Graz University of Technology at Touché: Prompting Strategies for LLM-based Fallacy Detection

Touché at CLEF 2026

Alex Ghiriti[1], Roman Kern[1]

[1]Graz University of Technology, Institute of Machine Learning and Neural Computation, Graz, Austria

Abstract

This paper presents a comparative study of prompting strategies for large language model-based fallacy detection on the Touché 2026 task. Three strategies are evaluated across two axes of variation: classification topology (per-class binary versus single-prompt multiclass) and the prior given to the model (strictness versus few-shot annotator-alignment). The strategies are run on two models, gpt-5.4-mini and gemma-4-31b-it, and on the base and enhanced data conditions provided by the task organisers. Four distinct observations emerge from this analysis. On fallacy type classification, the multiclass topology consistently outperforms per-class binary prompting, and few-shot calibration is counterproductive with the harm growing on the enhanced data condition. On argument scheme classification, the ranking inverts: few-shot calibration is the best strategy and multiclass the weakest. A per-class router fitted on the evaluation slice adds a small further gain on top of the best single model. In summary, the results support a single methodological principle: prompt structure should mirror task structure, with multiclass formulations suited to mutually confusable categories and per-class binary prompting with annotator-aligned examples suited to conceptually orthogonal binary decisions.

Keywords

fallacy detection, argument mining, large language models, prompting strategies, in-context learning, Touché, CLEF 2026

1. Introduction

The detection of logical fallacies in argumentative text is a long-standing problem in natural language processing, and one with growing practical relevance for media literacy as online debate moves to platforms with little verification [1]. The task is considered to be hard, even for humans. The line between a sound argument and a fallacious one often depends on context, intent, and annotator judgment, and the same surface pattern can be acceptable in one setting and fallacious in another [2]. The Touché 2026 fallacy detection task at CLEF [3, 4] frames this challenge as three subtasks on a shared dataset: a binary judgment of whether an argument is fallacious (Subtask 1), an eight-way classification of the fallacy type (Subtask 2), and a two-dimensional classification of the argument scheme for non-fallacious arguments (Subtask 3). The dataset is derived from the informal fallacy corpus of Sahai et al. [5]. Our work studies how the structure of a prompt affects the behaviour of a large language model on this task. Three prompting strategies are compared: a zero-shot binary baseline that asks one yes/no question per class, a binary variant with a small number of few-shot calibration examples chosen near the annotator decision boundary, and a single-prompt multiclass formulation that presents all classes together with a *none* option. Each strategy is run on two models, `gpt-5.4-mini` [6] and `gemma-4-31b-it`[7], and on both the base and the enhanced data conditions provided by the organisers. The aim is to isolate the effect of structural choices in prompt design on a task whose categories are semantically close and whose boundaries are subjective. The main contributions of this work are the following:

- A controlled comparison of three prompting strategies, based on two design dimensions (classification topology, strictness prior) yielding a binary baseline, a binary variant with few-shot calibration and a single prompt multiclass formulation.

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

aghiriti@student.tugraz.at (A. Ghiriti); rkern@tugraz.at (R. Kern)

- We address the research gap of one-vs-all approaches for LLM-based classification in combination with a bespoke router method
- Comparison of two model architectures of estimated similar sizes, with one representative being available as open-weight model, allowing for reproducibility, and a proprietary widely used model

The broader takeaway is methodological. Prompt structure should mirror task structure: when categories are mutually confusable, a multiclass formulation provides a comparative anchor that per-class binary prompting lacks.

2. Related Work

The present study sits at the intersection of three lines of work: computational fallacy detection and the dataset lineage on which the Touché 2026 task builds, the broader literature on in-context learning and few-shot prompting for large language models, and the evolving literature on how classification topology shapes large language model behaviour.

