the expert annotation criteria for each dataset are distilled into a natural-language prompt; a large language model then acts as an annotator who has read those criteria and applies them at inference time. Prompts are refined iteratively by analyzing shared misclassifications across multiple LLMs on the training set and adding explicit rules that address recurring error patterns. Per-dataset model selection is performed by comparing development Macro F_1 across a pool of five open-source models. The best submitted run places second on the official leaderboard with Macro F_1 = 0.7834.

The remainder of this paper is organized as follows. Section 2 reviews related work on argument mining generalization and LLM prompting. Section 3 introduces the task, datasets, and evaluation metric. Section 4 describes the system. Section 5 presents development results and the effect of iterative prompt refinement. Section 6 describes the submitted runs and reports official test-set scores. Section 7 concludes.

2. Related Work

Argument mining covers the automatic identification, segmentation, and classification of argumentative discourse [1, 2]. Argument identification, the binary decision of whether a text unit expresses an argument, is typically modeled as sentence-level classification. Early approaches relied on engineered features; more recent work fine-tunes pre-trained transformer encoders such as BERT [6] and RoBERTa [7] on corpus-specific labeled data.

The core difficulty is that argument corpora use incompatible annotation schemes, so models trained on one corpus do not transfer reliably to another. Feger et al. [3] systematically quantified this problem across 17 argument mining benchmarks, finding that 97% of cross-dataset generalization experiments fell below the mean benchmark result, with 62% scoring below 0.65. Crucially, removing discourse markers and function words changed Macro F_1 by less than 0.02 for standard transformer baselines (BERT, DistilBERT), suggesting that these models learn dataset-specific surface patterns rather than genuine argumentative structure. The best-performing model in that study, WRAP [8], adds contrastive pre-training on Twitter argument pairs and achieves a mean leave-one-out Macro F_1 of 0.66 across 17 benchmarks [3] (the strongest cross-dataset result among the transformer models evaluated), yet still substantially below within-dataset performance.

Large language models with instruction tuning [9, 10] can follow task specifications expressed as natural language without parameter updates [11]. Gilardi et al. [12] show that instruction-prompted LLMs match or exceed crowd-worker agreement on classification tasks, motivating the use of annotation criteria as an explicit classification signal.

Large language models have recently been applied to argument mining subtasks such as claim and evidence detection via prompting and in-context learning [13]. Conditioning classifiers on natural-language label descriptions has similarly been explored in zero-shot text classification [14] and structured prediction [15]. What remains unexplored is conditioning argument identification on multiple, incompatible annotation guidelines applied to the same text: prior work assumes a single fixed label definition, whereas the setting addressed here requires switching between annotation frameworks for identical underlying sentences.

[1] https://touche.webis.de/clef26/touche26-web/generalizable-argument-mining.html

3. Task and Dataset

The Touché 2026 shared task on *Generalizability of Argument Identification in Context* [4, 5] frames argument identification as a cross-domain generalization challenge. Given a sentence together with metadata about its provenance (source document and corpus annotation guidelines), a system must predict `Argument` or `No-Argument`.

The shared task draws on ten publicly available benchmark corpora spanning biomedical (ABSTRCT [16]), scientific (ARGUMINSCI [17], SCIARK [18]), financial (FINARG [19]), political (USELEC [20]), and online-discussion (AEC [21], AFS [22]) domains, together with open-domain product and technology comparisons (ACQUA [23]), heterogeneous multi-topic web text (IAM [24]), and persuasive student essays (PE [25]). Each corpus contributes 1,700 labeled sentences partitioned 60/20/20 into train, development, and test splits, all perfectly balanced at 50% Argument.

For primary evaluation, the shared task introduces three evaluation-only variants derived from TACO [26], a Twitter debate corpus: TACO (original inference vs. information labels), TAPE (same sentences re-annotated under Persuasive Essay guidelines), and TAUS (re-annotated under US presidential debate guidelines). All three apply different annotation criteria to the same 340 tweets, yielding 1,020 labeled instances in total. No training or development labels are provided for any of the three variants.

Systems are ranked by the unweighted mean Macro F_1 over the three Twitter variants (TACO, TAPE, TAUS), referred to as the primary metric throughout this paper. Per-corpus Macro F_1 on the ten benchmark test splits is reported as a supplementary metric. Runs are submitted through TIRA [27], the official evaluation platform of the shared task.

4. Approach

The goal of the *arginvariant* system is to construct a corpus-specific classification prompt that encodes each corpus's annotation logic. For each corpus, the annotation criteria are distilled into an initial prompt, which is then iteratively refined using training-set error patterns as a feedback signal; the resulting prompt is applied at inference time by a large language model without any parameter updates (Figure 1). Running refinement across multiple LLMs simultaneously ensures that only error patterns shared across model families drive prompt updates, making the resulting prompts robust regardless of which model is used at inference time.

4.1. Dataset-Specific Guideline Distillation

Thirteen prompts are created, one for each corpus in the evaluation (ten benchmark datasets plus TACO, TAPE, and TAUS). For each corpus, the annotation guidelines and the originating research paper are read to derive an understanding of what constitutes an argument under that corpus's definition, which is then distilled into an initial classification prompt. All prompts share the same output instruction: the model must respond with exactly `Argument` or `No-Argument`, with no explanation and no other text. The full set of prompts used in the system is available at https://github.com/Midhun-Kanadan/arginvariant-touche2026. The repository includes all prompt versions for each corpus, from v1 through the final refined version, to support reproducibility and comparison of individual refinement steps.

