ArguMind at Touché: A Unified Multi-Task Transformer Approach for Fallacy and Argument Scheme Classification

Touché at CLEF 2026

Sevitha Simhadri[1,*,†], Madduri Chandan[2,2,3,†]

[1]Department of Information Technology, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India
[2]Department of Electronic Systems, Indian Institute of Technology Madras, Chennai 600036, India
[3]Department of Electronics and Communication, International Institute of Information Technology Hyderabad 500032, India

Abstract

In this paper, an approach for fallacy detection, fallacy classification, and argument scheme classification is presented for the Touché 2026 Fallacy Detection shared task. Two complementary systems are investigated: a traditional machine learning pipeline based on TF-IDF word and character n-gram features with LinearSVC classifiers, and a multi-task DeBERTa-v3-xsmall architecture with a shared encoder and task-specific prediction heads. Both systems operate on enhanced argument representations constructed from rewritten argument text, extracted claims, and supporting statements. Experimental results show that the multi-task DeBERTa model achieves the strongest performance across the subtasks, while the TF-IDF + LinearSVC approach remains a strong sparse-feature baseline. Overall, the results indicate that contextual multi-task modeling is particularly effective for capturing both fallacious reasoning and argument scheme structure in informal online arguments.

Keywords

Fallacy Detection, Argument Mining, Multi-Task Learning, DeBERTa, Transformer Models, Argument Scheme Classification, Computational Argumentation

1. Introduction

The automatic identification of fallacious reasoning has become an important problem in computational argumentation and natural language processing. Fallacies are errors in reasoning that weaken the validity of an argument, and their detection is relevant for applications such as misinformation analysis, debate understanding, and automated fact-checking. At the same time, fallacies are difficult to detect automatically because they often resemble valid argumentative patterns and rely on implicit assumptions and contextual cues.

The Touché 2026 Fallacy Detection task provides a structured benchmark for this problem through three related subtasks [1, 2]. The first subtask focuses on determining whether an argument is fallacious or non-fallacious. The second requires identifying the specific fallacy type for fallacious arguments. The third addresses non-fallacious arguments and requires predicting the corresponding argument scheme through the two dimensions of argument goal and argument basis. Since the dataset is derived from informal Reddit discussions, the task is further complicated by noisy language, context dependence, and the frequent overlap between flawed and valid reasoning patterns.

Two systems are developed and evaluated. The first uses TF-IDF word and character n-gram features with LinearSVC classifiers. The second uses a multi-task DeBERTa-v3-xsmall architecture with a shared encoder and task-specific prediction heads [3]. Both systems are trained on enhanced argument representations, which make the argumentative structure and reasoning cues more explicit. The experimental results show that the multi-task DeBERTa model achieves the strongest performance across the subtasks, while the TF-IDF + LinearSVC approach remains a competitive sparse-feature baseline.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
†These authors contributed equally.
munnysimhadri5544@gmail.com (S. Simhadri); maddurichandan@gmail.com (M. Chandan)

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the task and dataset. Section 4 describes the baseline and multi-task neural systems. Section 5 reports the experimental results. Section 6 concludes the paper.

2. Literature Survey

Fallacy detection has become an important topic in computational argumentation and natural language processing because flawed reasoning in online discussions is often subtle, context dependent, and difficult to identify through surface-level cues alone. This makes the task closely related to argument mining, where the goal is not only to understand text but also to recover the underlying structure and reasoning pattern of an argument.

Early work by Palau and Moens [4] established a foundation for automatic argument detection and structural analysis in text. Later, Lawrence and Reed [5] provided a broad survey of argument mining methods, emphasizing the importance of identifying argument components, relations, and schemes for deeper reasoning analysis.

More recent work has focused directly on fallacy detection. Jin et al. [6] studied logical fallacy detection using structure-aware modeling strategies, showing that argumentative reasoning cannot be captured adequately through lexical cues alone. Alhindi et al. [7] explored cross-domain fallacy recognition with a multitask prompting framework, demonstrating the value of shared learning across related fallacy datasets. Goffredo et al. [8] further showed that combining transformer representations with explicit argument structure features improves fallacy classification, especially when premise–conclusion interactions are important.

Taken together, these studies suggest that effective fallacy analysis benefits from both contextual semantic modeling and explicit reasoning structure. This motivates the use of enhanced argument representations and a multi-task transformer architecture in the present work.