2.1. Computational Fallacy Detection

Automated fallacy detection has matured from small, schema-driven resources toward larger and more naturalistic corpora. Argotario [8] introduced a serious-game crowdsourcing platform with five fallacy classes and was the first dedicated natural language processing resource for the task. Subsequent work expanded both the schema and the linguistic register. Jin et al. [9] released the LOGIC and LogicClimate datasets with thirteen fallacy types and reported that pre-trained language models perform poorly on fallacy reasoning even with substantial supervision. Goffredo et al. [10] combined fallacy labels with argumentative structure on political debates. The Sahai et al. corpus [5] from which the Touché 2026 data is derived sits in this lineage: a crowd-cleaned Reddit corpus mapped against the pragma-dialectical theory of argumentation, with eight informal fallacy classes. Closer to the per-class analysis pursued in this paper, Habernal et al. [11] showed that the dynamics of even a single fallacy type vary substantially across web argumentation contexts, which motivates evaluating model performance class by class rather than only in aggregate. The broader field is surveyed by Lawrence and Reed [12].

2.2. Prompting and In-Context Learning

The few-shot prompting paradigm was popularised by Brown et al. [13] as a way of adapting large language models to new tasks without parameter updates. Subsequent work has documented that this paradigm is more fragile than the original framing suggested. Liu et al. [14] showed that example selection matters substantially, with retrieval-based selection outperforming random sampling. Lu et al. [15] demonstrated that the order of the same set of examples can swing performance from near state-of-the-art to near-random. Zhao et al. [16] identified systematic majority, recency, and common-token biases in few-shot classification, and proposed contextual calibration as a corrective. Min et al. [17] reported that the correctness of the demonstration labels matters less than the label-space and formatting exposure they provide, which complicates the interpretation of any few-shot prompt as supplying task-specific supervision. Together, these findings frame the second finding of the present paper: a few-shot calibration strategy is plausible in principle, but its effect is sensitive to design choices and is not guaranteed to be positive.

2.3. Classification Topology in Large Language Model Classifiers

The choice between framing a classification task as a sequence of per-class binary decisions and framing it as a single multiclass decision has received less attention in the large language model literature than the choice of demonstrations or instructions. Edwards and Camacho-Collados [18] benchmarked in-context classification under binary, multiclass, and multilabel framings across sixteen datasets and

reported that framing materially affects accuracy. Kokkodis et al. [19] studied prompt-format choices for multiclass classification and found that label-set presentation and structure shape accuracy, calibration, and cost. Closest to the design comparison pursued in this paper, Langner et al. [20] reformulated a multi-label classification task as a sequence of dichotomic decisions and compared it with single-prompt structured generation, reporting trade-offs in both efficiency and accuracy. Within the literature surveyed for this paper, no work was found that directly compares per-class binary prompting and single-prompt multiclass prompting on a fine-grained fallacy taxonomy. The present paper occupies that gap.

3. Method

3.1. Task Formulation

The Touché 2026 fallacy detection challenge defines three subtasks on a shared dataset. Subtask 1 is a binary judgment of whether an argument is fallacious. Subtask 2 is an eight-way classification of the fallacy type for fallacious arguments, drawn from a fixed taxonomy of informal fallacies: Appeal to Authority, Black-White / False Dichotomy, Hasty Generalization, Appeal to Nature, Appeal to Popularity / Bandwagon, Slippery Slope, Appeal to Tradition, and Worse Problems / Whataboutism [3, 4]. Subtask 3 is a two-dimensional classification of the argument scheme for non-fallacious arguments: a *basis* dimension distinguishing internal support (definitions, principles, normative reasoning) from external support (data, sources, observable facts), and a *goal* dimension distinguishing practical claims (what should be done) from epistemic claims (what is true). Subtasks 1 and 2 are treated jointly. A classifier that distinguishes among the eight fallacy types and an explicit *none* label is, by construction, a fallacy detector: the binary judgment of Subtask 1 is obtained by collapsing the eight type labels to *fallacious* and treating *none* as *not fallacious*. This avoids running two separate systems for what is structurally one decision. Subtask 3 is addressed as a parallel two-dimensional classification using the same set of prompting strategies.

