Label at FadeIT: Fallacy-Aware LLM Reasoning for Score-Based Classification

Tiziano Labruna[1], Eleni Papadopulos[1,2]

[1]Università degli Studi di Padova, Italy
[2]Politecnico di Torino, Italy

Abstract

Recognizing fallacies in everyday argumentation is a crucial step towards fostering critical thinking and mitigating the spread of harmful discourse, especially in online environments. In this paper, we present our submission to Subtask A of the FadeIT shared task at EVALITA 2026, which focuses on coarse-grained fallacy detection in Italian social media posts. Our approach leverages large language models (LLMs) as structured reasoning components rather than end-to-end classifiers. Inspired by strategy-aware prompting methods, we decompose the detection process by independently assessing each fallacy type through a two-step prompting procedure: first eliciting a targeted analysis, and then producing a graded score indicating the strength of the fallacy's presence. These scores are subsequently employed in two distinct configurations: a threshold-based approach that directly converts scores into predictions using optimized cutoffs, and a hybrid architecture that combines scores with a supervised multi-label classifier trained to model each annotator's judgments independently. This method aims to combine the interpretative capabilities of LLMs with the robustness of supervised learning, while explicitly accounting for annotator disagreement.

Keywords

Fallacy Detection, Argumentation Mining, Large Language Models, Strategy-aware Prompting, Multi-label Classification, Annotator Disagreement

1. Introduction

Fallacies are widespread in everyday argumentation, particularly in online public discourse, where complex social and political issues are often discussed in emotionally charged and polarized contexts. This scenario justifies the necessity of recognizing fallacious reasoning in order to promote fair debate and limit the spread of disinformation. Recognizing this need, a growing number of researchers in natural language processing have focused on methods to automatically detect fallacies in text, spanning different domains, languages, and levels of granularity.

Early work on fallacy detection framed the task as a supervised problem, adopting traditional machine learning methods and neural models [1, 2] combined with hand-crafted lexical and contextual features. More recent approaches leverage pre-trained language models in an end-to-end fashion [3, 4, 5, 6, 7, 8] to better capture the subtle and context-dependent nature of fallacious reasoning. However, fallacy detection remains a particularly challenging task due to several factors: the abstract and overlapping nature of fallacy definitions, the frequent co-occurrence of multiple fallacies within the same text, and the presence of genuine disagreement among human annotators when interpreting argumentative content, a scenario of high uncertainty where model judgments can become unstable [9].

The FadeIT shared task [10] addresses these challenges by explicitly embracing annotator disagreement rather than collapsing it into a single ground truth, through the FAINA dataset [11] that models human label variation. In Subtask A, participants are required to perform coarse-grained fallacy detection, identifying which of 20 predefined fallacy types are present in a given post. Importantly, systems are evaluated against multiple gold standards corresponding to different annotators, reflecting the subjective nature of the task.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT
tiziano.labruna@unipd.it (T. Labruna); eleni.papadopulos@polito.it (E. Papadopulos)
0000-0001-7713-7679 (T. Labruna); 0009-0002-0994-5142 (E. Papadopulos)

In this work, we propose an approach that departs from purely using LLMs as end-to-end classifiers for fallacy detection. Instead, drawing inspiration from research on teaching LLMs to make strategic decisions about resource use [12] and acknowledging the limitations of their consistency in constrained tasks [13, 14], we leverage LLMs as intermediate reasoning agents that provide structured, fallacy-specific assessments of a text. Our method is inspired by recent work on strategy-aware prompting, such as PCoT [15], which injects knowledge about persuasive strategies into the reasoning process of LLMs. While PCoT focuses on jointly modeling multiple strategies in the context of disinformation detection, our approach adapts this idea to fallacy detection by considering each fallacy type independently.

