TiGRO 🐯 at FadeIT: *E Pluribus Unum* – A Multi-task Approach to Fallacy Detection and Span Identification

Stefano **Atzeni**[1,2], Gabriele **Sarti**[2], Tommaso **Caselli**[2] and Malvina **Nissim**[2]

[1]*University of Turin, Turin, Italy*
[2]*Rijksuniversiteit Groningen, Groningen, the Netherlands*

Abstract

As a joint Turin-Groningen effort, the TiGRO team participated in the Fallacy Detection in Italian Social Media Texts (FadeIT) task at EVALITA 2026. We submitted three systems for Subtask A, which involves predicting fallacy labels across 20 classes and 2 annotators. We also submitted three systems for Subtask B, which involves predicting the text segments corresponding to the fallacies. We merged the two annotations and addressed the problem using different configurations of multi-task learning, framing Subtask A as 20 binary classification tasks and Subtask B as 20 BIO tagging tasks. Our best model jointly trains on both post-level and span-level tasks, resulting in a total of 40 tasks. Our team ranked **first for Subtask A** among fifteen submissions, achieving a micro F1 score of **56.39**, and fourth for Subtask B among nine submissions, with a micro F1 score of **44.98**. We also discuss strategies which we eventually did not adopt in our submitted runs, such as multi-lingual data augmentation and attribution-based span detection, and put forward a few ideas to enhance dataset quality.

Keywords

fallacy detection, Italian, encoder-based models, multi-task learning

1. Introduction

Argumentation and dialogue are two pillars supporting peaceful coexistance among individuals and nurturing democratic societies. However, conversational exchanges can be flawed by fallacious arguments. Fallacies are identified as instances of faulty reasoning in the construction of an argument. Superficially, they are arguments that appear to be valid while being faulty [1, 2]. Fallacies tend to be widespread across highly debated topics, whether introduced unintentionally through negligence or used intentionally to persuade audiences. Fallacies are particularly heinous in the context of political debates as they can be used as strategies to convince audiences and gain support [3], thus poisoning the public discourse and contributing to the spread of misinformation and fake news. As a matter of fact, previous work has shown that fallacies played a role in events such as the Brexit referendum [4] and in the "infodemic" crisis surrounding the COVID-19 pandemic [5].

The study of fallacies has a long tradition dating back to Aristotle and have always been contrasted to an ideal model of good arguments and reasoning [6]. The changes in human communication mediated by the growth and promotion of social media offers an excellent playground to test the functionalities of recent Natural Language Processing (NLP) systems in detecting fallacies in every day communication. The availability of a robust tool for fallacy detection in social media communication can contribute to the development of effective countering interventions against misinformation and harmful content [7, 8].

Automatic Fallacy Detection (AFD) has recently seen a surge with the availability of new datasets [9, 10, 11, 12] covering different domains such as politics and news, and LLM-based models [13, 14, 15, 16, 17, 8], including new shared task on multimodal fallacy detection [18].

Tasks Description The FadeIT Shared Task at EVALITA 2026 [19] is the first shared task on AFD for Italian using social media texts. The task is organized into two subtasks:

- **Subtask A: Coarse-grained fallacy detection.** Given the text of a social media post, the participants are asked to predict all the fallacies expressed in it. This is a multi-label classification task involving 20 fallacy classes.

- **Subtask B: Fine-grained fallacy detection.** Given the text of a social media post, the participants are asked to identify the text spans corresponding to each of the fallacies expressed in it. This is a multi-label sequence labeling task.

This paper presents the description of the systems our team submitted to the FadeIT shared task. Section 2 introduces the FAINA dataset used for training and evaluation. Our systems are described in Section 3, our best runs ranked first in Subtask A and fourth in Subtask B among all other participants. Results are discussed in Section 4, including a per-class analysis which highlights some limitations of the dataset. Section 5 reports on methods explored during system development but ultimately not adopted. Finally, we outline directions for future work in Section 6.

2. Dataset

The task is based on the FAINA dataset introduced by Ramponi et al. [8], which consists of 1,440 Italian Twitter/X posts collected over a four-year period (from 2019-01-01 to 2022-12-31) covering topics such as migration, climate change, and public health.

A total of 20 fallacy types were identified by harmonizing definitions and names from Da San Martino et al. [20] and Musi et al. [21], among others. The complete list of definitions and examples is provided in Ramponi et al. [8]. The dataset was annotated by two expert annotators following a prescriptive protocol [22], with multiple rounds and discussion between the annotators. As Inter-annotator Agreement (IAA) measure, the authors report γ and γ_{cat} as proposed by Mathet [23]. When compared to other IAA coefficient used in NLP, these two metrics offers insights both on how well annotations align (γ) and the extent to which annotation disagreement depends on categories rather than on the identification of spans (γ_{cat}). The reported results indicates that fallacy annotation is challenging and highly subjective with γ = 0.624 (span identification agreement) and γ_{cat} = 0.544 (span classification agreement).

Following the Perspectivist manifesto principles[1], annotation differences have not been reconciled into a single ground truth: each annotation has been considered valid, resulting in a total of 11,064 spans. Figure 1 illustrates an example of two different, but equally valid, annotations for the same post.