4.2. Iterative Error-Driven Prompt Refinement

The annotation guidelines provided for each corpus are a necessary but incomplete specification. Human annotators applying the same guidelines develop implicit shared conventions: boundary decisions, edge-case resolutions, and domain-specific exclusions that are never written down but consistently reflected in the gold labels. A prompt derived solely from the written guideline will therefore exhibit systematic errors that reveal these unwritten conventions. The refinement procedure treats those errors as a learning signal and uses them to recover the implicit annotation patterns.

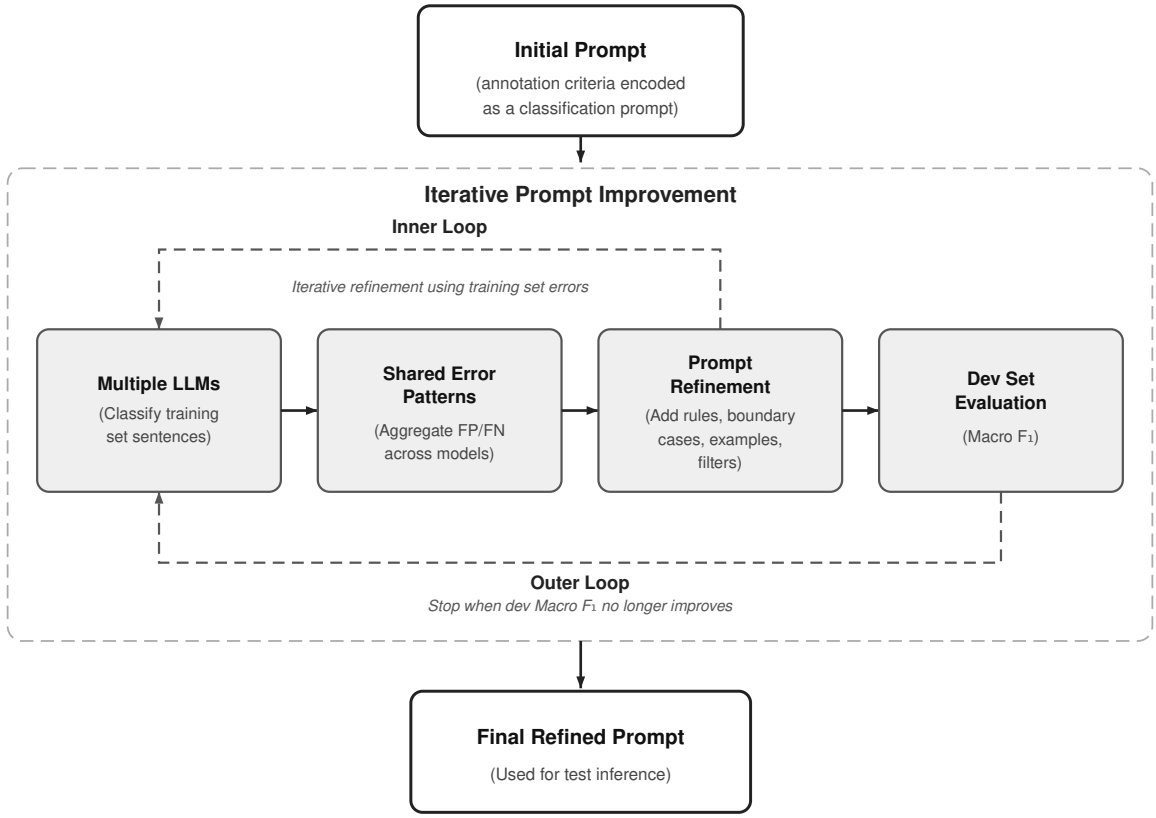

Figure 1: The *arginvariant* error-driven prompt refinement pipeline. Annotation criteria for each corpus are distilled into an initial classification prompt. The inner loop runs the prompt across multiple LLMs on the training set, aggregates error patterns shared across models, and updates the prompt with targeted rules. The outer loop validates each revised prompt on the development set and continues refinement while Macro F_1 improves, stopping when no further gain is observed. The final refined prompt is used directly for test inference without any parameter updates.

For the ten benchmark corpora, where training and development labels are available, refinement proceeds as two nested loops.

Inner loop: multi-LLM error aggregation on the training set. The current prompt version is run on the training split across multiple LLMs simultaneously. For each LLM, all misclassifications are identified by comparing predicted labels against gold labels, and false positives (predicted `Argument`, true `No-Argument`) and false negatives (predicted `No-Argument`, true `Argument`) are recorded. Errors are then aggregated across models: only patterns that appear consistently across multiple LLMs are treated as prompt deficiencies. This aggregation provides a natural filter: an error made by one LLM but not others is likely a model-specific instruction-following limitation rather than a gap in the prompt, while an error that appears across all or most models indicates that the prompt itself fails to encode the relevant decision criterion. Because manually reviewing complete misclassification batches is time-consuming, identifying the shared error pattern within each batch was assisted by an LLM coding agent (Claude Code), which proposed candidate patterns that we then reviewed; translating an accepted pattern into a concrete prompt rule remained a manual step. Only the shared errors are used to update the prompt, adding one targeted rule per identified shared pattern, after which the inner loop repeats. Each refinement iteration adds targeted constraints, decision rules, or illustrative examples to address the identified shared error patterns.

Because refinement targets only errors common across models, the resulting prompt does not overfit to any single LLM and generalizes across model families.

Outer loop: dev-set validation. After the inner loop stabilizes (that is, after no new shared

error patterns are identified on the training set), the prompt is evaluated on the development split. Development Macro F_1 serves as the held-out validation signal. If performance improves relative to the previous version, the inner loop resumes with the updated prompt. If no improvement is observed, the prompt is finalized. This outer loop prevents over-refinement: a prompt that perfectly fits training-set error patterns but fails to generalize would be penalized by the development metric.