3. Overview of Tasks and Dataset

The Touché 2026 Fallacy Detection shared task [1, 2] is designed around argumentative Reddit conversations and combines binary fallacy detection, fine-grained fallacy categorization, and argument scheme classification within a unified benchmark.

3.1. Task Definition

Three subtasks are defined, each targeting a distinct aspect of argumentative reasoning:

- **Sub-Task 1 — Fallacy Detection:** Determine whether a given argument is fallacious or non-fallacious.
- **Sub-Task 2 — Fallacy Classification:** Identify the specific fallacy type from eight categories: *authority*, *black-white*, *hasty_generalization*, *natural*, *population*, *slippery_slope*, *tradition*, and *worse_problems*.
- **Sub-Task 3 — Argument Scheme Classification:** Classify a non-fallacious argument along two orthogonal dimensions — *argument_goal* (Practical or Epistemic) and *argument_basis* (Internal or External) — yielding four combined labels: *practical-internal*, *practical-external*, *epistemic-internal*, and *epistemic-external*.

3.2. Dataset Specifics

The dataset is derived from the informal fallacies corpus of Sahai et al. [9], consisting of Reddit arguments annotated for fallacy existence, fallacy type, and argumentation scheme dimensions. The training split contains 1,440 instances equally divided between fallacious and non-fallacious arguments, with each of the eight fallacy categories represented by 90 instances. Two levels of processed fields

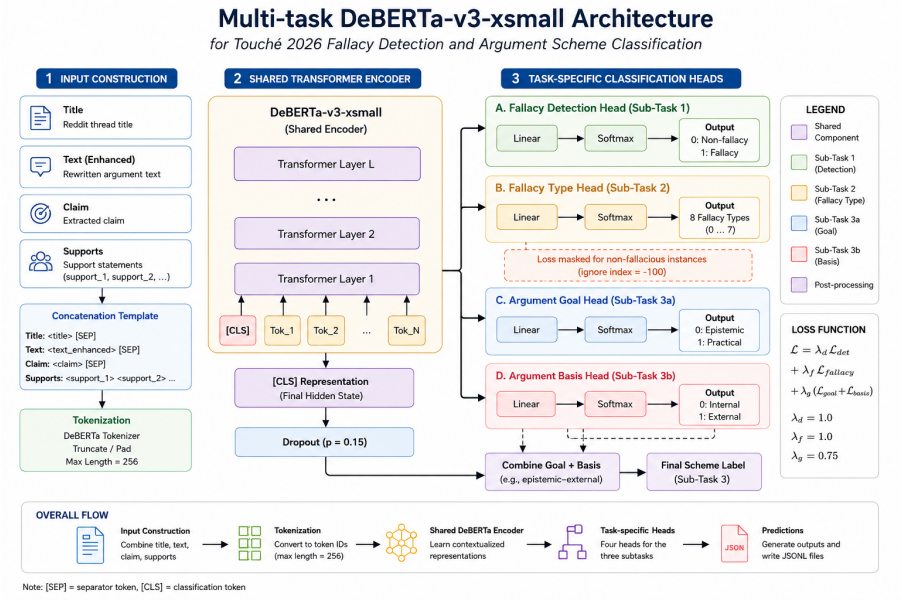

Figure 1: Architecture of the multi-task DeBERTa-v3-xsmall framework.

are provided: *base* fields offer self-contained argument rewrites incorporating surrounding context, while *enhanced* fields additionally expose fallacy and scheme information explicitly. Argument scheme dimensions follow the framework of Macagno [10]. The main dataset fields are summarized in Table 1.

Table 1
Main fields in the Touché 2026 Fallacy Detection dataset.

Field	Description
`text_raw`	Original Reddit comment.
`text_raw_parent`	Parent comment of `text_raw`.
`text_raw_title`	Title of the Reddit thread.
`text_base`	Self-contained argument rewrite with context.
`argument_base`	{claim, supports} from `text_base`.
`text_enhanced`	Rewrite exposing fallacy and scheme information.
`argument_enhanced`	{claim, supports} from `text_enhanced`.
`fallacy_exists`	Binary fallacy label (Sub-Task 1).
`fallacy_type`	Specific fallacy category (Sub-Task 2).
`resembles_fallacy`	Most similar fallacy type for non-fallacious entries.
`classification`	{argument_goal, argument_basis} (Sub-Task 3).