3.2. Data

The dataset is derived from the informal fallacy corpus of Sahai et al. [5] and distributed by the Touché 2026 organisers [3, 4]. It contains 938 entries in total. Each item is provided by the organisers in several parallel forms; the experiments in this paper use the `text_base` and `text_enhanced` fields. The `text_base` version combines the original comment with its context into a single self-contained text that is intended to be naturalistic. The `text_enhanced` version is the same text rewritten, using Claude Opus 4.6, so that the argument scheme dimensions become clearly identifiable and, where a fallacy is present, the fallacious reasoning pattern is made more explicit, while the rewrite is constrained to preserve the original tone and stance. The two conditions are evaluated separately, since strong performance on enhanced data partly reflects the ability to extract this surfaced signal rather than to detect fallacies from naturalistic text. Evaluation is carried out on a stratified slice of 160 items drawn from the released training data. The slice is balanced 50/50 between fallacious and non-fallacious arguments. The fallacious half is split evenly across the eight fallacy types (10 items per type). The non-fallacious half is split evenly across the same eight types according to which fallacy each item *resembles* without committing it (10 items per resembled type). This composition keeps every fallacy boundary equally represented at evaluation time. The remaining 778 items serve as a pool from which calibration examples are drawn for the few-shot strategy. The official test split is not used as the primary evaluation surface, for reasons discussed in Section 4.

3.3. Prompting Strategies

The main aim of our study is to compare prompting strategies and their impact on the performance of LLMs on the target task. They vary along two design dimensions:

- *Classification topology*: per-class binary similar to a one-vs-all approach, in which one yes/no decision is made for each class, versus single-prompt multiclass, in which one decision is made over all classes plus a *none* option.
- *Prior given to the model*: a *strictness* prior that biases the model toward the *none* label by asking it to require clear and unambiguous evidence, or an *annotator-alignment* prior that asks the model to follow how human annotators in the dataset actually labelled comparable cases rather than the strict definition.

The two priors are mutually exclusive in their wording: an instruction to be strict toward the textbook category and an instruction to follow annotator labels that may deviate from that category cannot coexist without contradiction. Every prompt therefore uses one prior or the other. The three strategies described below combine these two axes in three different ways.

3.3.1. Binary Baseline

The binary baseline poses one yes/no question per class. For the joint Subtask 1+2 group this yields eight independent binary classifiers, one per fallacy type. For Subtask 3 it yields two additional binary classifiers, one for the *basis* dimension and one for the *goal* dimension. Each prompt provides the definition of the target class and asks the model whether the argument commits that specific fallacy or falls on a given side of the scheme dimension. The strictness prior is used: the model is instructed to require clear and unambiguous evidence before returning a positive verdict. No in-context examples are provided. Listing 1 shows the template instantiated for the Appeal to Authority class.

Listing 1: Binary baseline system prompt for the Appeal to Authority class.

```
You are an expert fallacy classifier. Decide whether the argument commits the Appeal to
 Authority fallacy. This fallacy uses a person's or institution's prestige or
credentials as the primary warrant for a claim, with no engagement with the actual
reasoning. Be strict: choose none unless a fallacy is clearly and unambiguously present.
 Reply with valid JSON only: {"verdict": true|false}
```

3.3.2. Binary with Few-Shot Calibration

The second strategy retains the per-class binary topology of the baseline but replaces the strictness prior with an annotator-alignment prior and adds four in-context examples per request. The system prompt asks the model to classify by how human annotators in coarse-grained fallacy datasets actually labelled, not by the strict philosophical definition, and to recalibrate against any disagreement between the textbook definition and the examples before classifying the real argument. The system prompt template is shown in Listing 2.

Listing 2: Binary calibration system prompt template. The class name is filled in per classifier.

```
You are a {name} fallacy classifier. Classify by how human annotators in coarse-grained
 fallacy datasets actually labeled - not by the strict philosophical definition. The 4
calibration examples reveal the annotator boundary; recalibrate against any
disagreement before classifying the REAL argument.
```

The motivation for this strategy follows from the subjective nature of fallacy labels. The boundary between a hasty generalisation and a well-supported generalisation, or between an appeal to authority and a legitimate appeal to expertise, depends on annotator judgment in ways that the textbook definition cannot fully capture. A small number of boundary-spanning examples is one natural way to communicate that judgment to the model. The four calibration examples per request follow a fixed composition. One example is a positive case of the target fallacy. One is a negative case that resembles the target fallacy without committing it. One is a positive case of a different fallacy. One is a negative

case that resembles a different fallacy. The four examples are presented in random order. They are sampled at random from the 778-item pool described in Section 3.2, with resampling per request: every evaluation item is paired with a freshly drawn set of four examples. The four-shot prompt is delivered across two conversational turns. The first turn presents the four examples and asks the model to label them. The model's response is recorded. The second turn provides the model's own labels back as context, together with the actual argument to be classified. This two-turn structure keeps token consumption manageable without altering the four-example calibration content.