We were provided with 80% of the original FAINA dataset (1,152 posts) for training and validation, with the remaining 20% used as a held-out test set. We counted 4,124 unique fallacy annotations, with 3.58 fallacies per post on average. Table 1 summarizes the number and percentage of posts containing each fallacy in the released dataset. When a fallacy appears in both annotations for the same post, it is counted once. As shown in Table 1, the dataset is highly skewed, with few classes dominating the distribution.

Figure 1: Example of different annotations provided by the two annotators (en: *"A flood of immigrants is coming in and nobody is talking about it. Problems keep piling up, where are we going to end up?"*)

---

[1]https://pdai.info

Table 1
Number and percentage of posts annotated with each fallacy (posts may contain multiple fallacies).

Fallacy	Count	Percentage (%)
Loaded-language	699	60.68
Appeal-to-emotion	652	56.60
Vagueness	534	46.35
Name-calling-or-labelling	362	31.42
Hasty-generalization	225	19.53
Doubt	214	18.58
Evading-the-burden-of-proof	204	17.71
Flag-waving	158	13.72
Ad-hominem	140	12.15
Thought-terminating-cliches	138	11.98
Slogan	130	11.28
Red-herring	129	11.20
False-analogy	125	10.85
Appeal-to-authority	97	8.42
Slippery-slope	84	7.29
Causal-oversimplification	68	5.90
Strawman	58	5.03
Cherry-picking	58	5.03
False-dilemma	39	3.39
Circular-reasoning	10	0.87

3. Systems

In this section, we describe the settings and methods of the submitted systems for both subtasks. As a general strategy, we focused on identifying learning approaches that could maximize the results on both subtasks keeping in mind that, on the one hand, they are framed as separate tasks for the challenge while, on the other, they are very closely related and can inform each other. Indeed, a given text span can trigger a specific fallacy but, at the same time, knowing that a specific fallacy is present in the text can facilitate the identification of the corresponding span.

This has led us to experiment using two main modeling strategies: single-task and multi-task learning. In both cases, we decided to fine-tuned models rather than opting for in-context learning with (generative) LLMs. This decision is mostly dictated by two elements: firstly, LLMs still underperform when prompted to use many classes [24], such as in this case; secondly, LLMs are known to struggle with such a complex task as previously reported by Ramponi et al. [8].

Based on the results obtained on our development split (see Training Setup), we selected six systems to submit to the official challenge, three per subtask. Table 2 summarizes the main characteristics of each system; the remainder of this section provides details and explanations of our modeling choices.

Table 2
Overview of the submitted TiGRO models.

System	Approach	Encoder	Subtask
TiGRO-A1	Single-task (one vs rest)	mmBERT	A
TiGRO-A2	Multi-task (post only)	ALBERTo	A
TiGRO-A3	Multi-task (post and span)	ALBERTo	A
TiGRO-B1	Multi-task (span only)	ALBERTo	B
TiGRO-B2	Multi-task (post and span)	ALBERTo	B
TiGRO-B3	Multi-task (post and span)	mmBERT	B

Dealing with Two Annotators Fallacy labels and spans for two different annotators are made available for the dataset, under the vision that single ground truth labels in inherently subjective tasks (such as AFD) are not desirable. Embracing this vision as well, rather than treating the two sets of annotations as separate, we took a broader view and decided to merge the annotations. In practice, if multiple labels are present, all of them are preserved, so the final label set for each post or token is the union of the labels assigned by the two annotators. At the span level, when two spans of the same fallacy class overlap but have slightly different boundaries, we keep a single span covering the union of both annotations. Besides theoretical considerations, the decision to adopt this strategy has been also dictated by the limited size of the dataset and the large number of labels where further splitting the signal across annotators substantially increases the task's complexity. This strategy led to consistent performance improvements across all tested architectures.

Modeling Approaches One overall decision we made, independently of other implementation choices, is to address fallacy labeling (Subtask A) as a collection of binary tasks. Rather than dealing with a multi-class problem with 20 classes, we deal with 20 different binary tasks, one per fallacy. Subtask B was framed as a BIO tagging problem. As mentioned, for Task A we experimented with both single- and multi-task approaches. In the single-task setup, we fine-tuned 20 independent binary classifiers, one for each fallacy type (**TiGRO-A1**).

In multi-task learning, we experimented with leveraging interactions both across classes within Subtask A as well as between Subtask A and Subtask B. In one setting, we trained a multi-task classifier by fine-tuning 20 parallel tasks which jointly learn binary fallacy classification at the post level for Subtask A (**TiGRO-A2**). For Subtask B, we adopted the same multi-task strategy but framed the problem as BIO tagging at the span level (**TiGRO-B1**). Finally, in three of our systems, we jointly address *both* the fallacy classification at post level *and* the span detection for a total of 40 tasks (**TiGRO-A3**, **B2**, **B3**). Please note that **TiGRO-A3** and **TiGRO-B2** are exactly the same system and get a different name simply because they are used on Subtask A and Subtask B, respectively. **TiGRO-B3** differs from that model only in the pre-trained encoder used (ALBERTo vs mmBERT). While trained jointly, we do not impose any explicit dependency between the two tasks.