Prompts evolved from basic guideline definitions toward more structured architectures. For most corpora, this meant layering explicit negative exclusion criteria onto a positive core definition: IAM's final prompt makes this rejection-filter structure explicit through seven enumerated exclusion patterns layered onto the positive four-criterion CDC test. AEC is the clear exception: its dominant over-prediction pattern reflected an implicit surface convention in the gold labels rather than an overly broad semantic definition, for which a positive rule matching that convention proved more effective than any rejection filter (Section 5.3).

For the three Twitter evaluation corpora, where no labeled data is available, a different procedure applies. For TAPE and TAUS, the annotation guidelines are identical to those of two benchmark corpora: TAPE re-annotates the TACO tweets under the Persuasive Essay guidelines of Stab and Gurevych [25], the same guidelines under which PE was annotated, and TAUS re-annotates them under the US presidential debate guidelines of Haddadan et al. [20], the same guidelines under which USELEC was annotated [5]. Their prompts are therefore derived from the already-refined PE and USELEC prompts respectively, adapted only for tweet-specific surface characteristics such as hashtag usage as a standalone claim marker, informal register, and sentence brevity. For TACO, whose annotation framework has no benchmark counterpart, the prompt is derived from the TACO paper's inference-vs.-information definition [26] and the provided guideline. Without gold labels, error-driven refinement is replaced by a consistency-based procedure: inter-LLM disagreements on the same tweet identify ambiguous decision rules, and the prompt is revised to resolve them.

4.3. Models for Refinement and Evaluation

Five open-source LLMs are used across the prompt refinement experiments and development evaluation:

- `deepseek-r1:14b` (DeepSeek) [28]
- `gemma2:27b` (Google) [29]
- `gemma3:27b` (Google) [30]
- `gemma4:26b` (Google) [31]
- `mistral:7b` (Mistral AI) [32]

All models are served via Ollama[2] on a university cluster at temperature 0.0 to promote reproducibility and support reliable cross-model error pattern aggregation in the refinement loop.

5. Results and Analysis

This section reports development results on the ten benchmark corpora and the effect of iterative prompt refinement.

5.1. Development Results

Table 1 reports development Macro F_1 per corpus for each model under the final prompt version. The mean across ten benchmark corpora is 0.802. The three strongest corpora are AEC (0.977), ABSTRCT (0.888), and ACQUA (0.835). ABSTRCT and ACQUA have well-defined, narrow argument definitions that map cleanly to prompt rules; AEC's high score instead reflects an implicit surface convention in its annotations rather than a semantic definition (Section 5.3). The three weakest are USELEC (0.692), FINARG (0.735), and PE (0.728), reflecting the difficulty of distilling nuanced political

[2] https://www.ollama.com/

Table 1
Development set results for each corpus. Each row reports the best-performing model–prompt combination found during development, selected by Macro F_1. *Arg* and *No-Arg* are the per-class F_1 scores; both should be close to *Macro* F_1 for a well-calibrated classifier. *Mean* is the unweighted mean Macro F_1 over all ten benchmark corpora.

Corpus	Model	Macro F_1	Arg F_1	No-Arg F_1
ABSTRCT	`gemma3:27b`	0.888	0.888	0.888
ACQUA	`gemma4:26b`	0.835	0.856	0.830
AEC	`gemma4:26b`	0.977	0.976	0.977
AFS	`gemma3:27b`	0.800	0.793	0.807
ARGUMINSCI	`gemma4:26b`	0.784	0.767	0.801
FINARG	`gemma2:27b`	0.735	0.727	0.743
IAM	`deepseek-r1:14b`	0.779	0.780	0.779
PE	`gemma3:27b`	0.728	0.773	0.683
SCIARK	`gemma3:27b`	0.800	0.808	0.791
USELEC	`mistral:7b`	0.692	0.664	0.719
Mean (10 corpora)	—	0.802	—	—

Table 2
Effect of iterative prompt refinement on development Macro F_1. *Macro* F_1 *(v1)* is the score of the initial prompt derived directly from annotation guidelines; *Macro* F_1 *(best)* is the score of the best prompt version found through iterative refinement. *Version* is the version number of the best prompt (refinement starts from v1, so this also indicates how many iterations were needed). Δ is the absolute gain. For each corpus, both scores use the same model (the one selected as best on the development set), isolating the effect of prompt refinement from model selection.

Corpus	Model	Macro F_1 (v1)	Macro F_1 (best)	Version	Δ
ABSTRCT	`gemma3:27b`	0.728	0.888	v5	+0.160
ACQUA	`gemma4:26b`	0.650	0.835	v5	+0.185
AEC	`gemma4:26b`	0.419	0.977	v5	+0.558
AFS	`gemma3:27b`	0.553	0.800	v4	+0.247
ARGUMINSCI	`gemma4:26b`	0.660	0.784	v3	+0.124
FINARG	`gemma2:27b`	0.612	0.735	v6	+0.123
IAM	`deepseek-r1:14b`	0.678	0.779	v5	+0.101
PE	`gemma3:27b`	0.581	0.728	v6	+0.147
SCIARK	`gemma3:27b`	0.615	0.800	v6	+0.185
USELEC	`mistral:7b`	0.548	0.692	v5	+0.144
Mean	—	0.604	0.802	—	+0.198

discourse criteria [20], the claim/premise distinction [25], and domain-specific financial framing [19] into a prompt-based classifier. No single model dominates: `gemma3:27b` leads on four corpora (ABSTRCT, AFS, PE, SCIARK), `gemma4:26b` on three (ACQUA, AEC, ARGUMINSCI), and `deepseek-r1:14b`, `gemma2:27b`, and `mistral:7b` each on one corpus.

