FALCON: A multi-label graph-based dataset for fallacy classification in the COVID-19 infodemic

Mariana Chaves, Elena Cabrio, Serena Villata

FALCON: A multi-label graph-based dataset for fallacy classification in the COVID-19 infodemic

Mariana Chaves
Université Côte d'Azur, CNRS, Inria
Sophia Antipolis, France
machaves@i3s.unice.fr

Elena Cabrio
Université Côte d'Azur, CNRS, Inria
Sophia Antipolis, France
elena.cabrio@univ-cotedazur.fr

Serena Villata
Université Côte d'Azur, CNRS, Inria
Sophia Antipolis, France
serena.villata@cnrs.fr

ABSTRACT

Fallacies are arguments that seem valid but contain logical flaws. During the COVID-19 pandemic, they played a role in spreading misinformation, causing confusion and eroding public trust in health measures. Therefore, there is a critical need for automated tools to identify fallacies in media, which can help mitigate harmful narratives in future health crises. We present two key contributions to address this task. First, we introduce FALCON, a multi-label, graph-based dataset containing COVID-19-related tweets. This dataset includes expert annotations for six fallacy types—*loaded language*, *appeal to fear*, *appeal to ridicule*, *hasty generalization*, *ad hominem*, and *false dilemma*—and allows for the detection of multiple fallacies in a single tweet. The dataset's graph structure enables analysis of the relationships between fallacies and their progression in conversations. Second, we evaluate the performance of language models on this dataset and propose a dual-transformer architecture that integrates engineered features. Beyond model ranking, we conduct statistical analyses to assess the impact of individual features on model performance.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; • **Applied computing** → **Annotation**.

KEYWORDS

natural language processing, fallacious argumentation, text classification, language models, transformer models

ACM Reference Format:

Mariana Chaves, Elena Cabrio, and Serena Villata. 2025. FALCON: A multi-label graph-based dataset for fallacy classification in the COVID-19 infodemic. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31-April 4, 2025, Catania, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3672608.3707913

1 INTRODUCTION

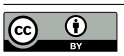
The COVID-19 pandemic was accompanied by an *infodemic*—an overwhelming flood of information, including false or misleading content, spreading rapidly during a disease outbreak, as defined by the World Health Organization.[1] Within this landscape, fallacies were a common mechanism for disseminating disinformation, misinformation, and propaganda. Fueling skepticism about vaccines, promoting ineffective treatments, and undermining trust in public health guidelines are just a few examples of how fallacies have been used to manipulate public opinion and behavior.

A fallacy is an argument that appears valid but contains logical flaws [16, 21]. Fallacies are not necessarily false statements; rather, they are arguments that fail to provide valid support for their conclusions [22]. This distinction matters because fallacies often contain elements of truth, making them particularly challenging for audiences to detect. In the context of social media, the fast-paced and information-saturated environment, combined with algorithms that prioritize engagement over accuracy, can make it hard for users to evaluate each piece of content critically. Identifying fallacies helps users spot misleading narratives, make informed decisions, and promote public health.

However, merely detecting the presence or absence of a fallacy is often insufficient for users. To better understand the flawed logic, we need to identify the specific type of fallacy present. For example, consider the statement: "I saw several people who wore masks still getting COVID-19, so wearing masks doesn't work at all." This statement commits a *hasty generalization* by using a small, anecdotal set of observations to make a broad conclusion about the effectiveness of masks. While it is not false that some people who wore masks contracted COVID-19, the conclusion that masks "don't work at all" is unwarranted based on this limited evidence. Knowing the type of fallacy helps users understand the exact nature of the logical flaw. Moreover, different fallacies can co-occur within the same piece of text. Therefore a multi-label classification approach, where multiple fallacies can be identified within the same sequence of text, offers a more comprehensive understanding than traditional multi-class classification. Most existing approaches to fallacy detection and classification frame the problem as a multi-class classification task, where only one fallacy label is assigned to each data point [2, 3, 15, 16, 19, 25, 34, 36, 38, 40]. This is partly due to the limitations of many available datasets, which permit only a single fallacy label per instance. In contrast, we approach it as a multi-label classification task.

Our contribution is two-fold. First, we present the FALCON (Fallacies in COVID-19 Network-based) dataset, a collection of tweets[2] related to the COVID-19 pandemic and politically associated discussions annotated with 6 fallacy categories: *loaded language*, *appeal to fear*, *appeal to ridicule*, *hasty generalization*, *ad hominem*, and *false dilemma*. Annotations are provided at the tweet level and in a multi-label format, meaning that a tweet can be associated with

[1] https://www.who.int/health-topics/infodemic.



SAC '25, March 31-April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03.
https://doi.org/10.1145/3672608.3707913

[2] When the data was extracted, the platform was called Twitter instead of X. Therefore, we refer to the posts as tweets.

more than one fallacy category. The dataset includes an underlining graph structure that can be used to model the relationships between the fallacies. To the best of our knowledge, this is the first dataset to offer multi-label human-expert fallacy annotations for tweets related to the COVID-19 pandemic. Second, we use our dataset to evaluate the performance of a set of language models on the task of multi-label fallacy classification. Among these, we propose a transformer-based architecture that leverages non-textual engineered features and context information related to the tweet, extracted from the graph structure of the dataset.[3]

2 RELATED WORKS

Corpora. Several datasets have been developed to study fallacies in text. For example, `Argotario` [19] (five fallacy types) is a dataset derived from the game of the same name. `LOGIC` and `LOGICCLIMATE` [26] (13 fallacy types) include logical fallacies collected from online educational materials and climate change news articles, respectively. `ElecDeb60To20` [15] (six fallacy types) comprises political debates from United States presidential candidates. Habernal et al. [20] (only *ad hominem*) and Sahai et al. [34] (eight fallacy types) created datasets by mining Reddit. Payandeh et al. [32] proposed a dataset containing over 5,000 pairs of logical and fallacious arguments from debates generated by Large Language Models (LLMs) on controversial topics.