Encoders As base model for our systems, we experimented with several pretrained encoders. The two models used in our submissions are ALBERTo [25] and mmBERT (Multilingual Modern BERT)[26]. ALBERTo is a BERT-based model pre-trained on 191GB of Italian social media texts from the TWITA corpus [27]. mmBERT [26] is a recent multilingual encoder with 307M parameters and 22 layers, supporting more than 1,800 languages. Compared to BERT-like encoders, it introduces three main innovations: (i) a multi-stage learning paradigm composed of pre-training, mid-training, and decay phases; (ii) masked language modeling with a dynamic masking rate from 30% to 5% to enhance more nuanced language representations; (iii) dynamically varying data size and number of languages at each phase. Additionally, model temperature is dynamically adjusted to promote uniform sampling across languages and reduce bias toward high-resource ones.

In our experiments, although mmBERT is larger and more advanced, ALBERTo often performed better, likely due to its Italian specialization and Twitter pretraining, closely matching the FadeIT domain. For each submitted system we selected the encoder performing best on the development set

Training Setup We randomly split the train-dev portion released by the FadeIT shared task's organizers into training (80%) and development (20%) sets, resulting in 921 and 231 instances, respectively, corresponding to 64% and 16% of the total FAINA dataset. We used the same fixed split to train and evaluate all setups, guiding our decisions during system development. For the final submissions, we trained our best models on all available data (1152 instances).

Multi-task training was performed using the MaChAmp v0.4 toolkit [28]. Unless otherwise specified (see Table A.1 in the Appendix), models were trained using the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.99$), with a batch size of 32, learning rate of 1e-4, encoder dropout of 0.2, and for 20 epochs.

Evaluation We evaluated our systems using the official evaluator provided by the organizers and optimized our runs for micro-averaged F1 score, used for the shared task ranking. Macro-averaged scores were also used for assessing and comparing systems (see Section 4). For Subtask B, the micro-averaged metrics also take into account of partial span matches, assigning credit proportionally to the overlap in terms of tokens [20]. Additionally, the evaluation distinguishes between *strict* and *soft* modes. In the soft setting, partial credit is assigned when the predicted fallacy, albeit wrong, is nevertheless related to the gold label according to the fallacy hierarchy introduced by Ramponi et al. [8] and provided by the organizers. Although aware that soft scoringis based on class relationships, we did not take this aspect into consideration when developing our systems.

4. Results and Discussion

The TiGRO family turned out to be very competitive, ranking first (#1) for Subtask A with **TiGRO-A3** and fourth (#4) for Subtask B with **TiGRO-B3**. For the latter, the official rank is based on soft scoring; when considering strict scoring, the TiGRO systems occupy the top three positions: **TiGRO-B3**, **TiGRO-B1**, **TiGRO-B2**, respectively, showing the same internal order observed for the soft scoring rank (see Table 6). Notably, our best-performing systems for both subtasks share the same architecture, namely a multi-task fine-tuned encoder jointly trained on 20 post-level and 20 span-level classification tasks. The only difference lies in the encoders used, ALBERTo and mmBERT, respectively.

4.1. Subtask A

Table 3 reports the results for Subtask A on the development and test sets. Our best-performing model on the development set, **TiGRO-A1**, did not generalize well to the test set and exhibits significant drops in both micro and macro F1, likely due to overfitting. In contrast, our weakest model on the development set, **TiGRO-A3**, achieved the highest micro-F1 on the test set, with a score of **56.39**.

Table 3
Results for Subtask A (*post-level fallacy detection*) on development and test sets. Best result per evaluation set in bold.

Model	Development				Test				Rank
	P	R	micro-F1	macro-F1	P	R	micro-F1	macro-F1	
TiGRO-A1	57.84	57.03	**57.40**	31.58	62.52	39.85	48.65	20.57	5
TiGRO-A2	55.19	53.78	54.46	32.03	53.43	51.48	52.41	27.94	3
TiGRO-A3	49.87	58.47	53.81	32.71	53.24	59.99	**56.39**	33.35	1
Baseline	32.21	11.37	16.8	3.44	38.53	14.28	20.84	5.1	

Table 4 shows **TiGRO-A3**'s per-class scores on the test set averaged across the annotators. Performance reflects the skewed class distribution in the training set: dominant classes, such as *Appeal-to-emotion*, achieve high F1 scores (75.47), while the least frequent classes, such as *Circular-Reasoning*, which appear in less than 1% of training instances, are not predicted at all.

Furthermore, the dominant classes tend to be more based on superficial cues in the text, making it easier to capture. In contrast, the minority classes mostly correspond to complex logical fallacies, which are inherently more challenging to identify. As a result, it is difficult to reliably assess how well the models actually deal with complex fallacies. As illustrated in Example 1, superficial lexical cues (highlighted in bold) characterize a highly frequent class such as *Loaded-language* while this is not the case for a minority class such as *Strawman* as shown in Example 2.