5.2. Effect of Iterative Prompt Refinement

Table 2 shows the Macro F_1 of the initial prompt (v1) and the final refined prompt for each corpus. For each corpus, both scores are obtained with the same model (the one selected as best on the development set), so the gain reflects prompt improvement alone. Refinement improves Macro F_1 on every corpus without exception, with a mean gain of 0.198 points across the ten benchmarks.

The largest absolute gain is on AEC (+0.558): the initial prompt achieved a Macro F_1 of 0.419, below the 0.5 expected from a random classifier on a balanced dataset, because its semantic test for argumentative content severely over-predicted `Argument`. As Section 5.3 details, the corpus's gold

Table 3
Dominant v1 error pattern, a representative example, and the rule that corrected it, for the three corpora with the largest refinement gains. All examples are exact development-split sentences, verified as false positives shared across at least two models at v1 and corrected in the corpus's final prompt version.

Corpus	Error type	Example sentence	Gold	v1 pred	Rule added
AEC	Semantic content mistaken for stance-taking; true signal is a sentence-initial connective, not content	"Sadly, creationism is an assertion of dogmatism."	No-Arg	Arg	Classify by discourse-connective opener alone (*so, but, if, first, however*, …): present → `Argument`, absent → `No-Argument`, regardless of content.
AFS	Reporting a ruling, event, or finding mistaken for taking a position on it	"Now the courts have held in some instances, the right to keep and bear arms is an individual right and in some instances, it is a states right."	No-Arg	Arg	Describing what courts ruled, what history shows, or what a study found, without the sentence itself taking a position, is `No-Argument` (*describe* vs. *argue*).
ACQUA	Comparative language mistaken for a stable directional argument	"The whole Florida line didn't play well versus missouri, but was better against georgia."	No-Arg	Arg	An argument requires two specifically named items with one direction holding across the whole sentence; comparisons that split by domain or lack a named second item are `No-Argument`.

labels instead track a surface convention—the presence of a discourse-connective opener—which the final prompt encodes directly. AFS (+0.247) and ACQUA (+0.185) show similarly large gains, each correcting `Argument` over-prediction through its own targeted rules (Section 5.3). At the other end, IAM (+0.101) and ARGUMINSCI (+0.124) show smaller but consistent improvements: both corpora have inherently difficult multi-criterion definitions for which additional rules yield incremental rather than dramatic gains. Most prompts stabilized between v3 and v5; FINARG, PE, and SCIARK required the full six iterations, while ARGUMINSCI stabilized earliest at v3.

5.3. Qualitative Error Analysis

This subsection examines the v1 error patterns behind the three largest refinement gains, AEC (+0.558), AFS (+0.247), and ACQUA (+0.185), and the rules that corrected them. All sentences and counts below are exact, drawn from development-split predictions; each pattern is a false positive shared across at least two independently evaluated models at v1 and corrected in the corpus's final prompt version. Table 3 summarizes one representative example per corpus.

AEC shows the most unusual failure mode of the three. Its v1 prompt asks whether a sentence makes a substantive point or takes a side, a semantically reasonable test that nonetheless produces massive over-prediction of `Argument`: 146 of the 340 development sentences are false positives shared across two independently evaluated models, and every one of them fails to open with a discourse connective. Conversely, all eight shared false negatives open with one, including short or rhetorical sentences such as "So what do you end up with?" that carry little semantic content of their own. This indicates that AEC's gold annotations follow an implicit markup convention: the discourse-connective opener, not the semantic content of the sentence, is the operative annotation signal. The final AEC prompt (v5) abandons semantic reasoning in favor of a direct connective-detection rule, reaching development Macro F_1 of 0.977 (test: 0.959). This result complicates the framing of prompt refinement as teaching a model to recognize genuine argumentative structure: for AEC, the most effective prompt matches the same kind of superficial surface signal that Feger et al. [3] identify as a shortcut in fine-tuned encoder models, rather than avoiding it.

AFS illustrates a more conventional failure. v1's decision rule, whether the sentence makes a claim or provides reasoning for one side, over-triggers on sentences that report a legal ruling, a historical event, or a study finding without the sentence itself taking a position, such as the example in Table 3. The corpus's guideline distinguishes *describing* a state of affairs from *arguing* for a position, a distinction

Table 4
Per-dataset model assignment for Run 1 (primary submission). Development Macro F_1 guided model selection for the ten benchmark corpora; proxy evaluation on PE and USELEC development sets guided selection for TACO, TAPE, and TAUS. All models served via Ollama at temperature 0.0.

Corpus / Group	Assigned model
ABSTRCT, AFS, PE, SCIARK	`gemma3:27b`
ACQUA, AEC, ARGUMINSCI	`gemma4:26b`
FINARG	`gemma2:27b`
IAM	`deepseek-r1:14b`
USELEC	`mistral:7b`
TACO, TAPE, TAUS	`gemma2:27b`

the initial prompt does not encode. The final prompt (v4) makes this distinction explicit, alongside a rule excluding sentences whose reference depends on unnamed external context ("the study," "they"), reaching development Macro F_1 of 0.800 (test: 0.747), up from 0.553. Appendix A traces the intermediate refinement steps for this corpus in detail.

ACQUA over-predicts `Argument` on comparative-sounding sentences that do not assert a single stable direction, such as the example in Table 3 (worse against one named opponent, better against another) and sentences comparing against an unnamed baseline (e.g., a company relying on a service "for faster …outsourcing" with no second party named). The final prompt requires a comparison between two specifically named items with one direction holding across the whole sentence, reaching development Macro F_1 of 0.835 (test: 0.853). Not every case is resolved: a rhetorical question comparing two named institutions remains misclassified by the selected model, indicating that the assertive-question exception is harder to encode reliably than the other criteria.