4. System Overview

Two main modeling directions are explored in this work. The first uses traditional machine learning with TF-IDF representations and LinearSVC classifiers, while the second uses a neural multi-task architecture based on DeBERTa-v3-xsmall [3]. The two approaches serve different purposes in the final system. The TF-IDF + LinearSVC pipeline provides strong and interpretable baselines, whereas the multi-task DeBERTa model achieves the strongest submitted results across all three subtasks. This section describes the preprocessing pipeline, the baseline system, the proposed multi-task transformer model, and the final prediction strategy.

The architecture of the multi-task neural framework is illustrated in Figure 1. Since the TF-IDF + LinearSVC baseline follows a standard sparse-feature classification pipeline, a separate figure is not strictly necessary for that system.

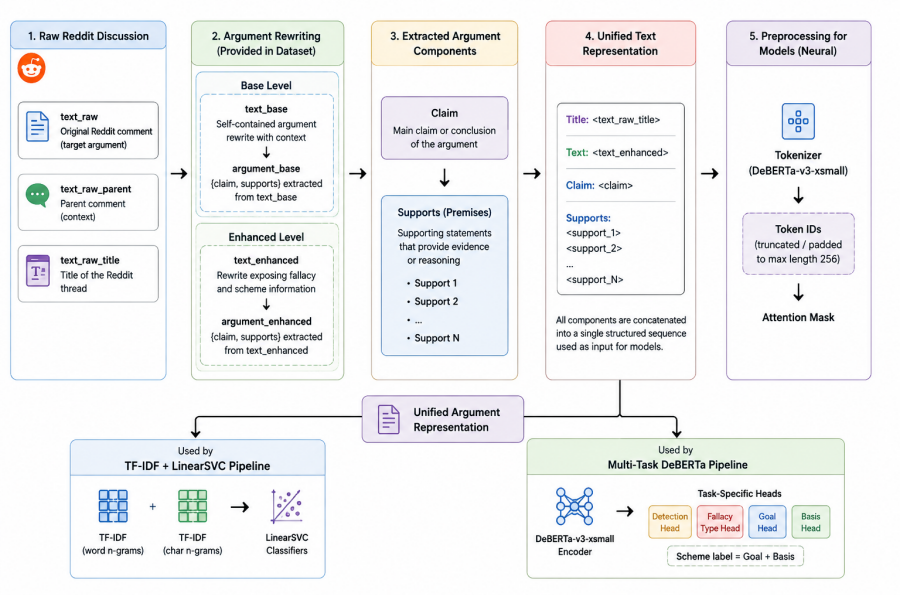

Figure 2: Pipeline for constructing unified argument representations from Reddit discussion fields and enhanced argument annotations.

4.1. Data Preprocessing

Each dataset instance contains several textual fields, including the original Reddit comment, contextual metadata, and processed argument representations. The experiments focus on the processed *base* and *enhanced* fields, since these provide self-contained reformulations of the original argument and reduce the ambiguity typical of informal online discussions. The final systems rely primarily on the *enhanced* fields, as these make argumentative structure and reasoning cues more explicit. Figure 2 illustrates the preprocessing pipeline used to construct the unified argument representations shared across both modeling approaches.

For each instance, a unified textual representation is constructed by concatenating the thread title, rewritten argument text, extracted claim, and supporting statements in a structured format:

```
Title: <text_raw_title>
Text: <text_enhanced>
Claim: <claim>
Supports: <support_1> <support_2> ...
```

This representation is used in both the baseline and neural systems so that the same argumentative content is available across modeling settings. A minor label normalization step is also applied by mapping the variant `blackwhite` to the canonical label `black-white` in order to match the official task label set.

For the neural model, the constructed sequence is tokenized with the tokenizer associated with `microsoft/deberta-v3-xsmall` and truncated or padded to a maximum sequence length of 256 tokens.

4.2. Baseline System: TF-IDF with LinearSVC

A traditional machine learning pipeline is implemented as a baseline and serves as a strong comparison system. The unified textual representation described above is vectorized using a combination of word-level and character-level TF-IDF features.

Word-level TF-IDF features use unigrams and bigrams and are intended to capture lexical and phrase-level patterns associated with fallacious reasoning and argument structure. Character-level TF-IDF features use character n-grams of length 3 to 5 and provide robustness against spelling variation,

punctuation differences, and stylistic irregularities common in Reddit discussions. Both feature sets are combined through `FeatureUnion` and passed to class-balanced `LinearSVC` classifiers [11].