Other datasets provide annotations for propaganda techniques, including certain fallacy types. For example, `PTC-SemEval20` [12] consists of news articles, `TWEETSPIN` [38] is based on tweets, and the Reddit-based dataset provided by Balalau and Horincar [4] includes information from six major political forums in the US and UK. Closer to our work, Musi et al. [30] present a dataset that compiles news articles on COVID-19. Finally, datasets like `MAFALDA` [22] and the one proposed by Alhindi et al. [2] were created by merging existing fallacy datasets, with the latter also proposing `Climate`, a dataset of climate change articles fact-checked by scientists.

Fallacy Classification. Several studies have addressed fallacy classification at the text snippet or token level using machine learning models. Initially, Habernal et al. [19] employed SVM and BiLSTM models on the `Argotario` dataset, achieving a macro F1 score of 42.1% (6 classes). Later studies mainly employed transformer-based models. For example, Da San Martino et al. [12] introduced several models in the SemEval-2020 Task 11 challenge for detecting propaganda techniques, with the 10 best-performing models utilizing a transformer architecture. Many of those combine outputs from multiple transformers with engineered features, with the best model achieving a 63.4% macro F1 score (14 classes). Goffredo et al. [15, 16] proposed transformer-based architectures that integrate text, argumentative features, and engineered features, achieving macro F1 scores of 84.0% and 73.9% (7 classes) at the text snippet and token levels, respectively. Vorakitphan et al. [40] developed a transformer-based pipeline that, when combined with semantic and argumentative features, achieved a 64.0% macro F1 score (14 classes). Sahai et al. [34] used a fine-tuned BERT model for token-level fallacy classification on Reddit comments, achieving a macro F1 score of 53.4%. Their findings also indicated improved model performance when the conversation context of a post was included. Sourati et al. [36] combined LLMs with explainable methods based on prototype reasoning, instance-based reasoning, and knowledge injection to classify fallacies, achieving macro F1 scores of 82.7% and 57.3% on the `LOGIC` and `LOGICCLIMATE` datasets (13 classes), respectively. Vijayaraghavan and Vosoughi [38] utilized a transformer-based model incorporating additional features (*e.g.* context and relational information) to classify propaganda in tweets, achieving a 63.7% F1 score (19 classes).

Some approaches focused on transforming text into logical forms to distill argumentative structures; for instance, Jin et al. [26] proposed a structure-aware classifier based on a pre-trained Natural Language Inference (NLI) model, achieving macro F1 scores of 58.8% and 29.4% (13 classes) on the `LOGIC` and `LOGICCLIMATE` datasets, respectively. Similarly, Lalwani et al. [27] translated natural language into First-Order Logic (FOL) using LLMs and then applied Satisfiability Modulo Theory (SMT) solvers to classify fallacies, achieving F1 scores of 71.0% and 73.0% in two binary classification tasks (fallacious vs. non-fallacious).

More recently, research has explored the use of transformers in zero-shot, few-shot, and full-shot scenarios for fallacy classification. For instance, Helwe et al. [22] used various versions of Falcon, LLAMA2, Mistral, Vicuna, WizardLM, Zephyr, and GPT-3.5 in zero-shot and few-shot settings on the `MAFALDA` dataset for binary classification, three broad fallacy categories, and 23 fallacy types, achieving F1 scores of 62.7%, 20.1%, and 13.8%, respectively. Alhindi et al. [2] experimented with different prompts and zero, few, and full-shot scenarios using the T5 model on the `PTC-SemEval20`, `LOGIC`, `Argotario`, `COVID-19`, and `Climate` datasets, achieving F1 scores of 56%, 66%, 64%, 28%, and 20%, respectively. In subsequent work, Alhindi et al. [3] improved on this same setting by leveraging GPT-3.5 to generate examples to increase the representation of rare classes and incorporate additional contextual information.

In summary, the F1 scores vary significantly across different datasets, even when the same model architecture is applied. While some variability can be attributed to different class counts within the corpora, models can produce vastly different results even with the same set of fallacies. This highlights how dataset characteristics, such as text complexity or domain specificity, can influence model performance. Moreover, most of the models mentioned share common characteristics: they are predominantly transformer-based, often enhanced by additional features such as argumentative or semantic structures, context, and external knowledge.

Challenges of Fallacy Annotation. By their very nature, fallacies are challenging for humans to recognize and classify. Helwe et al. [22] reported that human subjects achieved an F1 score of 35.2% when classifying fallacies into three broad categories, and a mere 18.6% for identifying 23 finer fallacy types. This complexity makes fallacy annotation especially costly, often limiting the size of datasets. The ones listed above contain hundreds or a few thousands of data points, except for `TWEETSPIN`, which reaches 157,395 entries using automated processes instead of human-expert annotations. Also, the annotation process for fallacies is subjective [22]. To address these challenges, Helwe et al. [22] introduced a scheme that allows for multiple equally valid annotations for the same text span. While the dataset produced includes high-quality multi-label

[3]Code and data available at https://github.com/m-chaves/falcon-fallacy-classification.

annotations, the difficulties mentioned above resulted in a rather reduced count of 203 data points.