Across all three corpora, the dominant v1 failure is `Argument` over-prediction, but the corrective mechanism differs by corpus. AEC's fix is a positive trigger rule matching an implicit surface convention; AFS and ACQUA are corrected primarily by explicit negative criteria that narrow an over-broad positive definition. This suggests that the effective form of a refinement rule, a surface-matching heuristic versus a rejection filter, depends on whether a corpus's implicit annotation convention itself tracks a surface signal or a genuine semantic distinction. Errors that persisted after refinement, such as those remaining in USELEC and FINARG, involved nuanced political discourse or domain-specific framing that resisted rule-based encoding, suggesting a ceiling on how much implicit annotator judgment a finite set of written rules can capture.

6. Submissions and Official Test Results

Three runs were submitted through TIRA [27], all using the final refined prompts and differing only in model assignment. **Run 1** assigns each of the ten benchmark corpora the model with the highest development Macro F_1 from the evaluation pool; `gemma2:27b` is used for TACO, TAPE, and TAUS. **Run 2** applies `gemma2:27b` uniformly across all corpora. **Run 3** applies `gemma3:27b` uniformly across all corpora. Table 4 shows the full per-corpus model assignment for Run 1.

Table 5 reports the official blind-test scores. The *Main* score is the primary ranking metric (unweighted mean Macro F_1 over TACO, TAPE, and TAUS); *Supplementary* is the unweighted mean Macro F_1 over the ten benchmark corpora.

Run 1 achieves the highest Main score of 0.7834, placing second on the official leaderboard. Run 2 obtains an almost identical Main score of 0.7829 (a difference of 0.05 percentage points (pp) from Run 1) but substantially lower supplementary performance (0.7533 vs. 0.7899, a gap of 3.7 pp), indicating that per-corpus model selection primarily benefits in-domain benchmark generalization rather than the out-of-domain primary metric. The leading submission (`context-awakens`) scores 0.7955 on the Main metric, a gap of 1.2 pp; notably, the TAPE score (0.7757) exceeds theirs (0.7604) by 1.5 pp, with the overall gap attributable to TACO (0.8408 vs. 0.8133) and TAUS (0.7853 vs. 0.7612). Run 3 ranks sixth overall with

Table 5
Official Touché 2026 blind-test results for the three submitted runs. All values are Macro F_1. TACO, TAPE, and TAUS are the primary out-of-domain Twitter variants; *Main* is their unweighted mean and the official ranking metric. *Suppl.* is the mean Macro F_1 over the ten benchmark corpora. Bold marks the best value per column among the three runs.

Run	TACO	TAPE	TAUS	Main	Suppl.
Run 1 (per-corpus model)	**0.8133**	**0.7757**	**0.7612**	**0.7834**	**0.7899**
Run 2 (`gemma2:27b`, uniform)	**0.8133**	**0.7757**	0.7598	0.7829	0.7533
Run 3 (`gemma3:27b`, uniform)	0.8101	0.7204	0.7377	0.7561	0.6531
`context-awakens` (1st)	0.8408	0.7604	0.7853	0.7955	0.7308

Table 6
Comparison of the final refined prompt against two closed-source baselines, GPT-5 nano and Gemini 3.5 Flash, each applied with a guideline-only prompt (v1) and with the refined prompt, on the primary evaluation corpora. *Refined prompt, best open-source model* reproduces the Run 1 test-set Macro F_1 from Table 5. Bold marks the best value per row.

	Refined prompt,	GPT-5 nano		Gemini 3.5 Flash	
Corpus	best open-source model	v1	refined	v1	refined
TACO	0.8133	0.7463	0.8235	0.8088	**0.8466**
TAPE	**0.7757**	0.6264	0.6679	0.7482	0.7498
TAUS	0.7612	0.7056	0.7470	0.7785	**0.7870**
Mean	0.7834	0.6928	0.7461	0.7785	**0.7945**

a Main score of 0.7561; the lower TAPE score (0.7204 vs. 0.7757 for Runs 1–2) suggests that `gemma3:27b` is less effective than `gemma2:27b` at applying the Persuasive Essay annotation framework to tweets.

The supplementary score of 0.7899 substantially exceeds that of the leading submission (0.7308 for `context-awakens`), confirming that the *arginvariant* approach generalizes strongly across all thirteen corpora and not only on the held-out Twitter variants.

To assess whether the gains from iterative refinement are specific to the open-source models used or reflect genuine prompt improvement, Table 6 applies two closed-source models, GPT-5 nano [33] and Gemini 3.5 Flash [34], with both the guideline-only initial prompt (v1) and the final refined prompt, alongside our best open-source configuration. These results were obtained by running inference locally on the released test-set labels and were not submitted through TIRA; they are therefore not part of the official leaderboard evaluation and are reported for analysis only.

Refinement transfers across model families: replacing the guideline-only prompt with the refined prompt raises the mean Macro F_1 of both closed-source models (GPT-5 nano from 0.6928 to 0.7461, Gemini 3.5 Flash from 0.7785 to 0.7945), confirming that the improvement reflects genuine prompt quality rather than an artifact of the open-source models used during refinement. With the refined prompt, Gemini 3.5 Flash reaches a mean of 0.7945, exceeding our best open-source submission (0.7834); it leads on both TACO (0.8466 vs. 0.8133) and TAUS (0.7870 vs. 0.7612), trailing only on TAPE (0.7498 vs. 0.7757). The gap to the official leaderboard winner (0.7955) narrows to 0.10 pp under this configuration, compared with the 1.2 pp gap of the best submitted run, suggesting that the refined prompts leave little headroom on the table and that most of the remaining gap is attributable to the underlying model rather than to the prompt.