For Sub-Task 3, two separate `LinearSVC` classifiers are trained independently for `argument_goal` and `argument_basis`, and their predictions are combined to form the final scheme label. This decomposition follows the annotation design of the task and is more structured than directly predicting the four combined scheme labels. In the submitted runs, however, the transformer model outperformed this baseline on all three subtasks.

4.3. Proposed System: Multi-Task DeBERTa

The proposed neural model is based on `DeBERTa-v3-xsmall` and is designed as a multi-task architecture with a single shared encoder and four task-specific prediction heads. The shared transformer encoder produces contextualized representations of the input argument, allowing common reasoning patterns to be learned across the subtasks while still preserving task-specific outputs.

The contextual representation associated with the first input position is extracted from the final hidden layer, passed through a dropout layer, and then forwarded to four independent linear classification heads. These heads are responsible for the following predictions:

- **Detection head:** binary fallacy detection for Sub-Task 1.
- **Fallacy type head:** eight-class fallacy classification for Sub-Task 2.
- **Argument goal head:** prediction of *epistemic* vs. *practical*.
- **Argument basis head:** prediction of *internal* vs. *external*.

The final argument scheme label for Sub-Task 3 is obtained by combining the predictions of the argument goal and argument basis heads. This design reflects the task definition directly and allows argument scheme classification to be modeled as the interaction of two simpler decision problems.

4.4. Multi-Task Training Objective

Training is performed using a weighted multi-task objective in which cross-entropy loss is computed separately for each prediction head. Since fallacy type labels are only defined for fallacious instances, the fallacy classification loss is masked for non-fallacious examples. This prevents the model from learning invalid fallacy-type assignments where no fallacy is present.

The combined training objective is defined as:

$$\mathcal{L} = \lambda_d \mathcal{L}_{\text{detection}} + \lambda_f \mathcal{L}_{\text{fallacy}} + \lambda_s \left(\mathcal{L}_{\text{goal}} + \mathcal{L}_{\text{basis}} \right), \tag{1}$$

where λ_d, λ_f, and λ_s denote the loss weights for fallacy detection, fallacy type classification, and argument scheme prediction, respectively. This formulation makes it possible to emphasize binary fallacy detection while still retaining fine-grained supervision for the other tasks.

4.5. Training Setup

The multi-task DeBERTa model is fine-tuned on the full training split using enhanced argument representations. Optimization is performed with AdamW [12], together with a linear learning-rate schedule and warmup. Class-weighted losses are used to reduce the effect of label imbalance, and gradient clipping is applied for training stability.

The stronger final configuration increased the number of training epochs and assigned greater weight to the detection loss in order to improve fallacy detection performance while preserving strong results for scheme classification. The main hyperparameters for the final neural configuration are summarized in Table 2.

Table 2
Hyperparameters used for the final multi-task DeBERTa configuration.

Hyperparameter	Value
Transformer Model	DeBERTa-v3-xsmall
Maximum Sequence Length	256
Training Batch Size	8
Evaluation Batch Size	16
Training Epochs	10
Learning Rate	3×10^{-5}
Weight Decay	0.01
Dropout	0.15
Warmup Ratio	0.08
Gradient Clipping	1.0
Optimizer	AdamW
Detection Loss Weight	2.0
Fallacy Loss Weight	1.1
Scheme Loss Weight	0.9
Random Seed	19
CPU Threads	4

Table 3
Official evaluation results on the Touché 2026 Fallacy Detection held-out test set using enhanced representations.

Sub-Task	System	Precision	Recall	F1
Sub-Task 1 — Fallacy Detection	TF-IDF + LinearSVC	0.884	0.884	0.884
	Multi-Task DeBERTa	**0.931**	**0.926**	**0.927**
Sub-Task 2 — Fallacy Classification	TF-IDF + LinearSVC	0.957	0.953	0.954
	Multi-Task DeBERTa	**0.960**	**0.955**	**0.956**
Sub-Task 3 — Scheme Classification	TF-IDF + LinearSVC	0.880	0.880	0.880
	Multi-Task DeBERTa	**0.958**	**0.998**	**0.976**

4.6. Inference and Final Submission Strategy

During inference, each test instance is processed through the shared DeBERTa encoder and predictions are generated independently by the four classification heads via argmax. The `argument_goal` and `argument_basis` predictions are concatenated with a hyphen to form the final scheme label for Sub-Task 3. Separate JSONL files are produced for each subtask in the format required by the shared task.