Our work extends previous studies on fallacy annotation and classification. We present the first dataset providing multi-label expert annotations of fallacies in tweets related to the COVID-19 pandemic. The dataset contains 2,916 tweets, each annotated with one or more labels from seven categories (including six fallacy types and a non-fallacious class), substantially increasing the availability of multi-label fallacy data. The dataset offers a graph-based representation, facilitating the linking of data points, extraction of contextual information, and modeling of relationships between fallacies. Second, while prior studies have advocated for using contextual information and engineered features [12, 15, 16, 34, 38, 39, 39] to improve fallacy classification, they have largely relied on ranking models to assess the impact of these features. In contrast, we employ statistical tests to evaluate their effects on model performance. Finally, while most existing approaches focus on multi-class classification, where each snippet or token is limited to a single fallacy label, we propose transformer-based architectures for multi-label fallacy classification, allowing each tweet to be associated with multiple fallacies.

3 DATASET

In this section, we present the FALCON (Fallacies in COVID-19 Network-based) dataset. This is a collection of tweets related to the COVID-19 pandemic and politically associated discussions annotated with six fallacy categories: *loaded language*, *appeal to fear*, *appeal to ridicule*, *hasty generalization*, *ad hominem*, and *false dilemma*. Annotations are provided at the tweet level and in a multi-label format, meaning that a tweet can be associated with more than one fallacy category.

3.1 Data Collection and Preprocessing

The dataset was created using a collection of tweets web-scraped by the Barcelona Supercomputing Center [14][4]. They collected the data via the Twitter (X) API from March 25, 2020, to March 25, 2021, and focused on topics related to the COVID-19 pandemic and politically associated discussions (*e.g.*, army mobilizations during the pandemic). For example, some of the keywords used for extraction were "covid," "azithromycin," "ivermectin," "bleach," and "vaccine." Some of those keywords are related to conspiracy theories and misinformation about the COVID-19 pandemic. For instance, azithromycin and ivermectin were promoted as treatments for COVID-19 despite the lack of scientific evidence, leading some to advocate for the use of these drugs instead of vaccines, masks, and lockdowns. This made this dataset suitable for studying fallacies. From the data collection, we extracted variables related to the tweet's text, user (*e.g.*, username, number of followers), engagement (*e.g.*, number of retweets, replies, and likes), context (*e.g.*, identifiers that indicate whether a tweet is a reply and allow the retrieval of the original tweet and its text), hashtags, and mentions.

The main steps for data cleaning included filtering for tweets in English, removing non-ASCII characters and URLs, and anonymizing usernames by replacing them with unique IDs. Notably, we retained emojis as those often convey emotions, so they can help detect fallacies such as *loaded language*, *appeal to fear*, and *appeal to ridicule*. We also retained regional indicator symbols since those are used to represent flag emojis and, thus, can assist in identifying fallacies related to stereotyping, generalization, and national bias.

[4]This work describes only part of the data collection.

3.2 Graph-based Processing

3.2.1 Context Information Extraction. Our analysis considered the context of each tweet because it can be decisive in identifying fallacies. That is, we considered the conversation or thread in which the tweet is inserted. More generally, contextual information tends to improve the accuracy of both human and machine learning-based annotations [12, 15, 16, 34]. To properly capture the richness of the contextual information in the data, we modeled the dataset as a directed graph. This choice allowed us to effectively handle the relationships between the data points when further analyzing and processing the data.

Formally, given a set of vertices (data points) V, we consider a directed graph $G = (V, A)$ where $A \subseteq V \times V$. Given two vertices $u, v \in V$, we say there is an arc from u to v if and only if the ordered pair (u, v) belongs to A. In such a case, we say that u is a parent, or in-neighbour, of v, and that v is a child, or out-neighbour, of u. Notice that we employ ordered pairs rather than sets, so $(u, v) \neq (v, u)$.

We started with a graph of tweets $G_{\text{raw}} = (V_{\text{raw}}, A_{\text{raw}})$ where given tweets $u, v \in V_{\text{raw}}$ we have an arc from u to v if and only if v replies to, quotes, or retweets u.

A central issue we faced was that while G_{raw} has many data points ($|V_{\text{raw}}| = 4,184,314$), many of them are duplicates in that they contain very similar or identical text. The main reasons for this are (i) retweets, as those contain the same text as the original tweet prepended by a string indicating the author of the original tweet; (ii) tweets that contain more than one of the extraction keywords, as, for example, a tweet that contains both "covid" and "azithromycin" in their text would be captured twice by the API; and (iii) repeated tweets by the same user, sometimes mentioning different users but keeping the rest of the text identical. To preserve the structural information of the graph, we resolved duplicates by merging the information of the tweets instead of simply removing them.

When merging two tweets, we kept the data (namely, the text, timestamp, and user-related and engagement metrics) of the oldest one (timestamp-wise) and took the union of their arcs. That is, after merging duplicate tweets u and u', those are replaced by a new vertex v containing the attributes of the oldest among u and u' and such that v has an arc to (from) w if and only if either u or u' had an arc to (from) w.