(1) **Invasione**, sono partiti circa 900 immigrati dalla Libia: le Ong li stanno portando qui
  (en: ***Invasion**: 900 immigrants depart from Libya; NGOs are bringing them here*)

(2) [USER] propone di distribuire i vaccini "anche in base al Pil regionale". In pratica, per lei le regioni più ricche dovrebbero essere più protette dal virus rispetto a quelle più povere. Quale sarà la prossima idea? Vaccinare i bianchi e non i neri? Vergogna

(en: *[USER] proposes distributing vaccines "also based on regional GDP". In short, for her, richer regions should be better protected from the virus than poorer ones. What will the next idea be? Vaccinate white people but not black people? Shame.*)

Table 4

TiGRO-A3 per-class performance on test set, averaged across annotators, ordered by F1 score, with the percentage of training posts containing each class.

Fallacy	Precision	Recall	F1	Train (%)
Appeal-to-emotion	64.04	91.86	75.47	56.60
Loaded-language	58.33	92.16	71.26	60.68
Doubt	65.38	69.19	67.19	18.58
Name-calling-or-labelling	61.17	68.95	64.73	31.42
Slogan	72.73	57.14	64.0	11.28
Vagueness	50.0	73.94	58.9	46.35
Ad-hominem	56.9	48.53	52.37	12.15
Flag-waving	41.3	32.24	36.21	13.72
Hasty-generalization	34.38	37.71	35.91	19.53
Evading-the-burden-of-proof	29.76	33.55	31.53	17.71
Thought-terminating-cliches	28.57	26.7	27.59	11.98
Slippery-slope	37.5	18.75	25.0	7.29
False-analogy	26.92	13.95	18.18	10.85
Red-herring	17.39	14.67	15.88	11.20
Cherry-picking	25.0	7.14	11.11	5.03
Appeal-to-authority	10.0	4.95	6.6	8.42
Causal-oversimplification	10.0	3.33	5.0	5.90
Strawman	0.0	0.0	0.0	5.03
False-dilemma	0.0	0.0	0.0	3.39
Circular-reasoning	0.0	0.0	0.0	0.87

This phenomenon raises concerns about the reliability of micro F1 as an evaluation metric in highly skewed datasets. Although **TiGRO-A3** achieves a micro F1 of 56.39 on the test set, it performs poorly on most classes, highlighting how the overall score is dominated by the easiest, majority classes. To further validate this observation and assess whether the high performance on frequent classes might be due mostly to exploitable surface patterns, we trained a linear Support Vector Machine (SVM) classifier as a baseline. We used the `scikit-learn` implementation with default parameters [29] and `xlm-r-distilroberta-base-paraphrase-v1` sentence embeddings [30]. Since the test set gold labels were not released, we report results on the development set to ensure a fair comparison.

The SVM achieves a micro F1 of 47.38, notably close to **TiGRO-A3**'s 53.81 on the development set (see Table 3). However, macro F1 provides a fairer assessment of model capabilities with **TiGRO-A3** achieving 32.71 compared to only 15.04 for the SVM. More revealing insights emerge from the per-class analysis. On the development set, **TiGRO-A3** achieves a per-class F1 score of at least 50% for only 5 out of 20 fallacy classes. Notably, these are the same classes for which the SVM obtains a non-zero F1 score. Table 5 reports the development set results for these top five classes for both SVM and **TiGRO-A3**. Performance on the majority classes is similar for both models, confirming that these classes are inherently easier to detect even without complex language understanding, likely due to exploitable lexical biases in the dataset. In contrast, the SVM fails entirely on all other fallacies, whereas **TiGRO-A3** captures several mid-difficulty classes, although with limited performance. This indicates that the more complex model does identify some patterns beyond surface cues, making it difficult to reliably assess how well it actually "understands" complex fallacies.

Table 5
Performance of SVM and **TiGRO-A3** on the top five fallacy classes on development set. All remaining classes not shown achieve a F1 score of 0 for SVM and below 50% for TiGRO-A3.

Fallacy	SVM			TiGRO-A3		
	P	R	F1	P	R	F1
Doubt	89.58	54.56	67.79	73.75	74.84	74.26
Loaded-language	54.24	82.59	65.30	55.90	91.71	69.28
Appeal-to-emotion	58.03	72.59	64.47	57.52	80.43	67.04
Name-calling-or-labeling	52.56	39.87	45.33	51.35	73.85	60.57
Vagueness	50.46	68.95	57.89	48.39	75.37	58.56

Performance per Annotator During our experiments on development set we observed that the results for the second annotator are consistently higher across all setups. This is also observed on the test set, where **TiGRO-A3** achieves an F1 of 58.68% on Annotator2 and of 54.10% on Annotator1. This difference may be due to the first annotator providing noisier signals. Combining the annotations likely helped stabilize the training process and improve overall performance.

4.2. Subtask B

Table 6 present the results for Subtask B on the development and test sets. Our best model on the development set, **TiGRO-B3**, also obtained the highest performance on the test set among our submissions, with a Soft-F1 score of **44.98**. In addition, it achieved the highest Strict-F1 score among all submitted systems, reaching 42.13.

Table 6
Results for Subtask B (*span-level fallacy detection*) on development and test sets, using micro averaged scores. Best result per evaluation set in bold.