7. Conclusion

We present *arginvariant*, an error-driven prompt refinement system for cross-domain argument identification. Distilling each corpus's expert annotation criteria directly into the prompt reduces reliance on

the dataset-specific surface shortcuts that limit fine-tuned encoder models, while keeping the system adaptable to changing annotation criteria, a property that is particularly relevant for the primary evaluation, where the same 340 tweets are labeled under three incompatible guidelines.

The best submitted run places second on the official leaderboard with a mean Macro F_1 of 0.7834, with prompt refinement improving the mean development score from 0.604 to 0.802 across the ten benchmark corpora. The key driver of improvement is iterative error-driven prompt refinement: systematic analysis of false positives and false negatives on the training set produces rule updates that transfer to unseen test sentences. Per-dataset model selection provides an additional gain on in-domain benchmark generalization (3.7 percentage points improvement in Supplementary score over a uniform model), with a negligible effect on the primary out-of-domain metric. The closed-source evaluation further shows that the refined prompts transfer across model families: with the strongest closed-source model, the gap to the leaderboard winner narrows to 0.10 pp, indicating that model capacity, not prompt quality, is the primary constraint on further improvement.

Several limitations apply. First, the approach requires one hand-crafted prompt per corpus: creating the 13 prompts used here involved substantial manual effort, including reading annotation guidelines, reviewing error patterns, and writing targeted rules. Second, the iterative refinement loop was only available for the ten benchmark corpora, which provide labeled training and development data; TAPE and TAUS had no labeled development data and relied on the transferability of the already-refined PE and USELEC prompts to the Twitter domain, while TACO used a consistency-based procedure guided by inter-LLM disagreements.

The current refinement loop requires human effort at two points: reading annotation guidelines to construct the initial prompt, and translating an identified error pattern into a concrete prompt rule. The initial prompt is written entirely by hand; for the second point, only the sub-step of translating an accepted pattern into a prompt rule remains entirely manual, since identifying the pattern itself is already partially assisted by an LLM coding agent (Section 4.2). Both remaining manual steps are candidates for further automation via a language-model-in-the-loop feedback cycle. For the initial prompt, an LLM agent could distill annotation guidelines directly from the guideline document. For the refinement step, an agentic framework (e.g., Codex or Claude Code) could take the reviewed error pattern one step further and propose the prompt rule itself, closing the loop from misclassification to prompt update without manual rule-writing. Whether such end-to-end automated refinement recovers the same implicit annotation conventions as the current human-in-the-loop process remains an open question and is the subject of ongoing investigation.

A qualitative analysis of the corpora with the largest refinement gains (Section 5.3) shows that different corpora require different kinds of rules—from explicit rejection filters to surface-convention matching—and that some residual errors resist rule-based encoding entirely. Future work includes combining error-driven prompt refinement with lightweight supervised fine-tuning.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) in order to: Grammar and spelling check, Improve writing style. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] E. Cabrio, S. Villata, Five Years of Argument Mining: A Data-driven Analysis, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018), 2018, pp. 5427–5433. doi:10.24963/ijcai.2018/766.

[2] J. Lawrence, C. Reed, Argument Mining: A Survey, Computational Linguistics 45 (2019) 765–818. doi:10.1162/coli_a_00364.

[3] M. Feger, K. Boland, S. Dietze, Limited Generalizability in Argument Mining: State-Of-The-Art Models Learn Datasets, Not Arguments, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Association for Computational Linguistics, 2025, pp. 23900–23915. URL: https://aclanthology.org/2025.acl-long.1164/. doi:10.18653/v1/2025.acl-long.1164.

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

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

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[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, CoRR abs/1907.11692 (2019).

[8] M. Feger, S. Dietze, BERTweet's TACO Fiesta: Contrasting Flavors On The Path Of Inference And Information-Driven Argument Mining On Twitter, in: Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, 2024, pp. 2256–2266. URL: https://aclanthology.org/2024.findings-naacl.146/. doi:10.18653/v1/2024.findings-naacl.146.

[9] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, Finetuned Language Models are Zero-Shot Learners, in: Proceedings of the Tenth International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=gEZrGCozdqR.

[10] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, R. Lowe, Training Language Models to Follow Instructions with Human Feedback, in: Advances in Neural Information Processing Systems, volume 35, Curran Associates, Inc., 2022, pp. 27730–27744. URL: https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, in: Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[12] F. Gilardi, M. Alizadeh, M. Kubli, ChatGPT Outperforms Crowd Workers for Text-Annotation Tasks, Proceedings of the National Academy of Sciences 120 (2023) e2305016120. doi:10.1073/pnas.2305016120.

[13] H. Li, V. Schlegel, Y. Sun, R. Batista-Navarro, G. Nenadic, Large Language Models in Argument Mining: A Survey, CoRR abs/2506.16383 (2025). URL: https://arxiv.org/abs/2506.16383.