The merge of duplicates yields a new graph $G_{\text{merged}} = (V_{\text{merged}}, A_{\text{merged}})$ with a much reduced number of vertices, namely $|V_{\text{merged}}| = 382,581$. On the other hand, the merge process can increase the neighborhood of vertices. While G_{merged} is still quite sparse, some vertices can have many neighbors. In particular, vertices associated with popular tweets in G_{raw} would have many children, ending up with an even larger neighborhood in G_{merged}. Vertices with high connectivity can be problematic in defining an exact notion of context. Indeed, we captured the context of a tweet by considering its first and second-degree neighbors: Its children, grandchildren (children of children), parents, and grandparents (parents of parents). However, for tweets corresponding to highly

Fallacy type	Definition	Example
Loaded language	The use of words and phrases with strong connotations (either positive or negative) to influence an audience and invoke an emotional response [16, 42].	It's just idiotic to think he meant for people to go out and buy hypodermics and inject themselves.
Appeal to fear	Eliciting fear to support a claim [16, 41].	#thegreatreset #Agenda21 all our freedoms are been erased, loss of private property, do you still think this is all about a virus??
Appeal to ridicule	Presenting an opponent's argument as absurd, ridiculous, or humorous. Mocking the opponent's point of view [7].	The COVID guidelines be like "Make sure you touch a coffee cup with three fingertips when lowering your mask or the virus will mutate."
Hasty generalization	Making a broad statement about a group or population based on a limited or unrepresentative sample. It usually follows the form: X is true for A, X is also true for B, therefore, X is true for C, D and E [34, 42].	Ivermectin KILLS BAD #COVID-19 IN 2-6 DAYS: my 90-yro Aunt, on edge of intubation, ICU, got rid of it in 5 days; feeling better after 1. Get it approved!
False dilemma	Presenting a situation as having only two alternatives, when in reality there are more options available. It oversimplifies a complex issue by reducing it to only two possible outcomes or choices, often in a way that excludes other possibilities, nuances, or middle-ground [11, 12, 34].	Don't let people die in hospitals from COVID-19 when #ivermectin is available.
Ad hominem	Attacking the person or some aspect of the person making the argument rather than addressing the argument itself [20, 41, 42].	What kind of a fool would even consider testing the injection of disinfectants? Let alone say that it might be interesting to try it. Ignorance abounds in the Chump cult.

Table 1: Fallacy definitions and examples.

connected vertices, this rule can lead to contexts with hundreds of tweets. We prevented such excessively large contexts by restricting it to a maximum of six tweets, selected based on their temporal proximity to the main tweet.

3.2.2 Graph Clustering and Pruning. Despite its advantages, using context can induce train/test contamination. For instance, naïvely partitioning the data at the level of individual tweets can give the model access to test data during training as a train tweet could end up with test tweets in its context. To avoid such scenarios, we ensured that all tweets in a conversation or thread were assigned to the same part. In terms of the graph structure, this means that we should split the graph at the level of components, where a (connected) component is a maximal subgraph in which any two vertices are connected by a path. G_{merged} contains 128,661 components, most of them with only a few vertices. However, the distribution of component sizes is highly skewed as the largest component contains 23.1% of the vertices while the second-largest is only 0.4%. Assigning the largest component entirely to one of the sets could bias the results, so we designed a method to split it. We remark that naïve approaches, such as the removal of vertices with the largest degree or at random vertices, destroy too much information before having any significant impact.

Ideally, we would partition connected components by identifying good vertex separators[5]. Alas, this task is known to be difficult to solve precisely, NP-hard, in fact [1, 8]. Thus, we instead approached the problem heuristically, employing graph clustering techniques to identify communities in the component and pruning the graph to disconnect the communities found. The pruning method searches for pairs of nodes that share an arc with nodes outside their community. Once such a pair is found, the method removes the node with the smallest neighborhood, the one belonging to the largest community, at random, in increasing order of priority.

We considered three graph clustering methods: Clauset-Newman-Moore greedy modularity maximization [9], the fluid communities algorithm [31], and the Louvain method [6]. The latter was the most consistent in identifying communities across different runs. When applied to the largest component of G_{merged} and combined with the pruning method, the technique split the component into 809 sub-components while removing only 409 nodes.

Finally, we observed that some components within the dataset contained mostly tweets unrelated to the COVID-19 pandemic. To remove these irrelevant components, we employed topic modeling and Natural Language Processing techniques. Specifically, we used Latent Dirichlet Allocation (LDA) [5], BERTopic [17], and TF-IDF [35] to identify key n-grams (up to trigrams) that corresponded to topics outside our scope of our study. We filtered out components with high TF-IDF scores on those n-grams. 453 components were removed using this method.

The processes described above resulted in a graph G^*, containing 273,947 vertices, 144,646 arcs, and 127,689 connected components.

[5]In graph theory, a subset $S \in V$ is a vertex separator (or vertex cut, separating set) for non-adjacent vertices u and v if the removal of S from the graph leaves u and v into distinct connected components.

Fallacy	Count	Cohen's Kappa
Hasty generalization	91 (3.12%)	0.46
Appeal to fear	157 (5.38%)	0.81
False dilemma	168 (5.76%)	0.55
Appeal to ridicule	238 (8.16%)	0.77
Ad hominem	259 (8.88%)	0.79
Loaded language	457 (15.67%)	0.56
None of the above	1907 (65.40%)	0.72

Table 2: Per class statistics. The percentage Count is relative to the size of the full dataset. The Cohen's Kappa was computed over a sample of 50 tweets.

3.3 Fallacy Annotation

We divided our annotation process into a pilot and a final annotation stage. We randomly sampled five components (containing 325 tweets) for the pilot stage and 1398 components (containing 2916 tweets) for the final stage.