	Development						Test							
	Strict			Soft			Strict				Soft			
Model	P	R	micro-F1	P	R	micro-F1	P	R	micro-F1	Rank	P	R	micro-F1	Rank
TiGRO-B1	37.61	32.37	34.79	41.87	36.80	39.17	38.33	40.35	39.30	2	42.50	45.20	43.80	5
TiGRO-B2	37.59	33.14	35.20	41.50	37.20	39.24	38.23	40.05	39.11	3	42.68	44.87	43.74	6
TiGRO-B3	44.45	31.95	**37.18**	47.81	35.13	**40.50**	47.82	37.67	**42.13**	1	51.01	40.25	**44.98**	4
Baseline	90.49	0.77	1.53	90.49	0.77	1.53	60.94	3.05	5.80		65.97	3.28	6.25	

Table 7 shows TiGRO-B3 per-class scores on the test set, averaged across the annotators, together with span frequency and average span length in our unified version of train-dev set. In this case, span frequency appears to be less predictive of model performance, whereas span length seems to play a more important role. For instance, *Slogan* achieves strong results (F1 71.28) while representing only 2.73% of the training spans, but it has a relatively short average span length of 4.32 tokens, which may facilitate more accurate predictions.

Classes with the strongest performance in Subtask A tend to be associated with shorter spans, whereas the most difficult classes often involve much longer spans, sometimes even covering the entire post. A notable exception is *Doubt*, which performs well (F1 64.80) despite having a long average span length of 17.16 tokens and representing only 4.02% of all spans. In contrast, *Appeal-to-Authority* performs poorly (F1 12.90) despite a relatively short average span length of 6.19 tokens.

Table 7
TiGRO-B3 per-class performance on test set, averaged across annotators, ordered by F1 score, with the percentage of training spans and average span length.

Fallacy	Precision	Recall	F1	Train (%)	Avg span length
Slogan	71.67	70.93	71.28	2.73	4.32
Doubt	56.95	75.21	64.80	4.02	17.16
Appeal-to-emotion	53.94	53.93	53.90	17.85	5.74
Name-calling-or-labelling	68.51	43.67	53.27	9.84	2.71
Loaded-language	47.37	46.66	46.90	23.12	2.77
Thought-terminating-cliches	34.12	37.18	35.55	2.52	5.44
Slippery-slope	80.00	21.59	34.00	1.54	10.70
Vagueness	34.57	32.03	32.66	13.03	9.56
Flag-waving	47.22	21.62	29.66	3.46	4.66
Ad-hominem	44.87	18.25	25.91	2.62	17.07
Evading-the-burden-of-proof	28.03	10.25	15.01	3.97	16.52
Appeal-to-authority	41.67	7.69	12.90	1.90	6.19
Cherry-picking	48.48	7.14	12.45	1.03	29.21
False-analogy	26.79	7.19	11.27	2.23	20.93
Hasty-generalization	18.52	7.85	11.03	4.38	11.01
Red-herring	11.11	3.49	5.30	2.48	12.79
False-dilemma	0.00	0.00	0.00	0.80	16.96
Causal-oversimplification	0.00	0.00	0.00	1.28	20.78
Strawman	0.00	0.00	0.00	1.03	37.79
Circular-reasoning	0.00	0.00	0.00	0.18	28.90

5. Methods Not Adopted

During system development, we experimented with several alternative approaches to improve performance. While not all succeeded, discussing these attempts may benefit future work.

5.1. Data Augmentation

To address the issue of data underrepresentation, we attempted to expand the training dataset by integrating external resources. This is a well-known challenge in fallacy detection, as there is no standard taxonomy, definitions, or annotation guidelines for fallacies.

We focused on the MAFALDA dataset [31], since its fallacy classes are partially aligned with those in our task, making label mapping feasible. The dataset consists of 200 annotated English examples from different sources, including Reddit discussions, toy examples, and political debates.

We first trained the model on a multi-task setup using mmBERT (see TiGRO A2), leveraging its multilingual capabilities; however, this did not lead to noticeable performance improvements. As a second attempt, we also automatically translated the MAFALDA examples into Italian using Claude Sonnet 4.5, so that we could use the ALBERTo encoder which overall was offering better performances on our development set. This approach however proved unsuccessful to the point that we observed a decrease in performance. This degradation is likely due to (i) domain mismatch as the additional data differ substantially in style from Twitter posts, and (ii) differences in the fallacy class definitions leading to different annotation guidelines.

5.2. Span Extraction with Attribution

For addressing Subtask B, our initial approach aimed to build a performative post-level classifier for the 20 fallacy types (Subtask A), followed by span extraction using attribution methods. In other words, assuming that the classifier correctly predicts the post-level class, we can analyze which tokens the model focuses on to make that prediction and extract the most relevant ones.

Due to weak performance on Subtask A across several classes, this approach was not feasible. However, we still experimented with the highest-performing class *Appeal to Emotion* as a proof-of-concept for this idea. We used the LXT library, which provides an efficient attribution method based on AttnLRP [32], a backpropagation-based technique that corrects gradient flow through non-linearities. After computing token-level relevance scores, we selected tokens whose relevance exceeded a threshold defined as θ standard deviations above the mean. The parameter θ was tuned to maximize macro F1 on a BIO token-level evaluation for the considered class.