3.3.3. Single-Prompt Multiclass

The third strategy presents all classes together in a single prompt and asks the model to select the most appropriate label, including an explicit *none* option. For the Subtask 1+2 group this is a nine-way decision over the eight fallacy types and *none* (Listing 3). For Subtask 3 the basis and goal dimensions are presented jointly in one prompt, and the model returns one label for each dimension in a single response (Listing 4). The strictness prior is used, in the same wording as the binary baseline: this strategy differs from the baseline along the topology axis but not along the prior axis. No in-context examples are provided.

Listing 3: Multiclass fallacy prompt.

```
You are an expert fallacy classifier. Identify which single fallacy the argument
commits, or none.
- authority: uses a person's or institution's prestige as the primary warrant, no
engagement with actual reasoning
- blackwhite: presents two options as exhaustive when other alternatives exist
- hasty_generalization: draws a broad conclusion from an explicitly stated small or
unrepresentative sample
- natural: argues something is good or bad because it is natural or artificial
- population: uses majority opinion, popularity, or bandwagon pressure as the primary
warrant
- slippery_slope: opposes an action by asserting it will lead to an inevitable chain of
 worse outcomes
- tradition: justifies a practice by its longstanding historical acceptance
- worse_problems: deflects a concern by pointing to a more serious problem elsewhere,
or "your side does it too"
Be strict: choose none unless a fallacy is clearly and unambiguously present.
Reply with valid JSON only: {"fallacy": "authority|blackwhite|hasty_generalization|
natural|population|slippery_slope|tradition|worse_problems|none"}
```

Listing 4: Multiclass argument scheme prompt.

```
You are an expert argument analyst. Classify the argument on two dimensions.
Basis - is the support drawn from:
  internal: definitions, principles, logical structure, normative or theological
reasoning
  external: data, research, named sources, observable facts, historical events
Goal - is the main claim about:
  practical: what should or should not be done (action, recommendation, permission)
  epistemic: what is, was, or will be true (fact, description, causal claim)
Reply with valid JSON only: {"basis": "internal|external", "goal": "practical|epistemic
"}
```

The motivation for the multiclass strategy is structural. The eight fallacy types are not mutually exclusive in surface form: most arguments share features with more than one class, and asking the model to evaluate each class in isolation provides no anchor for a negative judgment. A single multiclass

prompt with an explicit *none* option turns the decision into a comparative one, which more closely matches the choice a human annotator makes when selecting among related labels.

3.4. Aggregation and Cross-Model Routing

For the binary strategies, a final single-label prediction per item is required. Two edge cases can occur. When none of the eight binary classifiers returns *true*, the prediction is set to *none*. When more than one classifier returns *true* which occurs for fewer than five items across all runs — the prediction is also collapsed to *none*, on the basis that a contradictory positive verdict from the model itself is weaker evidence for any specific class than a clean single-class verdict would be. For the multiclass strategy no aggregation is required, since the model returns a single label by construction. In addition to the single-model results, a per-class router is reported. The routing table of the router is calibrated once on the evaluation slice: for each fallacy class, the per-class F1 of the two models is computed, and the model with the higher score is recorded as the preferred model for that class. For the binary strategies the application is direct: each binary classifier is already class-specific, so the routing table determines which model is queried for each class, and in practice the non-preferred model need not be run at all once the routing is fixed. The same routing table is applied to the multiclass strategy in a slightly different form. Both models are run independently. For each fallacy class, only the predictions of the preferred model for that class are retained as positive instances of that class: for example, if gemma is the preferred model for Appeal to Authority, only items labelled *authority* by gemma are retained as authority predictions. In the rare case where two preferred models claim the same item for their respective classes, the item is labelled *none*, following the same conservative tiebreaker used for multi-fire conflicts in the binary strategies. Items that are not retained by any preferred model are also labelled *none*. The routing table itself is identical to the one used for the binary strategies and is calibrated once on the evaluation slice. The router is calibrated on labelled data once and applied to unlabelled data thereafter without further label access, so it is a deployable selection scheme rather than an upper bound. Its purpose in this paper is to test whether the two models are complementary or redundant at the per-class level.