Our method proceeds in two steps for each fallacy type: first, the LLM generates a short analytical explanation discussing potential presence of a specific fallacy in the text; second, it produces a numerical score from 1 to 10 reflecting the degree of fallacy presence, based on both the original text and its own analysis. This structured, chain-of-thought process is designed to enhance reasoning stability, a known challenge for LLMs in ambiguous tasks [16]. The resulting fallacy-specific scores are then used in two ways: first, scores serve as features for a supervised multi-label classifier that, rather than aggregating annotations, learns to model each annotator's judgments independently. In this way, the model aims to capture not only the presence of fallacies, but also systematic differences in how annotators interpret and apply fallacy definitions. In the second approach, we experiment with empirically selecting the optimal threshold for each score to improve prediction accuracy.

Our contributions can be summarized as follows: (I) We introduce a fallacy-aware prompting framework that decomposes coarse-grained fallacy detection into fallacy-specific reasoning steps, leveraging large language models to produce structured analyses and graded assessments for each fallacy type; (II) We assess the LLM-generated scores through an empirical threshold-selection strategy to optimize prediction performance for each fallacy; (III) We propose a hybrid architecture that combines LLM-generated fallacy scores with a supervised multi-label classifier, enabling robust prediction while avoiding fully end-to-end reliance on LLMs; (IV) We explicitly model annotator disagreement by training the classifier to predict the labels of each annotator separately, rather than collapsing annotations into a single gold standard.

2. Data and Resources

2.1. Dataset

Our experiments are conducted on the data released for the FadeIT [10] shared task as part of the EVALITA 2026 evaluation campaign. The dataset, provided by the task organizers, is based on FAINA [11] and consists of Italian social media posts discussing migration, climate change, and public health over a four-year time span. Each post is annotated for the presence of fallacies belonging to a predefined set of 20 fallacy types.

For Subtask A (coarse-grained fallacy detection), annotations are provided at span level by two independent annotators and then aggregated for each post. Each annotator may assign zero, one, or multiple fallacy labels to a post, reflecting the multi-label nature of the task. Importantly, the dataset preserves individual annotator labels rather than aggregating them into a single gold standard, in line with recent work highlighting the importance of modeling human label variation in subjective NLP tasks [17].

The data are distributed in a tab-separated format including post identifiers, temporal metadata, topic keywords, anonymized post text, and separate columns for each annotator's labels. The organizers provide a train/development split, corresponding to 80% of the data, and a held-out test set comprising the remaining 20%, which is used for the official evaluation. In our approach, the LLM-based analysis pipeline takes as input only the post identifier, the post text, and, in some settings, the topic while annotator-specific labels are employed to train the supervised classifier.

2.2. Language Model and Computational Resources

The proposed approach relies on a large language model as a reasoning component for fallacy-specific analysis and scoring. All experiments are performed using GEMMA-3-12B, an instruction-tuned multilingual open large language model released by Google. The model is accessed through the Hugging Face ecosystem using the official `transformers` library and is employed in inference-only mode, without any form of fine-tuning or parameter updates.

Model inference is carried out via API-based loading of the pre-trained weights (`google/gemma-3-12b-it`), using automatic device placement on a single NVIDIA A40 GPU. The model is executed in `bfloat16` precision to balance computational efficiency and numerical stability, and attention is computed using an eager implementation compatible with the employed PyTorch version. A corresponding processor is used for efficient tokenization and input preparation.

By relying exclusively on prompting and inference-time reasoning, our setup avoids task-specific fine-tuning and enables a clear separation between the LLM-based reasoning stage and the downstream supervised classifier. In practice, the generation stage requires $n \times k$ analyses, where n is the number of posts and k is the number of fallacy classes, since each post is evaluated independently for every fallacy type. This design choice allows the model to be treated as a reusable and modular component, while keeping computational requirements within the constraints typically available to shared task participants.

3. Methodology

Our approach leverages LLM reasoning to produce fallacy-specific scores that serve as the foundation for two classification strategies. Inspired by the strategy-aware prompting framework proposed in PCoT [15], we adapt the methodology to the fallacy detection setting by treating each fallacy type as an independent reasoning target. In this section, we outline the hybrid architecture, which consists of: (i) fallacy-specific analysis and scoring using an LLM to generate numerical features, (ii) a supervised multi-label classifier that combines these scores with contextual information. As an alternative, we also evaluate direct threshold-based classification of the LLM scores.