Overall, although the attribution method often highlights important words and can be useful as an explainability tool for human inspection, our preliminary experiments on automatic span extraction were inconclusive and yielded poor results. Some qualitative examples are reported on Figure B.1 in Appendix B. Further investigation is needed to include more classes and design sophisticated span extraction strategies that handle noisy relevance and adapt to gold span annotations.

6. Conclusions and Future Work

This paper presented the TiGRO family systems for the FadeIT task at EVALITA 2026. Our main focus was Subtask A, where TiGRO-A3 achieved first place (micro F1 of 56.39), using a multi-task approach that framed Subtask A as 20 binary classification tasks and Subtask B as 20 BIO tagging tasks, jointly training a total of 40 classifiers.

For Subtask B, our initial idea to use attribution methods to extract spans once obtained the fallacy classes from the predictions in Subtask A resulted unfeasible. This was due to poor performance on most classes in Subtask A. Instead, our submitted best model used the same multi-task approach as Subtask A, ranking fourth in soft evaluation (micro F1 44.98) and first in strict evaluation (micro F1 42.13).

Performance analysis per class revealed substantial differences across classes, reflecting the highly skewed distribution of the dataset. In addition to frequency in the training set, some classes appeared to be inherently easier, detectable even by a simple SVM, while others performed poorly or were not predicted at all, particularly the more complex logical fallacies (e.g., *Strawman* or *Circular-reasoning*), which are less likely to rely on surface clues only.

Future work should focus on extending the dataset, possibly incorporating examples from other domains to reduce biases inherent to the style of social media messages, which favors brief, provocative messages over complex arguments. Annotation could also be improved: we found that fitting to individual annotators' styles introduced noisy signals, whereas treating each annotation as equally valid significantly improved predictions. More structured guidelines with formal definitions of fallacies, following Helwe et al. [31], could improve consistency and allow the inclusion of additional annotators.

Finally, although attribution-based span extraction was not feasible in this setting, we plan to further develop this method for tasks where text-level labels are easier to obtain than precise span annotations.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly in order to: Grammar and spelling check, Text Translation, Improve writing style. After using these services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] D. Walton, Fundamentals of Critical Argumentation, Cambridge University Press, 2006.

[2] C. L. Hamblin, Fallacies, Advanced Reasoning Forum, Socorro, USA., 2022.

[3] P. Goffredo, S. Haddadan, V. Vorakitphan, E. Cabrio, S. Villata, Fallacious argument classification in political debates, in: L. D. Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 4143–4149. URL: https://doi.org/10.24963/ijcai.2022/575. doi:10.24963/ijcai.2022/575, main Track.

[4] F. Zappettini, The brexit referendum: how trade and immigration in the discourses of the official campaigns have legitimised a toxic (inter)national logic, Critical Discourse Studies 16 (2019) 403–419. URL: https://doi.org/10.1080/17405904.2019.1593206. doi:10.1080/17405904.2019.1593206. arXiv:https://doi.org/10.1080/17405904.2019.1593206.

[5] S. A. Elsayed, O. Abu-Hammad, A. B. Alolayan, Y. S. Eldeen, N. Dar-Odeh, Fallacies and facts around covid-19: The multifaceted infection, Journal of Craniofacial Surgery 31 (2020). doi:10.1097/scs.0000000000006752.

[6] H. Hansen, Fallacies, in: E. N. Zalta, U. Nodelman (Eds.), The Stanford Encyclopedia of Philosophy, Fall 2024 ed., Metaphysics Research Lab, Stanford University, 2024.

[7] U. Ecker, J. Roozenbeek, S. Van Der Linden, L. Q. Tay, J. Cook, N. Oreskes, S. Lewandowsky, Misinformation poses a bigger threat to democracy than you might think, Nature 630 (2024) 29–32.

[8] 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.

[9] P. Goffredo, S. Haddadan, V. Vorakitphan, E. Cabrio, S. Villata, Fallacious argument classification in political debates, in: Thirty-First International Joint Conference on Artificial Intelligence {IJCAI-22}, International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 4143–4149.

[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] 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.

[12] C. Helwe, T. Calamai, P.-H. Paris, C. Clavel, F. Suchanek, MAFALDA: A benchmark and comprehensive study of fallacy detection and classification, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 4810–4845. URL: https://aclanthology.org/2024.naacl-long.270/. doi:10.18653/v1/2024.naacl-long.270.

[13] T. Alhindi, T. Chakrabarty, E. Musi, S. Muresan, Multitask instruction-based prompting for fallacy recognition, 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. 8172–8187. URL: https://aclanthology.org/2022.emnlp-main.560/. doi:10.18653/v1/2022.emnlp-main.560.

[14] F. Pan, X. Wu, Z. Li, A. T. Luu, Are LLMs good zero-shot fallacy classifiers?, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural

Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 14338–14364. URL: https://aclanthology.org/2024.emnlp-main.794/. doi:10.18653/v1/2024.emnlp-main.794.