We selected the fallacy types by their relative prevalence in the existing works of Da San Martino et al. [12], Goffredo et al. [15], Habernal et al. [18, 20], Jin et al. [26], Musi et al. [30], Sahai et al. [34], Vijayaraghavan and Vosoughi [38]. Initially, we listed 10 fallacies for the pilot annotation stage. Based on the pilot, we shortlisted six fallacies for the final annotation process: *loaded language*, *appeal to fear*, *appeal to ridicule*, *hasty generalization*, *ad hominem*, and *false dilemma*. Notoriously, while *appeal to ridicule* is less explored in previous works, we opted to include it since it is recurrent in our dataset. Table 1 shows the definitions and examples of the selected fallacies.

The expert annotators were two members of the research team. There were two rounds of discussion to establish clear guidelines and resolve discrepancies. Annotations were conducted on the Label Studio platform [37], which presented users with a *main tweet* and up to six of its *context tweets*. They were instructed to identify fallacies from our predetermined list in the *main tweet*, or to select "none of the above" if no fallacy was present.

3.4 Dataset Statistics

The final dataset contains 2,916 tweets, 1,009 of which contain at least one fallacy. 708 feature a single fallacy, 250 contain two, 42 have three, and 9 include four. Table 2 shows the distribution of fallacies across the dataset. The dataset was split by components, with 60% allocated to training, 20% to validation, and 20% to testing. To assess the co-occurrence of fallacies, we computed the correlation between fallacy types. The values were generally low, with the highest being 0.21 between *loaded language* and *ad hominem*, based on 103 co-occurrences. Inter-annotator agreement was measured using Cohen's Kappa [10]. Table 2 shows the kappa values for each fallacy category, with an average value of 0.67, indicating substantial agreement [28].

The graph structure of the dataset allows to analyze the influence of fallacies over subsequent tweets. Figure 1 shows the likelihood of different types of fallacies following one another in sequential tweets. For example, if a tweet contains an *ad hominem* fallacy, the following tweet is more likely to contain an *ad hominem* fallacy (17.3% of the times) or use *loaded language* (20.5% of the times),

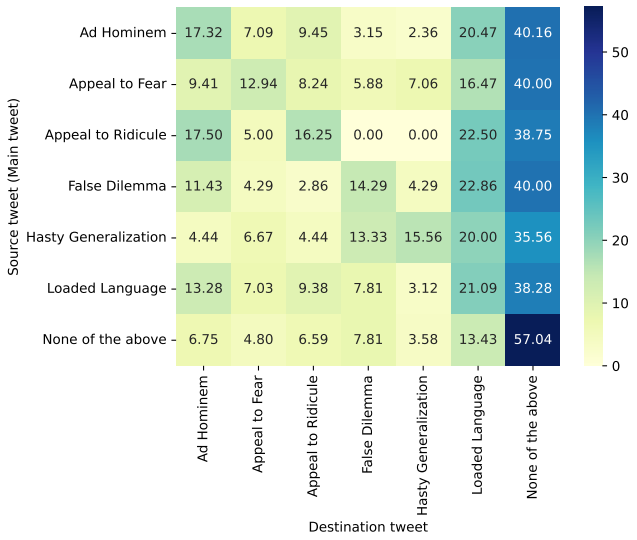


Figure 1: Stochastic (Markov) matrix of fallacy transitions.

rather than a *false dilemma* fallacy which happened only 3.2% of the times. Notoriously, the last column of the figure indicates that tweets containing any type of fallacy are more likely to receive a fallacious reply.

We performed statistical tests to verify if this visual analysis could be generalized. More specifically, our analysis aimed to answer two key questions: (i) are tweets containing fallacies more likely to be followed by fallacious replies than those without fallacies? (ii) if so, are tweets with certain types of fallacies more prone to get fallacious replies?

To address the first question, we performed a one-tailed Z-test for two proportions, comparing the proportion of replies containing a fallacy between tweets that were fallacious and those that were not. Tweets containing fallacies were significantly more likely to be followed by fallacious replies. For the second question, we conducted a one-sided Z-test for proportions to evaluate whether the likelihood of a fallacious reply was greater than random chance (*i.e.*, greater than 0.5) for each fallacy type. Given the multiple comparisons involved, we applied the Bonferroni correction. The results were significant only for tweets containing *loaded language*, indicating that such tweets are likely to receive a fallacious reply. However, there was insufficient evidence to support similar conclusions for other fallacy types.

4 MULTI-LABEL FALLACY CLASSIFICATION

We evaluate the performance of different language models on the task of multi-label fallacy classification using the proposed dataset. We test encoder-based models (*e.g.*, BERT [13]) and a sequence-to-sequence model (T5) and we also explore the impact of including additional features, such as context information, hashtags, mentions, emojis, and sentiment scores, on the performance of the models. Our main evaluation metric is the macro F1 score (averaged across three runs), as it is a good measure for imbalanced datasets.

Model	Avg. macro F1 score %
Classic `microsoft/deberta-v3-base`	47.8
Classic `roberta-base`	47.3
Classic `elozano/tweet_emotion_eval`	45.1
Dual `microsoft/deberta-v3-base`	47.2
Dual `roberta-base`	44.0
Dual `elozano/tweet_emotion_eval`	44.0
Dual `microsoft/deberta-v3-base` + sentiment scores	**48.8**
Dual `microsoft/deberta-v3-base` + emojis	**48.5**
Dual `microsoft/deberta-v3-base` + all engineered features	47.3
T5 (Prompt type: list of fallacies + NotA)	16.0
T5 (Prompt type: list of fallacies)	15.9
T5 (Prompt type: fallacy definitions)	11.4

Table 3: Average (across three runs) macro F1 scores for the top three performing models of each type. "NotA" indicates if the prompt instructed the model to return "none of the above" when no fallacies were detected.