3.5. Models and Inference Settings

Two language models are used. The first is `gpt-5.4-mini` [6], a proprietary model accessed through the OpenAI API. The second is `gemma-4-31b-it` [7], an open-weight instruction-tuned model accessed through OpenRouter. The two are chosen as representatives of the current generation of mid-sized instruction-tuned models from different developers, in order to test whether the findings hold across model families rather than only on a single system. The sampling temperature is set to 1 for both models. A fixed value across models is preferred to a model-dependent one, and a temperature of 0 is not uniformly supported across the model versions used. All other sampling parameters are left at their respective defaults. Each prompt instructs the model to return its answer as a JSON object; structured-output APIs are not used, and a parser handles minor deviations from strict JSON. Items whose response cannot be parsed as valid JSON are resubmitted to the same model with the same prompt. The two models produced parseable output for the large majority of requests, so retries account for a small fraction of all calls. The full experimental grid crosses three prompting strategies, two data conditions, and two models, yielding twelve runs per subtask group. All twelve runs share the same evaluation slice and the same aggregation procedure.

4. Evaluation

4.1. Setup

The evaluation is carried out on the held-out 160-item slice described in Section 3.2. For Subtask 2 the primary metric is the macro F1 averaged over the eight fallacy classes. For Subtask 3 the basis and

goal dimensions are each scored as binary F1. Subtask 1 is derivable from the multiclass predictions by collapsing the eight type labels to *fallacious* and *not fallacious*; the corresponding numbers for the binary strategies depend on the within-model aggregation rule described in Section 3.4. The official Touché 2026 leaderboard is not used as the primary surface for the controlled comparison, since the submitted runs cover only the base condition; official results are reported below and the development slice is used for the full grid. Runs were submitted to the shared task through TIRA [21]. Owing to runtime constraints, only base-data runs were submitted before the deadline; enhanced-data runs were not submitted. Systems are ranked by macro-averaged F1, with precision and recall macro-averaged as well. On the official base-data evaluation, the submitted system ranked first on Task 1 (fallacy detection), with macro precision, recall, and F1 of 0.807; second on Task 2 (fallacy type classification, macro F1 0.671, with precision 0.807 and recall 0.607); and second on Task 3 (argument scheme classification, macro F1 0.700, with precision 0.705 and recall 0.812). For the Task 2 submission, 25 predictions were repaired by the organisers' scorer, corresponding to items where the returned label was missing or invalid; these were counted as incorrect. These official results are consistent with the development-slice analysis reported below. On the binary detection task, the development slice gives a macro F1 of 0.744 for the stronger single model and 0.769 for the per-class router, against 0.807 on the official base evaluation; on fallacy type classification the official macro F1 of 0.671 matches the best base configuration on the slice (0.661); and on argument scheme classification the official 0.700 falls within the range observed on the slice. The development slice is used for the controlled comparison because it covers all three prompting strategies and both data conditions, whereas the official submissions cover only the base condition.

4.2. Subtask 2: Fallacy Type Classification

Table 1
Macro-averaged precision, recall, and F1 across the eight fallacy classes for the three prompting strategies, by data condition and model. Precision and recall are macro-averaged over the eight classes. Best F1 per model and data condition in bold.

		Base			Enhanced		
Strategy	**Model**	P	R	F1	P	R	F1
Binary baseline	gemma-4-31b-it	0.538	0.700	0.594	0.654	0.950	0.768
	gpt-5.4-mini	0.447	0.588	0.496	0.663	0.962	0.784
Binary calibration	gemma-4-31b-it	0.413	0.888	0.548	0.549	0.962	0.692
	gpt-5.4-mini	0.399	0.675	0.493	0.406	0.950	0.565
Multiclass	gemma-4-31b-it	0.671	0.662	**0.661**	0.815	0.938	**0.867**
	gpt-5.4-mini	0.673	0.612	**0.619**	0.782	0.988	**0.869**