[14] W. Yin, J. Hay, D. Roth, Benchmarking Zero-Shot Text Classification: Datasets, Evaluation and Entailment Approach, 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, Hong Kong, China, 2019, pp. 3914–

3923. doi:10.18653/v1/D19-1404.
[15] O. Sainz, I. García-Ferrero, R. Agerri, O. Lopez de Lacalle, G. Rigau, E. Agirre, GoLLIE: Annotation Guidelines Improve Zero-Shot Information-Extraction, in: Proceedings of the Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=Y3wpuxd7u9.
[16] T. Mayer, E. Cabrio, S. Villata, Transformer-based Argument Mining for Healthcare Applications, in: Proceedings of the 24th European Conference on Artificial Intelligence, 2020, pp. 2108–2115. doi:10.3233/FAIA200334.
[17] A. Lauscher, G. Glavaš, S. P. Ponzetto, An Argument-Annotated Corpus of Scientific Publications, in: Proceedings of the 5th Workshop on Argument Mining, 2018, pp. 40–46. URL: https://aclanthology.org/W18-5206/. doi:10.18653/v1/W18-5206.
[18] A. Fergadis, D. Pappas, A. Karamolegkou, H. Papageorgiou, Argumentation Mining in Scientific Literature for Sustainable Development, in: Proceedings of the 8th Workshop on Argument Mining, 2021, pp. 100–111. URL: https://aclanthology.org/2021.argmining-1.10/. doi:10.18653/v1/2021.argmining-1.10.
[19] A. Alhamzeh, R. Fonck, E. Versmée, E. Egyed-Zsigmond, H. Kosch, L. Brunie, It's Time to Reason: Annotating Argumentation Structures in Financial Earnings Calls: The FinArg Dataset, in: Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing, 2022, pp. 163–169. URL: https://aclanthology.org/2022.finnlp-1.22/. doi:10.18653/v1/2022.finnlp-1.22.
[20] S. Haddadan, E. Cabrio, S. Villata, Yes, we can! Mining Arguments in 50 Years of US Presidential Campaign Debates, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4684–4690. URL: https://aclanthology.org/P19-1463/. doi:10.18653/v1/P19-1463.
[21] R. Swanson, B. Ecker, M. Walker, Argument Mining: Extracting Arguments from Online Dialogue, in: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2015, pp. 217–226. URL: https://aclanthology.org/W15-4631/. doi:10.18653/v1/W15-4631.
[22] A. Misra, B. Ecker, M. Walker, Measuring the Similarity of Sentential Arguments in Dialogue, in: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2016, pp. 276–287. URL: https://aclanthology.org/W16-3636/. doi:10.18653/v1/W16-3636.
[23] A. Panchenko, A. Bondarenko, M. Franzek, M. Hagen, C. Biemann, Categorizing Comparative Sentences, in: Proceedings of the 6th Workshop on Argument Mining, 2019, pp. 136–145. URL: https://aclanthology.org/W19-4516/. doi:10.18653/v1/W19-4516.
[24] L. Cheng, L. Bing, R. He, Q. Yu, Y. Zhang, L. Si, IAM: A Comprehensive and Large-Scale Dataset for Integrated Argument Mining Tasks, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 2022, pp. 2277–2287. URL: https://aclanthology.org/2022.acl-long.162/. doi:10.18653/v1/2022.acl-long.162.
[25] C. Stab, I. Gurevych, Parsing Argumentation Structures in Persuasive Essays, Computational Linguistics 43 (2017) 619–659. URL: https://aclanthology.org/J17-3005/. doi:10.1162/COLI_a_00295.
[26] M. Feger, S. Dietze, TACO — Twitter Arguments from COnversations, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, 2024, pp. 15522–15529. URL: https://aclanthology.org/2024.lrec-main.1349/.
[27] 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.
[28] DeepSeek-AI, DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, CoRR abs/2501.12948 (2025). URL: https://arxiv.org/abs/2501.12948.
[29] Gemma Team, Gemma 2: Improving Open Language Models at a Practical Size, CoRR

abs/2408.00118 (2024). URL: https://arxiv.org/abs/2408.00118.
[30] Gemma Team, Gemma 3 Technical Report, CoRR abs/2503.19786 (2025). URL: https://arxiv.org/abs/2503.19786.
[31] Gemma Team, Gemma 4, Google DeepMind model card, 2026. URL: https://deepmind.google/models/gemma/gemma-4/.
[32] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, W. El Sayed, Mistral 7B, CoRR abs/2310.06825 (2023). URL: https://arxiv.org/abs/2310.06825.
[33] OpenAI, OpenAI GPT-5 System Card, CoRR abs/2601.03267 (2026). URL: https://arxiv.org/abs/2601.03267.
[34] Gemini Team, Gemini 3.5 Flash, Google DeepMind model card, 2026. URL: https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-5-Flash-Model-Card.pdf.

A. Worked Example: Instruction-Only Prompt Refinement on AFS

This appendix illustrates the refinement pipeline of Section 4 (Figure 1) in detail on a single corpus, AFS. All sentences, labels, and counts reported below are exact, drawn from the saved model predictions underlying the results in Section 5. The refinement step examined here consists entirely of added decision criteria: no corpus sentence is embedded in either prompt version as a worked example, so the illustration is independent of the reproducibility concerns that apply to example-based prompts.

A.1. Initial Prompt and Multi-Model Error Aggregation

The v1 prompt distills the AFS corpus description directly, verbatim below (`prompts/AFS/AFS1.txt`):

```
WHAT COUNTS AS AN ARGUMENT:
A sentence that asserts a position, makes a claim, or provides
reasoning for or against the debate issue:
- Pro/Con stance statements asserting a position on the topic
- Reasons or evidence supporting one side of the debate
- Rebuttal arguments challenging the opposing side
- Evaluative claims about policies, people, or events

WHAT DOES NOT COUNT AS AN ARGUMENT:
- Pure topic descriptions or background context without taking
  a position
- Neutral factual statements that do not support either side

DECISION RULE: Does the sentence make a claim or provide
reasoning that supports one side of the debated issue?
If yes -> Argument.
```

Table 7 reports the error counts obtained when this prompt is evaluated on the development split with two of the five candidate models. For both models, the failure is almost entirely one-directional: the prompt over-predicts `Argument` rather than under-predicting it. Of the errors, 120 sentences are false positives for both models simultaneously. Following the aggregation criterion of Section 4.2, this shared-error set is treated as a genuine prompt deficiency rather than a model-specific instruction-following limitation.