4.1 Models

This subsection describes the 4 groups of models we tested: classic transformer, dual transformer, dual transformer with engineered features, and T5 models. Table 3 shows the average macro F1 scores for the top three performing models of each group.

4.1.1 Classic Transformer Models. We fine-tuned several models from Hugging Face's Transformers library [43]. More specifically, we used the following checkpoints: `bert-base-uncased`, `distilbert-base-uncased`, `albert-base-v2`, `roberta-base`, `microsoft/deberta-v3-base`, `elozano/tweet_emotion_eval`, `m-newhauser/distilbert-political-tweets`, `Kev07/Toxic-Tweets`, and `jariasf/bert-tweets-covid`. For this class of models, the input consists of the text of the *main tweet*, only. The maximum sequence length was set to 128 as it covers the maximum length of the tweets in the dataset. The best-performing models attained an average macro F1 score of 47.8%.

4.1.2 Dual Transformer Models. Besides classic transformers, we employed dual-transformer architectures, which consist of two instances of the same type of pre-trained transformers, one for processing the *main tweet* and the other for the *context information*. Maximum sequence lengths of 128 and 512 were used respectively. The context data consists of the concatenation of the *context tweets* and the *main tweet* in chronological order. The last hidden states of the two transformers are concatenated and passed through a classification head. We fine-tuned the same checkpoints as in the classic transformer models, with the best-performing model reaching an average macro F1 score of 47.2%.

4.1.3 Dual Transformer Models with Engineered Features. Despite the extra resources, the dual-transformer models did not outperform the classic ones. To investigate this, we added extra engineered features to the two dual-transformer models. We concatenated the additional features with the output of the transformers before feeding them into the classification head. For fine-tuning, we utilized the `microsoft/deberta-v3-base` and `roberta-base` checkpoints, as these models demonstrated the best performance in our previous experiments. We experimented with different combinations of the proposed features, described below.

Emojis, Mentions, and Hashtags. Our descriptive analysis revealed that mentions of certain public figures often involved in controversial topics were more likely to be linked with *ad hominem* attacks. Moreover, emojis can aid in identifying the intention behind a tweet which is relevant for fallacy categories related to emotions. For example, various laughing emojis were commonly found in tweets that contained *appeal to ridicule* fallacies. To capture those elements, we considered binary features representing the most frequent hashtags, mentions, and emojis in our dataset. That is, if the *main tweet* contains a specific hashtag, mention, or emoji, the corresponding feature is set to 1; otherwise, it is set to 0.

Sentiment Scores. To assess emotional content, we used two sentiment scoring systems: Valence Aware Dictionary and sEntiment Reasoner (VADER) [24] and the Valence, Arousal, and Dominance (VAD) lexicon [29] on the *main tweet*. VADER provides multidimensional sentiment scores at the document (tweet) level and considers the effect of capital letters, punctuation, and emojis. The VAD lexicon provides word-level valence, arousal, and dominance scores, which we averaged across each tweet.

Part-of-Speech Tags. Part-of-speech (POS) tagging was included as a feature to provide syntactic information about the text. We used the spaCy implementation [23] and represented POS tags as counts of each tag within the *main tweet*.

In our experiments, using only sentiment scores was the best-performing combination (average macro F1 score of 48.8%).

4.1.4 T5 Models. We also experimented with the Text-To-Text Transfer Transformer (T5) model [33] using the `t5-large` checkpoint. Specifically, we used the T5 model settings proposed by Alhindi et al. [2] since the authors achieved good results in their fallacy classification task. In contrast to the previous models, when using the T5 model every task is cast in a text-to-text format. That is, its input is a text string that includes a specific prompt indicating the task type, followed by the text to be processed. The output model is also a text string. As Alhindi et al. [2], we tested prompts that included the definitions of the fallacies, and prompts that only listed the names of the fallacies. Additionally, we experimented with prompts that explicitly indicated the model to return "none of the above" if no fallacies were detected, and to render multiple fallacies if appropriate. We used full-shot fine-tuning, evaluating outputs by converting them into binary vectors indicating which fallacy types were mentioned in the text. This is a more relaxed approach than that of Alhindi et al. [2], which uses strict string matching. The best performing T5 model used the prompt that only listed the fallacy names (without the definitions) and indicated the use of "none of the above", reaching a macro F1 score of 16.0%, considerably lower than the encoder-based models.

4.2 Ablation Analysis

We conducted statistical tests to compare the macro F1 scores across different groups of models, evaluating the impact of various engineered features on model performance. First, we compared classic transformer models against dual transformer models to assess the

Features						Avg.
Context information	Hashtags	Mentions	Emojis	Sentiment scores	POS tags	Avg. macro F1 score %
✓				✓		48.8
✓			✓			48.5
						47.8
✓	✓	✓	✓	✓	✓	47.3
✓						47.2
✓					✓	46.4
✓	✓	✓	✓			45.5
✓		✓				41.3
✓	✓					41.2

Table 4: Average macro F1 scores (in descending order) of models using the `microsoft/deberta-v3-base` checkpoint with different combinations of features. Scores are based on the average of three runs.

effect of incorporating context information. Next, we compared models utilizing one of the engineered features against those that did not use that feature (*e.g.*, models that used sentiment scores against those that did not). Depending on the data's normality and homoscedasticity, we used either a t-test, Welch's t-test, or Mann-Whitney U test. In all cases, the results indicated no significant differences between the groups of models. This suggests that none of the features individually provided a substantial improvement in model performance. Therefore, although the best-performing model incorporated sentiment scores, we cannot conclude that this feature was the sole reason for its success.