[15] Y. Li, D. Wang, J. Liang, G. Jiang, Q. He, Y. Xiao, D. Yang, Reason from fallacy: Enhancing large language models' logical reasoning through logical fallacy understanding, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 3053–3066. URL: https://aclanthology.org/2024.findings-naacl.192/. doi:10.18653/v1/2024.findings-naacl.192.

[16] E. Cantín Larumbe, A. Chust Vendrell, Argumentative fallacy detection in political debates, in: E. Chistova, P. Cimiano, S. Haddadan, G. Lapesa, R. Ruiz-Dolz (Eds.), Proceedings of the 12th Argument mining Workshop, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 369–373. URL: https://aclanthology.org/2025.argmining-1.36/. doi:10.18653/v1/2025.argmining-1.36.

[17] J. Jeong, H. Jang, H. Park, Large language models are better logical fallacy reasoners with counterargument, explanation, and goal-aware prompt formulation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Findings of the Association for Computational Linguistics: NAACL 2025, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 6918–6937. URL: https://aclanthology.org/2025.findings-naacl.384/. doi:10.18653/v1/2025.findings-naacl.384.

[18] E. Mancini, F. Ruggeri, S. Villata, P. Torroni, Overview of mm-argfallacy2025 on multimodal argumentative fallacy detection and classification in political debates, in: Proceedings of the 12th Argument mining Workshop, 2025, pp. 358–368.

[19] 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.

[20] G. Da San Martino, A. Barrón-Cedeño, P. Nakov, Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection, in: A. Feldman, G. Da San Martino, A. Barrón-Cedeño, C. Brew, C. Leberknight, P. Nakov (Eds.), Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 162–170. URL: https://aclanthology.org/D19-5024/. doi:10.18653/v1/D19-5024.

[21] E. Musi, M. Aloumpi, E. Carmi, S. Yates, K. O'Halloran, Developing fake news immunity: fallacies as misinformation triggers during the pandemic, Online Journal of Communication and Media Technologies 12 (2022).

[22] P. Röttger, B. Vidgen, D. Hovy, J. Pierrehumbert, Two contrasting data annotation paradigms for subjective NLP tasks, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 175–190. URL: https://aclanthology.org/2022.naacl-main.13/. doi:10.18653/v1/2022.naacl-main.13.

[23] Y. Mathet, The agreement measure γcat a complement to γ focused on categorization of a continuum, Computational Linguistics 43 (2017) 661–681. URL: https://doi.org/10.1162/COLI_a_00296. doi:10.1162/COLI_a_00296. arXiv:https://direct.mit.edu/coli/article-pdf/43/3/661/1808405/coli_a_00296.pdf.

[24] A. Muti, C. Emmery, D. Nozza, A. Barrón-Cedeño, T. Caselli, The "r" in "woman" stands for rights. auditing LLMs in uncovering social dynamics in implicit misogyny, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2025, Association for Computational Linguistics, Suzhou, China, 2025, pp. 5462–5479. URL: https://aclanthology.org/2025.findings-emnlp.292/. doi:10.18653/v1/2025.findings-emnlp.292.

[25] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, V. Basile, AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets, in: Proceedings of the

Sixth Italian Conference on Computational Linguistics (CLiC-it 2019), volume 2481, CEUR, 2019. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074851349&partnerID=40&md5=7abed946e06f76b3825ae5e294ffac14.

[26] M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, B. V. Durme, mmbert: A modern multilingual encoder with annealed language learning, 2025. URL: https://arxiv.org/abs/2509.06888. arXiv:2509.06888.

[27] V. Basile, M. Lai, M. Sanguinetti, et al., Long-term social media data collection at the university of turin, in: Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), CEUR-WS, 2018, pp. 1–6.

[28] R. van der Goot, A. Üstün, A. Ramponi, I. Sharaf, B. Plank, Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP, in: D. Gkatzia, D. Seddah (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2021, pp. 176–197. URL: https://aclanthology.org/2021.eacl-demos.22/. doi:10.18653/v1/2021.eacl-demos.22.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[30] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: http://arxiv.org/abs/1908.10084.

[31] C. Helwe, T. Calamai, P.-H. Paris, C. Clavel, F. Suchanek, Mafalda: A benchmark and comprehensive study of fallacy detection and classification, 2024. URL: https://arxiv.org/abs/2311.09761. arXiv:2311.09761.

[32] R. Achtibat, S. M. V. Hatefi, M. Dreyer, A. Jain, T. Wiegand, S. Lapuschkin, W. Samek, AttnLRP: Attention-aware layer-wise relevance propagation for transformers, in: R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, F. Berkenkamp (Eds.), Proceedings of the 41st International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, PMLR, 2024, pp. 135–168.

A. Additional Experimental Details

Hyper-parameters Table A.1 shows hyperparameter values for all our submitted TiGRO models. All experiments were executed on the Hábrók high-performance computing cluster at the University of Groningen, using an NVIDIA A100 GPU.

Table A.1
Training hyperparameters for TiGRO models.