3.1. Overview

Given a social media post, the system processes it independently for each of the 20 fallacy types defined in the FadeIT task. For each fallacy type, the LLM is asked to generate a short analytical paragraph discussing the possible presence of a specific fallacy in the text, with the prompt explicitly including a description of that fallacy type. In the second step, the model is prompted again to produce a numerical score on a fixed scale from 1 to 10, indicating the degree to which the fallacy is present, conditioned both on the original text and on the previously generated analysis. This results in a vector of 20 scalar scores per post. These scores are subsequently used as input features to a supervised multi-layer perceptron (MLP), which predicts fallacy presence for each annotator separately.

This design allows us to decouple fallacy-specific reasoning from the final classification step, while explicitly modeling annotator disagreement.

3.2. Two-step Fallacy-aware Prompting

For each post and each fallacy type, we employ a two-step prompting strategy.

Step 1: Fallacy-specific analysis. In the first step, the LLM is asked to analyze the post with respect to a single fallacy type. The prompt explicitly injects domain knowledge by including a definition of the target fallacy, derived from the official task guidelines. The model is instructed to focus exclusively on the selected fallacy and to be conservative in its judgment, assuming absence when evidence is insufficient.

Formally, given a post p and a fallacy type f_i, the LLM produces an analysis text a_i describing whether and how f_i manifests in p.

The prompt used is presented as follows. All prompts are written in Italian, matching the language of the analyzed texts.

Fallacy-specific system prompt (example).

> *Sei un assistente che rileva le fallacie nei testi. Le fallacie sono definite come tipi di ragionamento che sembrano validi ma non lo sono. Si verificano quando qualcuno commette un errore nell'argomentazione, sia con l'intento di persuadere il pubblico sia in modo involontario. Queste fallacie sono suddivise in diverse categorie. La tua competenza e il tuo focus sono sull'Ad hominem. Ad hominem: Un attacco personale a un individuo o a un gruppo che devia dalla tesi principale.*
>
> *You are an assistant that detects fallacies in texts. Fallacies are defined as types of reasoning that appear valid but are not. They occur when someone makes an error in argumentation, either with the intent to persuade the audience or unintentionally. These fallacies are divided into different categories. Your expertise and focus are on Ad hominem. Ad hominem: A personal attack against an individual or a group that diverts from the main claim.*

Fallacy-specific user prompt (example).

> *Dato un certo testo, valuta criticamente il suo potenziale di fallacia. Inoltre, analizza se il testo utilizza il tipo di fallacia **Ad hominem**. Spiega come l'Ad hominem appare o non appare nel testo. Sii prudente nella valutazione e, se non sei completamente certo che la fallacia sia presente, assumi che non lo sia.*
>
> *Given a certain text, critically evaluate its potential fallaciousness. Additionally, analyze whether the text uses the **Ad hominem** fallacy. Explain how the Ad hominem appears or does not appear in the text. Be cautious in your assessment and, if you are not completely certain that the fallacy is present, assume that it is not.*

Equivalent prompts are defined for all remaining fallacy types, each including a task-specific definition provided by the organizers.

Step 2: Numerical scoring. In the second step, the LLM receives both the original post and the analysis generated in Step 1, and is asked to output a numerical score from 1 to 10 representing the degree of presence of the target fallacy in the post. A higher score indicates stronger or clearer manifestation of the fallacy.

This step enforces a structured output format to facilitate automatic parsing and downstream processing.

The prompts used are presented as follows.

Scoring system prompt.