Additionally, Table 4 presents the results obtained from different combinations of features using `microsoft/deberta-v3-base` as the backbone model. Using context information and sentiment scores together achieved the highest average macro F1 score. The second best was the combination of context information and emojis. However, the third-best model used none of the added features and outperformed the model that used the combination of all features. This shows additional evidence that the engineered features provide only marginal improvements in model performance.

4.3 Error Analysis

Table 5 shows the classification report of the best-performing model. The micro and weighted F1 scores indicated reasonably good performance, with values of 73.6% and 71.6%. However, the macro F1 score (50.3%) reveals that the model's performance is uneven across different classes, especially for those with fewer samples like *hasty generalization* and *false dilemma*. *Loaded language* and *none of the above* are predicted reasonably well, with balanced values in precision and recall. Nevertheless, *ad hominem*, *appeal to fear*, *appeal to ridicule*, *false dilemma*, and *hasty generalization* are likely to be under-predicted, as indicated by their low recall. Particularly, the model struggles to predict *hasty generalization* instances.

It is worth noticing the potential for confounding factors, particularly in cases of the *false dilemma* fallacy. In our dataset, this fallacy frequently appears in the context of debates contrasting vaccines, lockdowns, and the use of masks with azithromycin, hydroxychloroquine, and ivermectin. The model might inadvertently learn to predict a false dilemma based on the presence of these specific terms rather than understanding the underlying logical structure of a false dilemma.

Class	Precision %	Recall %	F1 Score %	Support
Ad Hominem	72.7	53.3	61.5	45
Appeal to Fear	61.9	44.8	52.0	29
Appeal to Ridicule	40.9	39.1	40.0	46
False Dilemma	57.1	28.6	38.1	28
Hasty Generalization	50.0	07.7	13.3	26
Loaded Language	59.3	58.7	59.0	92
None of the above	85.6	91.0	88.2	366
Micro avg.	75.8	71.5	73.6	632
Macro avg.	61.1	46.2	50.3	632
Weighted avg.	73.8	71.5	71.6	632
Samples avg.	76.1	75.5	75.0	632

Table 5: Test performance metrics for the best model across classes: a dual transformer using `microsoft/deberta-v3-base` and sentiment scores. Metrics refer to the best of three runs.

5 CONCLUDING REMARKS

We introduced the FALCON (Fallacies in Covid-19 Network-based) dataset, consisting of 2,916 tweets related to the COVID-19 pandemic and politically associated topics. This dataset provides multi-label, human-expert annotations for six categories of fallacies. By modeling the dataset as a graph, we captured the contextual information embedded in tweet interactions, offering a richer understanding of how fallacies propagate within online discussions.

Our empirical analysis demonstrates that language models can be utilized for multi-label fallacy classification. Encoder-based architectures outperformed sequence-to-sequence models, with the dual-transformer architecture incorporating context information and sentiment scores achieving the highest performance. Nevertheless, the complexity of the task still poses a challenge for this model, with an average macro F1 score of 48.8%. The model's performance varied by class; it was most effective in identifying the absence of fallacies, achieving 88.2% macro F1 score in the *none of the above* class, and showed reasonable success with the *loaded language* (59.0%) and *ad hominem* (61.5%) fallacies. However, it struggled with fallacies with fewer examples in the dataset, such as *hasty generalization* (13.3%). We conducted statistical tests to evaluate the impact of features such as sentiment scores and emoji usage. Our analysis found that neither context information nor engineered features led to statistically significant improvements in model performance. This indicates that the success of the best-performing model cannot be solely attributed to these features.

Our findings suggest that tweets containing fallacies are statistically more likely to receive fallacious replies, particularly those involving *loaded language*. This highlights how certain fallacies propagate in online discussions, underscoring the need for more effective methods to detect fallacious reasoning in social media.

6 ACKNOWLEDGEMENTS

This work was supported by EU Horizon 2020 project AI4Media (https://ai4media.eu/), under contract no. 951911, and the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

REFERENCES

[1] Gaurav Aggarwal, Sreenivas Gollapudi, Raghavender, and Ali Kemal Sinop. 2021. Sketch-based Algorithms for Approximate Shortest Paths in Road Networks. In *Proceedings of the Web Conference 2021 (WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3918–3929.

[2] Tariq Alhindi, Tuhin Chakrabarty, Elena Musi, and Smaranda Muresan. 2022. Multitask Instruction-based Prompting for Fallacy Recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 8172–8187.

[3] Tariq Alhindi, Smaranda Muresan, and Preslav Nakov. 2023. Large Language Models are Few-Shot Training Example Generators: A Case Study in Fallacy Recognition. *arXiv preprint arXiv:2311.09552* (2023).

[4] Oana Balalau and Roxana Horincar. 2021. From the Stage to the Audience: Propaganda on Reddit. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, Online, 3540–3550.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[6] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (Oct. 2008), P10008.

[7] Gregory L. Bock. 2018. Appeal to Ridicule. In *Bad Arguments*. John Wiley & Sons, Ltd, 118–120.

[8] Thang Nguyen Bui and Curt Jones. 1992. Finding good approximate vertex and edge partitions is NP-hard. *Inform. Process. Lett.* 42, 3 (May 1992), 153–159.