The final submitted runs are selected according to empirical performance. The multi-task DeBERTa model is the final submitted system for all three subtasks, while the TF-IDF + LinearSVC pipeline is retained as a strong classical baseline for comparison. This final setup reflects the fact that a shared contextual encoder is more effective for the complete task suite, even though sparse lexical features remain a useful reference point.

5. Experimental Results

This section presents the official evaluation results on the held-out test set for the submitted runs. The runs were evaluated through the shared-task infrastructure on TIRA [13]. All results reported here use the enhanced argument representations. Performance is summarized in Table 3 using precision, recall, and F1.

The updated results show that the multi-task DeBERTa model performs best on all three subtasks,

although the margin on Sub-Task 2 is very small. For Sub-Task 1, the multi-task model clearly outperforms the TF-IDF + LinearSVC baseline, suggesting that contextual representations help distinguish fallacious from non-fallacious arguments more effectively than sparse lexical features alone.

Sub-Task 2 remains comparatively close, with both systems performing strongly on the enhanced input representations. The multi-task DeBERTa model still has a slight edge, which indicates that fine-grained fallacy classification benefits from contextual modeling, even though surface-level cues remain highly informative for this task.

Sub-Task 3 shows the largest gain for the neural model. The multi-task DeBERTa architecture achieves substantially higher F1 than the TF-IDF + LinearSVC baseline, confirming that argument scheme classification is especially sensitive to richer contextual semantics and the shared modeling of argument goal and argument basis.

Overall, the results indicate that the enhanced argument representations are effective for both sparse-feature and neural models, but the multi-task transformer provides the strongest and most consistent performance across the benchmark. The TF-IDF + LinearSVC system remains a competitive baseline, yet the final evaluation suggests that contextual multi-task learning is the more effective strategy for this shared task.

6. Conclusion and Future Work

This paper presented two complementary approaches for the Touché 2026 Fallacy Detection shared task: a TF-IDF + LinearSVC pipeline and a multi-task DeBERTa-v3-xsmall architecture. The experimental results showed that the multi-task DeBERTa model achieved the strongest performance across all three subtasks on the enhanced test setting, while the TF-IDF-based system remained a competitive classical baseline. These findings indicate that contextual multi-task learning is particularly effective for modeling both fallacious reasoning and argument scheme structure in informal online arguments.

For future work, it would be valuable to explore larger transformer architectures, stronger multi-task optimization strategies, and alternative ways of encoding argument structure. Further gains may also come from cross-domain evaluation and from incorporating more explicit modeling of claim-support relations, especially for closely related fallacy categories and scheme labels.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-5 in order to generate images for the paper figures and assist with language polishing. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] 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.

[2] 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.

[3] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: Proceedings of the 11th International Conference

on Learning Representations (ICLR 2023), 2023, pp. 1–26. URL: https://openreview.net/forum?id=sE7-XhLxHA.

[4] R. M. Palau, M.-F. Moens, Argumentation mining: The detection, classification and structure of arguments in text, in: Proceedings of the 12th International Conference on Artificial Intelligence and Law, 2009, pp. 98–107. doi:10.1145/1568234.1568246.

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

[6] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schölkopf, Logical fallacy detection, in: Findings of the Association for Computational Linguistics: EMNLP 2022, 2022, pp. 7180–7198. doi:10.18653/v1/2022.findings-emnlp.532.

[7] T. Alhindi, T. Chakrabarty, E. Musi, S. Muresan, Multitask instruction-based prompting for fallacy recognition, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 8172–8187. doi:10.18653/v1/2022.emnlp-main.560.

[8] P. Goffredo, M. C. Gamboa, S. Villata, E. Cabrio, Argument-based detection and classification of fallacies in political debates, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 10902–10914. doi:10.18653/v1/2023.emnlp-main.684.

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

[10] F. Macagno, Argumentation profiles and the manipulation of common ground. The arguments of populist leaders on Twitter, Journal of Pragmatics 191 (2022) 67–82. doi:10.1016/j.pragma.2022.01.001.

[11] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: Proceedings of the 10th European Conference on Machine Learning (ECML 1998), 1998, pp. 137–142. doi:10.1007/BFb0026683.

[12] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), 2019, pp. 1–19. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[13] 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.