> *Sei un Rilevatore di Fallacie. Il tuo obiettivo è valutare il grado di fallacie presenti in un testo. Le fallacie sono definite come tipi di ragionamento che sembrano validi ma non lo sono. Si verificano quando qualcuno commette un errore nell'argomentazione, sia con l'intento di persuadere il pubblico sia in modo involontario. Dovrai dare un punteggio da 1 a 10 sulla base di quanto fallaci sono le argomentazioni nel testo.*
>
> *You are a Fallacy Detector. Your goal is to assess the degree of fallacious reasoning present in a text. Fallacies are defined as types of reasoning that appear valid but are not. They occur when someone makes an error in argumentation, either with the intent to persuade the audience or unintentionally. You must assign a score from 1 to 10 based on how fallacious the arguments in the text are.*

Scoring user prompt.

> *Dato un testo e un'analisi dei possibili tipi di fallacie, devi rispondere con un numero da 1 a 10 in base al grado di presenza di fallacie nel testo, preceduto da una breve spiegazione del tuo ragionamento. Fornisci la tua risposta sotto forma di dizionario: {"explanation": "...", "response": "..." }.*
>
> *Given a text and an analysis of the possible types of fallacies, you must respond with a number from 1 to 10 based on the degree to which fallacies are present in the text, preceded by a brief explanation of your reasoning. Provide your answer in dictionary form: {"explanation": "...", "response": "..." }.*

Note that the end of each of the fallacy-specific system prompts, the correspondent fallacy definition is directly embedded, as taken from the organizer's guidelines. This ensures that the LLM's reasoning is grounded in the task-specific interpretation of each fallacy and avoids cross-contamination between categories.

The two-step prompting process is repeated independently for all fallacy types, yielding a vector of scores:

$$\mathbf{s} = (s_1, s_2, \ldots, s_{20}),$$

where each $s_i \in \{1, \ldots, 10\}$ is relative to a fallacy type.

For each post, the final output of the LLM stage is the complete vector of fallacy scores. No aggregation across fallacy types is performed at this stage, as each score is treated as an independent feature capturing the model's confidence in the presence of a specific fallacy. These vectors are stored and reused across runs to ensure reproducibility and to avoid redundant inference calls.

3.3. MLP-based Aggregation of LLM Scores and Human Annotations

To combine the fallacy-related signals extracted from LLMs with human annotations, a supervised multi-layer perceptron (MLP) classifier is employed. The goal of this component is to map a structured representation derived from LLM outputs to fallacy labels, while explicitly modeling annotator-level disagreement.

Each input instance corresponds to a single post and is represented as a feature vector built from three main components:

- **LLM-based fallacy scores.** For each fallacy category, the numeric score produced by the LLM prompting pipeline is included as a feature. The ordering of these scores follows a fixed fallacy list shared across all data splits.
- **Aggregate statistical features.** In addition to raw scores, three summary statistics are computed over the fallacy score vector: mean, variance, and entropy. These features provide a compact characterization of the overall distribution of fallacy scores for a post.
- **Topic information (optional).** When enabled, topic metadata associated with each post is incorporated either as a one-hot encoded vector or as a normalized scalar index. An explicit `__UNKNOWN__` category is used to handle unseen topics.

The resulting feature vector is used as input to an MLP composed of a configurable number of hidden layers, each followed by a ReLU activation function. The final linear layer outputs a vector of size $2 \times N$, where N is the number of fallacy categories. This structure reflects the presence of two independent annotators in the dataset: the first N dimensions correspond to annotator 1, and the remaining N to annotator 2.

Formally, given an input feature vector $\mathbf{x}$, the model computes:

$$\mathbf{z} = f_\theta(\mathbf{x}),$$

where f_θ denotes the MLP and $\mathbf{z} \in \mathbb{R}^{2N}$ are the output logits. A sigmoid function is applied independently to each logit to obtain per-fallacy probabilities.

Training is performed using binary cross-entropy with logits, allowing each fallacy label for each annotator to be treated as an independent binary classification problem. This design choice enables the

model to learn asymmetric or inconsistent labeling patterns across annotators, rather than forcing a single aggregated gold label.