[9] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical Review E* 70, 6 (Dec. 2004), 066111. Publisher: American Physical Society.

[10] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[11] Jennifer Culver. 2018. False Dilemma. In *Bad Arguments*. John Wiley & Sons, Ltd, 346–347.

[12] Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. International Committee for Computational Linguistics, Barcelona (online), 1377–1414. https://doi.org/10.18653/v1/2020.semeval-1.186

[13] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Dmitry Gnatyshak, Dario Garcia Gasulla, Sergio Álvarez Napagao, Jamie Arjona Martínez, and Tommaso Venturini. 2022. Healthy Twitter discussions? Time will tell. (2022).

[15] Pierpaolo Goffredo, Mariana Chaves, Serena Villata, and Elena Cabrio. 2023. Argument-based Detection and Classification of Fallacies in Political Debates. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 11101–11112.

[16] Pierpaolo Goffredo, Shohreh Haddadan, Vorakit Vorakitphan, Elena Cabrio, and Serena Villata. 2022. Fallacious Argument Classification in Political Debates. In *Thirty-First International Joint Conference on Artificial Intelligence {IJCAI-22}*. International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, 4143–4149.

[17] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

[18] Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational Argumentation Meets Serious Games. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Copenhagen, Denmark, 7–12.

[19] Ivan Habernal, Patrick Pauli, and Iryna Gurevych. 2018. Adapting Serious Game for Fallacious Argumentation to German: Pitfalls, Insights, and Best Practices. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan.

[20] Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. Before Name-Calling: Dynamics and Triggers of Ad Hominem Fallacies in Web Argumentation. In *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, 386–396.

[21] Charles Leonard Hamblin. 2022. Fallacies. Advanced Reasoning Forum.

[22] Chadi Helwe, Tom Calamai, Pierre-Henri Paris, Chloé Clavel, and Fabian Suchanek. 2024. MAFALDA: A Benchmark and Comprehensive Study of Fallacy Detection and Classification. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 4810–4845.

[23] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. 2020. spaCy: Industrial-strength natural language processing in python. (2020).

[24] C. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media* 8, 1 (May 2014), 216–225.

[25] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large Language Models on Graphs: A Comprehensive Survey. _eprint: 2312.02783.

[26] Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schölkopf. 2022. Logical Fallacy Detection.

[27] Abhinav Lalwani, Lovish Chopra, Christopher Hahn, Caroline Trippel, Zhijing Jin, and Mrinmaya Sachan. 2024. NL2FOL: Translating Natural Language to First-Order Logic for Logical Fallacy Detection. *arXiv preprint arXiv:2405.02318* (2024).

[28] Mary L. McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica* 22, 3 (2012), 276–282. Place: Croatia.

[29] Saif M. Mohammad. 2018. Obtaining Reliable Human Ratings of Valence, Arousal, and Dominance for 20,000 English Words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*. Melbourne, Australia.

[30] Elena Musi, Myrto Aloumpi, Elinor Carmi, Simeon Yates, and Kay O'Halloran. 2022. Developing Fake News Immunity: Fallacies as Misinformation Triggers During the Pandemic. *Online Journal of Communication and Media Technologies* 12, 3 (May 2022), e202217. Publisher: Bastas.

[31] Ferran Parés, Dario Garcia Gasulla, Armand Vilalta, Jonatan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. 2018. Fluid communities: A competitive, scalable and diverse community detection algorithm. In *Complex Networks & Their Applications VI: Proceedings of Complex Networks 2017 (The Sixth International Conference on Complex Networks and Their Applications)*. Springer, 229–240.

[32] Amirreza Payandeh, Dan Pluth, Jordan Hosier, Xuesu Xiao, and Vijay K Gurbani. 2024. How Susceptible Are LLMs to Logical Fallacies?. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 8276–8286.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21 (2020), 1–67.

[34] Saumya Sahai, Oana Balalau, and Roxana Horincar. 2021. 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)*. Association for Computational Linguistics, Online, 644–657.

[35] Gerard Salton and Chris Buckley. 1987. *Term weighting approaches in automatic text retrieval*. Technical Report. Cornell University.

[36] Zhivar Sourati, Vishnu Priya Prasanna Venkatesh, Darshan Deshpande, Himanshu Rawlani, Filip Ilievski, Hông-Ân Sandlin, and Alain Mermoud. 2023. Robust and explainable identification of logical fallacies in natural language arguments. *Knowledge-Based Systems* 266 (2023), 110418.

[37] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2022. Label Studio: Data labeling software. https://github.com/heartexlabs/label-studio

[38] Prashanth Vijayaraghavan and Soroush Vosoughi. 2022. TWEETSPIN: Fine-grained Propaganda Detection in Social Media Using Multi-View Representations. In *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, 3433–3448.

[39] Vorakit Vorakitphan, Elena Cabrio, and Serena Villata. 2021. "Don't discuss": Investigating Semantic and Argumentative Features for Supervised Propagandist Message Detection and Classification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. INCOMA Ltd., Held Online, 1498–1507.

[40] Vorakit Vorakitphan, Elena Cabrio, and Serena Villata. 2022. Protect: A pipeline for propaganda detection and classification. In *CLiC-it 2021-Italian Conference on Computational Linguistics*. 352–358.

[41] Douglas N Walton. 1987. *Informal fallacies*. John Benjamins Publishing Company.

[42] Anthony Weston. 2018. *A Rulebook for Arguments*. Hackett Publishing.

[43] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Qun Liu and David Schlangen (Eds.). Association for Computational Linguistics, Online, 38–45.