The multiclass strategy dominates both binary strategies across the experimental grid (Table 1). The ranking multiclass $>$ binary baseline $>$ binary calibration holds for every model and data condition, and the multiclass advantage is consistent throughout. The per-class gains are concentrated on classes for which the fallacy boundary depends on comparison with adjacent classes. Hasty generalization improves by $+0.373$ F1 on gemma base and $+0.238$ on gpt-5.4-mini base when moving from the binary baseline to multiclass; worse problems improves by $+0.151$ and $+0.216$ respectively. The pattern is consistent with the structural reading that motivated the multiclass design: when a binary classifier in isolation has no clear anchor for a negative judgment, the model tends to over-predict the target class, and a multiclass prompt with an explicit *none* option provides the comparative reference that the binary topology lacks. The few-shot calibration strategy is consistently weaker than the binary baseline on fallacy classification. The macro F1 drop is small on base data (-0.046 for gemma, -0.003 for gpt-5.4-mini) and substantial on enhanced data (-0.076 for gemma, -0.219 for gpt-5.4-mini). The strongest signal is on enhanced data with gpt-5.4-mini, where calibration regresses on all eight fallacy

Table 2
Per-class preferred model under the multiclass strategy on the evaluation slice. The routing table is calibrated once on this slice and applied without further label access at test time.

Class	Base	Enhanced
authority	gemma-4-31b-it	gemma-4-31b-it
blackwhite	gpt-5.4-mini	gpt-5.4-mini
hasty_generalization	gemma-4-31b-it	gpt-5.4-mini
natural	gemma-4-31b-it	gemma-4-31b-it
population	gemma-4-31b-it	gemma-4-31b-it
slippery_slope	gpt-5.4-mini	gpt-5.4-mini
tradition	gpt-5.4-mini	gemma-4-31b-it
worse_problems	gemma-4-31b-it	gpt-5.4-mini

classes, with the largest single drop on hasty generalization (-0.413 F1). Inspection of the confusion matrices indicates that the regression is driven by precision rather than recall, with the false-positive counts on the worst-affected classes roughly doubling under calibration. One plausible interpretation is that the model treats the calibration examples as positive targets to be matched rather than as boundary references, and the effect is more pronounced on enhanced data because the calibration examples then introduce a competing signal that the model weighs against otherwise clear evidence in the input. A per-class router that selects the better-performing model for each fallacy class achieves macro F1 of 0.905 under multiclass on enhanced data, an improvement of $+0.036$ over the best single model. The router gain is smaller under the binary strategies (between $+0.005$ and $+0.027$), because one model dominates the routing table on those classes: under the binary baseline on base data, gemma is the preferred model for seven of the eight classes. Under multiclass the two models are more evenly matched and the router has more substitution opportunities. This reported gain should be read as an optimistic estimate: the routing table is fitted on the same 160-item slice on which it is then evaluated, so the selection of the per-class preferred model has access to the scores it is measured against. The number therefore indicates how much per-class complementarity exists between the two models on this slice, rather than a gain that is guaranteed to carry over to unseen data. Confirming whether it generalises would require fitting the routing table on one split and evaluating it on a disjoint held-out or official test split. The routing table is also not stable across data conditions: three of the eight class preferences flip between base and enhanced (Table 2), which suggests that the complementarity is a property of the joint prompt-strategy and model pair on the specific data condition, not of either model alone.

4.2.1. Precision and Recall

The precision and recall columns in Table 1 show that the three strategies reach their F1 scores in very different ways. The binary strategies are recall-heavy and precision-poor: the binary baseline recalls between 0.59 and 0.96 of the true positives but at a precision of 0.45 to 0.66, and few-shot calibration pushes recall higher still while precision falls further. The most extreme case is calibration on enhanced data with gpt-5.4-mini, where recall reaches 0.950 but precision is only 0.406: the model flags almost every candidate as the target fallacy, so it misses little but is wrong more often than it is right. This is the precision collapse behind the F1 regression discussed above, and it is consistent with the reading that the calibration examples act as positive targets the model tries to match rather than as boundary references. The intervention raises recall and depresses precision, which is the signature of over-prediction. The multiclass strategy is the only one that keeps precision and recall in balance. On base data its precision rises to around 0.67 for both models while recall stays near 0.62 to 0.66, and on enhanced data precision reaches 0.78 to 0.82 while recall remains high. The multiclass advantage in F1 is therefore driven mainly by precision rather than recall: forcing the model to choose a single label among competing classes, with an explicit *none* option, suppresses the false positives that the per-class binary framing

produces. The comparative anchor that the multiclass prompt provides translates directly into fewer spurious positive predictions, which is where the binary strategies lose the most ground.