During inference, probabilities are thresholded to obtain binary predictions. Posts without gold annotations are included in the prediction phase but excluded from metric computation, which is performed using micro-averaged scores over all fallacy-annotator pairs.

3.4. Threshold-based classification

As an alternative to the MLP-based approach, we also evaluated a simpler optimized threshold method that directly converts LLM scores into binary predictions. For each fallacy type and each annotator, a threshold value was selected to maximize the F1 score per-fallacy on the training set. Fallacy classes with scores above the threshold are classified as positive, while those below are classified as negative. The optimized thresholds were then evaluated on the validation set. This approach offers several advantages: it provides greater interpretability, requires no training phase and allows us to assess whether the LLM-generated scores are already well-calibrated for classification without additional supervised learning.

4. Experimental Setup

A series of experiments was conducted to identify the most effective configuration for aggregating LLM-generated fallacy scores and auxiliary information. As described in Section 3, the starting point for all experiments consists of the 20 fallacy scores in the range $[1, 10]$ produced by the LLM, one for each fallacy category.

Different feature configurations were explored by selectively incorporating topic information, raw LLM scores, and aggregate statistical features. In particular:

- **Topic** indicates that topic metadata associated with each post (migration, climate change, public health) was included as an additional feature.
- **Topic (one-hot)** denotes a one-hot encoding of the topic labels.
- **Scores** refers to the inclusion of the 20 raw fallacy scores generated by the LLM.
- **Stats** refers to the inclusion of three aggregate statistics computed over the fallacy score vector, namely average, variance, and entropy.

For each feature configuration, the MLP classifier described in Section 3 was trained using the best hyperparameters selected via grid search. The explored hyperparameter space is reported below:

```
hidden_dim: {32, 64, 128}
hidden_layers: {1, 2, 3}
lr: {1e-3, 5e-4, 1e-4}
batch_size: {8, 16, 32, 64, 128}
```

Evaluation follows the shared task protocol and reports Micro F_1 scores, as well as Micro-F_1 computed separately for each annotator. Results are reported in Table 1.

5. Model Selection and Analysis

The experimental results provide several insights into the behavior of the proposed hybrid approach and its underlying design choices. A first observation concerns the overall effectiveness of incorporating topic information. As shown in Table 1, configurations that include topic features consistently outperform those relying solely on LLM-generated fallacy scores and their aggregate statistics. In particular, using topic information in combination with scores and stats yields the highest Micro F_1 scores, as well as the best performance for both annotators. This suggests that topical context plays a central role in

Table 1
Results on the FAINA validation set for different feature configurations. Ann. 1 and Ann. 2 report Micro F_1 computed separately for each annotator. Green intensity indicates higher values with proportions computed for each column.

Configuration	Micro F_1	Ann. 1	Ann. 2
Scores	39.01	30.14	42.21
Scores + Topic	40.09	30.22	47.60
Scores + Topic (one-hot)	39.18	33.42	43.82
Stats	38.23	28.77	44.98
Stats + Topic	38.69	30.05	45.61
Stats + Topic (one-hot)	40.05	28.98	47.89
Scores + Stats	37.59	28.80	44.84
Scores + Stats + Topic	**43.67**	**36.68**	**49.38**
Scores + Stats + Topic (one-hot)	42.32	35.20	48.05
Threshold-based	39.92	36.91	48.79
Baseline	20.84	15.84	25.84

fallacy detection, possibly by constraining the plausible set of fallacies and reducing ambiguity at the classification stage.

In contrast, the combination of both LLM scores and aggregate statistics does not systematically improve performance compared to those features taken individually, likely because it results to be redundant or noisy. In particular, the aggregate statistics are directly derived from the same score distributions and therefore provide limited additional information beyond what the classifier can already infer from the raw scores themselves.

The threshold-based approach demonstrates surprisingly competitive performance despite its simplicity, achieving a Micro-F1 of 39.92%. This indicates that the LLM-generated scores are reasonably well-calibrated for direct classification without necessarily requiring additional supervised learning, though combining them with scores statistics and topic information still yields the best performance.