Models	Encoder	Epochs	Batch size	LR	Dropout
TiGRO A1	mmBERT	3	16	2e-5	0.0
TiGRO A2	ALBERTo	10	32	1e-4	0.2
TiGRO A3, B1, B2	ALBERTo	20	32	1e-4	0.2
TiGRO B3	mmBERT	20	32	5e-4	0.2

Subtask A Table A.2 shows TiGRO-A3 confusion matrix counts on the development set for Subtask A. Figure A.1 illustrates the correlation matrix of fallacy labels in the development set and in TiGRO-A3 prediction, showing difficulty in discerning some closely related classes, such as *Slogan* and *Flag-waving*.

Table A.2

TiGRO-A3 confusion matrix counts per fallacy type on development set, ordered by true positive count.

Fallacy (tag)	TP	FP	FN	TN
Loaded-language (LL)	121	57	11	42
Appeal-to-emotion (AE)	101	52	27	51
Vagueness (VA)	74	50	30	77
Name-calling-or-labelling (NC)	44	30	18	139
Doubt (DO)	31	9	13	178
Evading-the-burden-of-proof (EP)	15	19	29	168
Hasty-generalization (HG)	15	27	24	165
Flag-waving (FW)	12	9	15	195
Slogan (SL)	11	5	19	196
Ad-hominem (AH)	10	12	16	193
Thought-terminating-cliches (TC)	7	5	26	193
Slippery-slope (SS)	4	3	12	212
False-analogy (FA)	3	14	20	194
Appeal-to-authority (AA)	3	1	23	204
Red-herring (RH)	2	15	17	197
Causal-oversimplification (CO)	1	3	15	212
False-dilemma (FD)	0	0	7	224
Circular-reasoning (CR)	0	0	3	228
Cherry-picking (CP)	0	1	12	218
Strawman (ST)	0	2	13	216
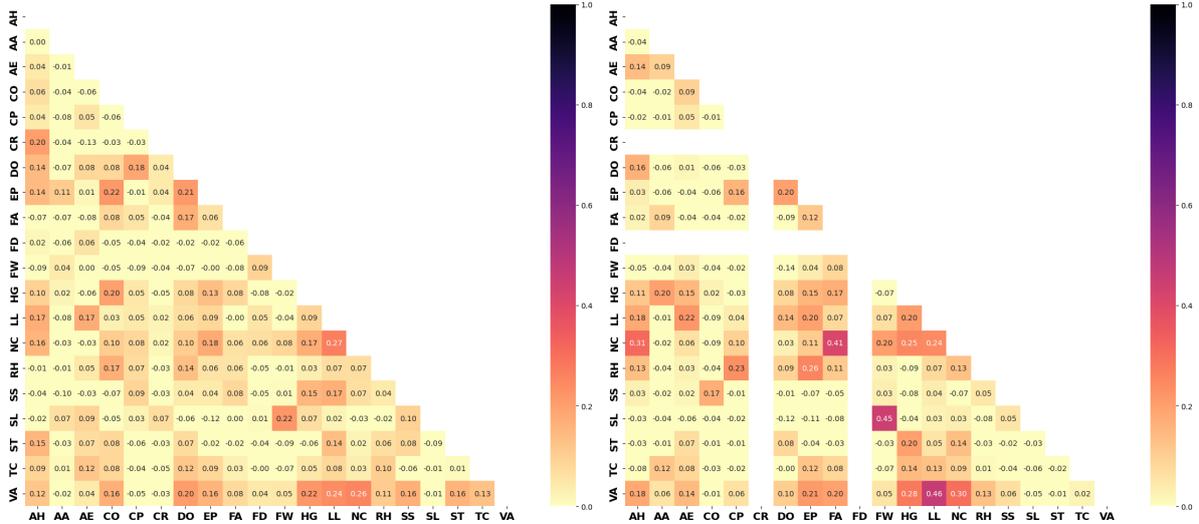
(a) Correlation matrix for gold labels in dev-set.   (b) Correlation matrix for predicted labels (TiGRO-A3).

Figure A.1: Correlation of fallacy label co-occurrences in the development set and in TiGRO-A3 predictions, considering merged annotations.

B. Attribution Examples

The examples in Figure B.1 illustrate two true positive predictions and one false positive prediction from TiGRO-A1 for *Appeal to Emotion*, highlighting words based on attribution relevance. Specifically, red indicates a positive contribution to the predicted class, and blue indicates a negative contribution. The gold span is highlighted in **bold**.

Results are noisy, but the model generally focuses on emotion-related words supporting correct predictions for Subtask A. Example B.1a shows a perfect gold span match for Subtask B, while B.1b illustrates typical mismatches. This mainly occurs because the model assigns relevance to single key

words rather than longer spans, and is sometimes distracted by words that contribute to post-level predictions but introduce noise for exact span extraction. Example B.1c shows a false positive, where attribution can be useful for human inspection.

(a) Example of perfect match with the gold span for Subtask B.

(b) Example of mismatch with the gold spans for Subtask B.

(c) Example of false positive, where attribution shows which textual portion has most likely led to the wrong class prediction.

Figure B.1: Attribution examples for TiGRO-A1 predictions on *Appeal to Emotion*. Red indicates positive contribution, Blue negative, and the gold span is in **bold**.