4.3. Subtask 3: Argument Scheme Classification

Table 3
F1 on Subtask 3 argument scheme classification, by dimension. Best strategy per cell in bold.

		Base		Enhanced	
Dimension	**Strategy**	gemma-4-31b-it	gpt-5.4-mini	gemma-4-31b-it	gpt-5.4-mini
Basis	Binary baseline	0.645	0.768	0.685	0.797
	Binary calibration	**0.688**	**0.800**	**0.761**	**0.864**
	Multiclass	0.615	0.728	0.615	0.749
Goal	Binary baseline	0.796	0.737	0.854	0.811
	Binary calibration	**0.884**	**0.832**	0.915	**0.905**
	Multiclass	0.819	0.647	**0.932**	0.878

On argument scheme classification the relative ranking essentially inverts (Table 3). Few-shot calibration is the best strategy on seven of the eight scheme cells, and multiclass is the weakest of the three on six of eight. The basis dimension shows the cleanest version of the pattern: calibration improves over the binary baseline by between $+0.043$ and $+0.077$ F1 across the four cells, while multiclass loses by between -0.030 and -0.069. The goal dimension follows the same direction with similar magnitudes, with one exception (multiclass narrowly beats calibration on the enhanced / gemma cell, 0.932 versus 0.915). The structural reading is that the scheme dimensions are conceptually orthogonal binary decisions: the binary topology already fits the task, and adding annotator-alignment calibration helps because the boundaries within each dimension are subjective in practice. The multiclass treatment of scheme is, in this view, a topological mismatch: the compound prompt asks the model to commit to two unrelated binary decisions in one response, and the joint commitment appears to introduce interference rather than to provide a comparative anchor.

4.4. Discussion

The four observations above can be read as a single methodological principle: prompt structure should mirror task structure. When the categories under classification can semantically easily be confused and share surface features, a single multiclass prompt with an explicit *none* option provides the comparative anchor that allows the model to discriminate, and per-class binary prompts, by removing this anchor, push the model toward over-prediction. When the categories are conceptually orthogonal and each constitutes a well-defined binary decision, per-class binary prompts with a small number of calibration examples that span the annotator boundary are the better fit, and the same calibration mechanism that hurts in the first setting helps in the second. Several limitations apply. The evaluation rests on a single 160-item development slice rather than the official Touché test split, and a 20-item-per-class sample is not large enough to support confident conclusions about smaller per-class effects. Results are obtained from a single sample per item at temperature 1; combined with the small per-class sample, this means the individual per-class deltas quoted in the text, such as the $+0.373$ F1 gain on hasty generalization in Section 4.2, should be read as indicative rather than precise, since they may fall within run-to-run variation. A variance estimate over multiple samples or seeds would quantify this and is left to future work. The per-class router reported in Section 4.2 is fitted and evaluated on the same slice, so its $+0.036$ gain is an upper estimate whose generalisation is not established here. The comparison spans two models from two developers, which tests whether the findings hold across model families but is not enough to make broader claims about model behaviour in general. The few-shot calibration condition uses a fixed four-example budget and a specific instruction template; the calibration regression observed

on the fallacy task may be sensitive to either choice, and a more careful demonstration-selection procedure or a different instruction template could in principle close part of the gap.

5. Conclusion

This paper compares three prompting strategies for large language model-based fallacy detection on the Touché 2026 task, varying classification topology (per-class binary versus single-prompt multiclass) and the prior given to the model (strictness versus few-shot annotator-alignment). Four observations emerge from the evaluation on a stratified 160-item slice. On fallacy type classification, the multiclass topology consistently outperforms per-class binary prompting, and few-shot calibration is counterproductive with the harm growing on the enhanced data condition. On argument scheme classification the ranking inverts: few-shot calibration is the best strategy and multiclass the weakest. A per-class router fitted on the evaluation slice adds a small further gain on top of the best single model. Read together, these observations point to one methodological principle: prompt structure should mirror task structure. A multiclass formulation provides the comparative anchor that mutually confusable categories need, while binary prompting with annotator-aligned examples fits decisions that are already conceptually binary but have subjective boundaries. The same intervention can therefore be helpful or harmful depending on whether the topology and the task match. The evaluation rests on a single development slice rather than the official test split, two models from two developers, and a fixed calibration budget and instruction wording. A natural direction for future work is to test whether the topology-mirrors-task pattern holds on other fine-grained classification settings where category confusability and orthogonality can be controlled independently.