Another notable aspect emerges from the annotator-level analysis. Across all configurations, a consistent performance gap is observed between Annotator 1 and Annotator 2, with the latter achieving higher Micro-F1 scores. This suggests the presence of substantial annotator disagreement in the dataset and supports the decision to model annotators separately rather than collapsing their labels into a single gold standard. The results indicate that the proposed MLP is able to capture annotator-specific labeling patterns, even when overall performance remains moderate.

From a broader perspective, these findings shed light on the theoretical motivation behind the adopted approach. By decomposing the fallacy detection task into fallacy-specific reasoning steps and isolating each category during the LLM prompting phase, the model explicitly mirrors the way fallacies are defined and annotated in the dataset, thus the experimental results support the theoretical intuition that fallacy detection benefits from category-wise isolation and explicit reasoning. We submitted our three best-performing configurations from the development set to the test phase.

6. Results

Results (Table 2) show that Run 3 (threshold-based) achieves the highest recall among all submitted runs (68.08% micro-averaged and 60.13% macro-averaged), though at the cost of precision. Despite this different optimization strategy, all three runs achieve similar overall F_1 scores (38-39% Micro F_1, 31-33% Macro F_1), though Runs 2 and 3 exhibit limited prediction diversity, concentrating on a subset of fallacy types.

Our systems generally demonstrate the strongest performance on fallacies such as *Loaded Language* (64.40% F_1 score on average), *Appeal to Emotion* (61.12% on average), *Vagueness* (52.01% on average), and *Name Calling* (58.47% on Run 3). Analysis of annotator-specific performance reveals systematic disagreement patterns across fallacy types, with Annotator 2 consistently outperforming Annotator 1

Table 2
Results on the FAINA test set. Annotator columns (Ann. 1, Ann. 2) report micro-F_1 computed separately for each annotator. **Bold**: best score within each metric column. **Bold**: best score across all submissions in the task. Green intensity indicates higher values with proportions computed for each column.

		Micro			Macro			Annotator	
Run	**Configuration**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**Ann. 1**	**Ann. 2**
1	Score+Stats+Topic	**52.82**	30.94	38.96	41.52	26.99	30.94	34.26	**43.65**
2	Score+Stats+Topic (one-hot)	52.76	30.11	38.32	**43.01**	26.81	30.84	33.98	42.66
3	Threshold-based	27.60	**68.08**	**39.26**	24.64	**60.13**	**33.37**	**37.39**	41.13
	Baseline	38.53	14.28	20.84	15.13	3.45	5.10	15.84	28.85

across all configurations by 3.74-9.39 F_1 points, with the smallest gap observed in Run 3. In particular, the class *Vagueness* exhibits the largest systematic difference with Annotator 2 outperforming Annotator 1 by 27.56% across all runs.

7. Conclusions

This paper presented a two-stage approach to coarse-grained fallacy detection for Italian social media posts, developed in the context of the FadeIT shared task at EVALITA 2026. Our method leverages large language models as structured reasoning components through a two-step prompting procedure that decomposes the detection task into independent evaluations for each of the 20 fallacy types, generating numerical scores that capture both the presence and strength of individual fallacies. These LLM-generated scores are exploited in two complementary ways: (i) a threshold-based approach that applies empirically optimized cutoffs directly to the scores for classification and (ii) a hybrid architecture that combines the scores with a supervised multi-label MLP classifier, optionally augmented with topic and statistical features. In the hybrid configuration, the classifier models each annotator's judgments independently, aiming to capture systematic differences in how annotators interpret and apply fallacy definitions.