A.2. The Dominant Error Pattern

Table 8 lists three sentences from the shared false-positive set, together with the underlying diagnosis. AFS is drawn from genuine online debate dialogue (the Internet Argument Corpus), so a substantial

Table 7
Error counts for the initial (v1) AFS prompt on the development split ($n = 340$).

Model	Wrong	False positives	False negatives
`gemma2:27b`	153 / 340	148	5
`gemma3:27b`	136 / 340	123	13

share of its `No-Argument` sentences are personal remarks, procedural commentary, and rhetorical flourishes directed at the interlocutor rather than at the debate topic itself. The v1 decision rule provides no mechanism for distinguishing debate-adjacent tone from an actual position on the topic, and consequently misclassifies all three as `Argument`.

Table 8
Representative shared v1 false positives on AFS and their diagnosis. Gold label is `No-Argument` in all three cases; both models predict `Argument` under v1.

Example sentence	Diagnosis
"He is so full of wrong information as well as lies it's astounding."	Attacks the opponent personally rather than taking a position on the debate topic.
"How many rapes done by females have you heard about?"	Rhetorical question with no clear embedded stance.
"You made a specific point on states interests needing to be served for any law to be passed."	Meta-discourse describing what the opponent argued, rather than an argument in itself.

A.3. From Error Pattern to Prompt Constraint

Refinement did not proceed directly from v1 to v3. An intermediate version, v2, restructured the prompt around explicit stance, evidence, rebuttal, and value categories accompanied by worked examples, but yielded negligible improvement under evaluation (Macro F_1 0.453 to 0.459 for `gemma2:27b`); the error pattern identified above persisted. Version v3 supersedes v2 with a more thorough revision and is the next version validated across the full model pool on the development split, which makes it the appropriate comparison point for this illustration.

v3 replaces the single v1 decision rule with four structural tests and a set of explicit exclusions, verbatim below (`prompts/AFS/AFS3.txt`):

```
A sentence is an ARGUMENT if ALL of the following hold:

(1) ARGUMENT CONTENT: The sentence explicitly expresses ONE of:
  - A CONCLUSION: A specific position or stance on the debate
    topic
  - A PREMISE: A specific reason or evidence supporting or
    attacking a position
  - A CAUSAL OR CONDITIONAL CLAIM: "X because Y", "If X then Y",
    where a position is supported by reasoning

(2) SELF-CONTAINEDNESS: A reader who knows the general debate
topic can identify what argument position or reason is being
expressed from this sentence alone -- without needing to read
surrounding posts.

(3) ADEQUATE DEVELOPMENT: The sentence has enough content to
express a complete idea -- not a single-word reaction or a
sentence fragment that gestures at a position without stating it.
```

```
THE SUMMARY TEST: Ask yourself -- could this sentence appear as
a bullet point in a Pro or Con list on a debate website like
iDebate or ProCon?

A sentence is NOT AN ARGUMENT if:
  - It addresses the opponent personally rather than the topic
  - It comments on the conversation itself (meta-discourse)
  - It is too vague to identify what position is being expressed
  - It relies entirely on pronouns without sentence-internal
    referents
  - It is a rhetorical question with no inferable embedded
    position
  - It is extremely short (under ~8 words) with no developed
    claim
```

Each of the three sentences in Table 8 maps directly onto one of these criteria (personal address, meta-discourse, and rhetorical question without an embedded position, respectively) without having been shown to the model: the rule generalizes to sentences beyond those that revealed the deficiency.

A.4. Validation on the Development Set

Table 9 reports the classification outcome for the three examples under v1 and v3. All three are corrected for both models. More broadly, 79 of the 120 shared v1 false positives (66%) are resolved by v3 through this instruction revision alone. Development Macro F_1 improves accordingly, from 0.553 to 0.758 for `gemma3:27b` and from 0.453 to 0.638 for `gemma2:27b`, consistent with the trajectory toward the final reported score of 0.800 in Table 2 (v3 constitutes an intermediate checkpoint, not the final prompt version).

Table 9
Classification outcome for the three examples in Table 8 under v1 and v3.

Example	v1 (`gemma2/gemma3`)	v3 (`gemma2/gemma3`)
"He is so full of wrong information…"	incorrect / incorrect	correct / correct
"How many rapes done by females…"	incorrect / incorrect	correct / correct
"You made a specific point…"	incorrect / incorrect	correct / correct

A.5. Residual Errors After Refinement

Forty-eight shared false positives remain after v3, clustering around a different failure mode: sentences that report a legal ruling, historical event, or statistic without taking a stance, and sentences whose reference depends on external context the model cannot observe. The self-containedness and personal-address criteria introduced in v3 do not target this pattern; addressing it required a further refinement iteration, which falls outside the scope of this instruction-only illustration. This distinction is itself informative: not every shared error pattern responds to the same category of rule, and identifying which structural criterion a given failure requires is part of the judgment exercised during refinement.

Mapped onto Figure 1, the steps above correspond to one pass through the inner loop (running the prompt across the model pool, aggregating errors shared across models, and adding one targeted rule) followed by an outer-loop check against development Macro F_1 that confirms genuine improvement before refinement continues. The example also clarifies a point that could otherwise be read into Section 4.2: refinement is not synonymous with accumulating worked examples. The dominant share of the v1-to-v3 improvement on AFS follows entirely from better-specified decision criteria, rules that transfer to sentences not seen during refinement, rather than from pattern-matching against specific worked examples.