Declaration on Generative AI

During the preparation of this work, the author used generative AI in order to: draft content, paraphrase, reword, and improve writing style. After using these tools and services, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] T. M. J. Hruschka, M. Appel, Learning about informal fallacies and the detection of fake news: An experimental intervention, PLOS ONE 18 (2023) e0283238. doi:10.1371/journal.pone.0283238.

[2] C. Bonial, A. Blodgett, T. Hudson, S. M. Lukin, J. Micher, D. Summers-Stay, P. Sutor, C. Voss, The search for agreement on logical fallacy annotation of an infodemic, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 4430–4438. URL: https://aclanthology.org/2022.lrec-1.471/.

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

[6] OpenAI, Introducing GPT-5.4 mini and nano, 2026. URL: https://openai.com/index/introducing-gpt-5-4-mini-and-nano/, official product announcement, 17 March 2026.

[7] Google DeepMind, google/gemma-4-31B-it, 2026. URL: https://huggingface.co/google/gemma-4-31B-it, model card, Hugging Face.

[8] I. Habernal, R. Hannemann, C. Pollak, C. Klamm, P. Pauli, I. Gurevych, Argotario: Computational argumentation meets serious games, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2017, pp. 7–12. doi:10.18653/v1/D17-2002.

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

[10] P. Goffredo, M. Chaves, S. Villata, E. Cabrio, Argument-based detection and classification of fallacies in political debates, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 11101–11112. URL: https://aclanthology.org/2023.emnlp-main.684/. doi:10.18653/v1/2023.emnlp-main.684.

[11] I. Habernal, H. Wachsmuth, I. Gurevych, B. Stein, Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation, in: M. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 386–396. URL: https://aclanthology.org/N18-1036/. doi:10.18653/v1/N18-1036.

[12] J. Lawrence, C. Reed, Argument mining: A survey, Computational Linguistics 45 (2019) 765–818. URL: https://aclanthology.org/J19-4006/. doi:10.1162/coli_a_00364.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. 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. 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: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[14] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, W. Chen, What makes good in-context examples for GPT-3?, in: E. Agirre, M. Apidianaki, I. Vulić (Eds.), Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Association for Computational Linguistics, Dublin, Ireland and Online, 2022, pp. 100–114. URL: https://aclanthology.org/2022.deelio-1.10/. doi:10.18653/v1/2022.deelio-1.10.

[15] Y. Lu, M. Bartolo, A. Moore, S. Riedel, P. Stenetorp, Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 8086–8098. URL: https://aclanthology.org/2022.acl-long.556/. doi:10.18653/v1/2022.acl-long.556.

[16] Z. Zhao, E. Wallace, S. Feng, D. Klein, S. Singh, Calibrate before use: Improving few-shot performance of language models, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International

Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 12697–12706. URL: https://proceedings.mlr.press/v139/zhao21c.html.

[17] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, L. Zettlemoyer, Rethinking the role of demonstrations: What makes in-context learning work?, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 11048–11064. URL: https://aclanthology.org/2022.emnlp-main.759/. doi:10.18653/v1/2022.emnlp-main.759.

[18] A. Edwards, J. Camacho-Collados, Language models for text classification: Is in-context learning enough?, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 10058–10072. URL: https://aclanthology.org/2024.lrec-main.879/.

[19] M. Kokkodis, R. Demsyn-Jones, V. Raghavan, Beyond the hype: Embeddings vs. prompting for multiclass classification tasks, 2025. URL: https://arxiv.org/abs/2504.04277. arXiv:2504.04277.

[20] M. Langner, J. Eliasz, E. Rudnicka, J. Kocoń, Divide, cache, conquer: Dichotomic prompting for efficient multi-label LLM-based classification, 2025. URL: https://arxiv.org/abs/2511.03830. arXiv:2511.03830.

[21] 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.