Preliminary experimental results for model selection on Subtask A demonstrate that both approaches achieve competitive performance. On the validation set, the hybrid configuration incorporating scores, aggregated statistics and topic information achieved the highest performance (43.67% Micro F_1), while the threshold-based approach proved remarkably effective given its simplicity (39.92% Micro F_1). In the test phase, the threshold-based approach attained the highest recall among all submitted systems (68.08% micro-averaged, 60.13% macro-averaged). Despite different optimization strategies, all three submitted runs achieved similar overall F_1 scores (38-39% micro F_1, 31-33% macro F_1).

Beyond task-specific performance, this work contributes to the broader discussion on how LLM-based reasoning can be integrated into supervised learning pipelines for subjective NLP tasks. By processing each fallacy type independently through dedicated prompts, our approach enhances transparency: each classification decision can be traced back to a specific analytical explanation and numerical score for that fallacy. At the same time, more sophisticated architectures aimed at transfering reasoning into downstream classifiers could potentially improve classification accuracy.

Future work may explore alternative representations of LLM outputs, such as structured explanations or uncertainty-aware features, as well as extensions to fine-grained span-level detection. More generally, this line of research points toward hybrid systems that balance interpretability, contextual awareness, and robustness to human disagreement in fallacy detection and related argumentation tasks.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5 and Gemini-3 in order to conduct grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and

take full responsibility for the publication's content.

References

[1] I. Habernal, P. Pauli, I. Gurevych, Adapting serious game for fallacious argumentation to German: Pitfalls, insights, and best practices, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: https://aclanthology.org/L18-1526/.

[2] 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.

[3] 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.

[4] M. Chaves, E. Cabrio, S. Villata, Falcon: A multi-label graph-based dataset for fallacy classification in the covid-19 infodemic, in: Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing, SAC '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 988–995. URL: https://doi.org/10.1145/3672608.3707913. doi:10.1145/3672608.3707913.

[5] 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.

[6] T. Alhindi, T. Chakrabarty, E. Musi, S. Muresan, Multitask instruction-based prompting for fallacy recognition, 2023. URL: https://arxiv.org/abs/2301.09992. arXiv:2301.09992.

[7] Y. Lei, R. Huang, Boosting logical fallacy reasoning in llms via logical structure tree, 2024. URL: https://arxiv.org/abs/2410.12048. arXiv:2410.12048.

[8] J. Jeong, H. Jang, H. Park, Large language models are better logical fallacy reasoners with counter-argument, explanation, and goal-aware prompt formulation, 2025. URL: https://arxiv.org/abs/2503.23363. arXiv:2503.23363.

[9] T. Labruna, S. Gallo, G. Da San Martino, Positional bias in binary question answering: How uncertainty shapes model preferences, in: Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), 2025, pp. 550–560.

[10] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[11] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: https://aclanthology.org/2025.naacl-long.34/. doi:10.18653/v1/2025.naacl-long.34.

[12] T. Labruna, J. A. Campos, G. Azkune, H. Center-Ixa, When to retrieve: Teaching llms to utilize information retrieval effectively, Context 12 (2024) 14.

[13] T. Labruna, S. Brenna, G. Bonetta, B. Magnini, Are you a good assistant? assessing llm trustability in task-oriented dialogues, in: Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), 2024, pp. 470–477.

[14] T. Labruna, B. Magnini, Evaluating task-oriented dialogue consistency through constraint satisfaction, arXiv preprint arXiv:2407.11857 (2024).

[15] A. Modzelewski, W. Sosnowski, T. Labruna, A. Wierzbicki, G. Da San Martino, Pcot: Persuasion-augmented chain of thought for detecting fake news and social media disinformation, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 24959–24983.

[16] S. Casola, T. Labruna, A. Lavelli, B. Magnini, Testing chatgpt for stability and reasoning: A case study using italian medical specialty tests, in: Proceedings of the Ninth Italian Conference on Computational Linguistics (CLiC-it 2023), 2023.

[17] B. Plank, The "problem" of human label variation: On ground truth in data, modeling and evaluation, 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. 10671–10682. URL: https://aclanthology.org/2022.emnlp-main.731/. doi:10.18653/v1/2022.emnlp-main.731.