Logical Fallacy Detection Using Case-Based Reasoning

Anju Vilashni Nandhakumar, Aswatth Ganapathy Subramanian, Navin Prasath Selvakumar, Nikita Vinod Mandal

Northeastern University, Boston MA

April 23, 2024

Abstract

Logical fallacies are common pitfalls in reasoning found within written or spoken arguments, rendering them invalid or unsound. These errors can take many different forms, such as erroneous reasoning, superfluous details, or dishonest language manipulation. This research uses cutting edge natural language processing (NLP) techniques to automatically detect logical fallacies. We create machine learning models to recognize different kinds of erroneous reasoning patterns in textual arguments by utilizing annotated datasets. To effectively identify fallacies, our method combines feature extraction, text preprocessing, and model training. We illustrate the efficacy of our approach with comprehensive tests on various datasets. This project aims to classify fallacies in each paragraph into the following categories: The Slippery slope, Bandwagon appeal, Hasty generalization, Post Hoc Ergo Propter Hoc, Genetic Fallacy, Begging the claim, Circular argument, Either /Or, Ad Hominem, Red herring, Straw Man and Moral equivalence. We will conduct experiments using multiple pretrained models, including BERT[1] and RoBERTa[2], to determine which one achieves superior performance. Our goal is to identify the model that yields the highest accuracy for the task. Additionally, depending on the time constraint, we would like to experiment with practical problems since this paradigm has the ability to counteract disinformation in public discourse, education, law, and media while also strengthening critical thinking and argument quality.

Keywords— Logical fallacy, Case-Based Reasoning,

1 Introduction

Logical fallacies are errors in reasoning that weaken the validity of arguments. These errors can take many forms, such as appealing to emotions rather than evidence (ad hominem, bandwagon appeal), or attacking the source of an argument rather than the argument itself (ad hominem, genetic fallacy). Logical fallacies are prevalent in written and spoken discourse, from political speeches and social media posts to news articles and academic papers.[3] They can have a negative impact on our ability to evaluate arguments and make sound decisions.

Being able to identify and avoid logical fallacies is essential for critical thinking. Critical thinking is a complex cognitive skill that involves the ability to analyze information, evaluate arguments, and form sound judgments. It is an essential skill for success in academic life, professional life, and personal life. By being aware of logical fallacies, we can learn to identify them in the arguments we encounter and avoid being misled by them. This is especially important in today's world, where we are constantly bombarded with information from a variety of sources, often through social media and online news outlets.[3]

The spread of misinformation and disinformation is a major problem in today's society. Misinformation is false or inaccurate information that is spread unintentionally, while disinformation is false or inaccurate information that is spread intentionally to deceive. Logical fallacies are a common tool used to spread misinformation and disinformation. By being able to identify logical fallacies, we can become more discerning consumers of information and help to slow the spread of misinformation and disinformation.[3]

Furthermore, being able to identify logical fallacies can help us to improve the quality of our own arguments. When we are aware of the different types of logical fallacies, we can avoid making them ourselves. We can also learn to construct arguments that are sound and well-reasoned. This is important for success in a variety of

contexts, such as academic debates, professional presentations, and even everyday conversations.

In conclusion, there are many reasons why being able to identify and avoid logical fallacies is important. Logical fallacies can weaken the validity of arguments, make it difficult to evaluate information, and hinder our ability to think critically. By learning to identify logical fallacies, we can become more discerning consumers of information, improve the quality of our own arguments, and promote critical thinking and reasoned argumentation.

2 Methods

This project tackles automatic logical fallacy classification using a Case-Based Reasoning (CBR)[4] approach with Language Models (LMs) as core components. CBR[4] is a powerful technique for dealing with new problems by leveraging past experiences stored in a case base. Our CBR[4] system consists of three main stages: retrieval, adaptation, and classification. (Figure 1).

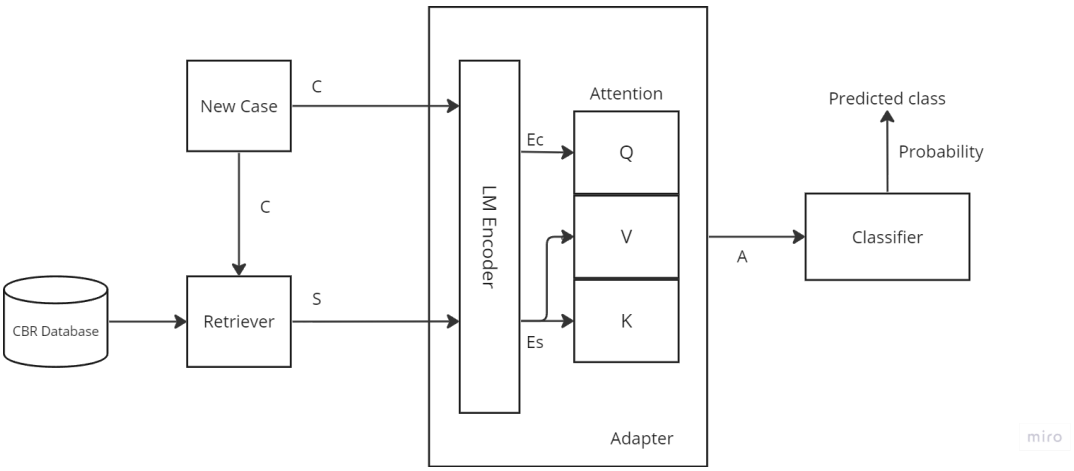

Figure 1: Case-Based Reasoning Workflow

2.1 Modules

1. **Retriever:**

   The retrieval stage identifies the most similar cases (Si) from a pre-existing case base to the new, unseen argument (C) we want to classify. To achieve this, we employ a Retriever module that utilizes an LM encoder. Both the new argument (C) and the stored cases (Si) are encoded into numerical representations by the LM encoder. We then calculate the cosine similarity between these encoded representations to determine their level of similarity. The Retriever retrieves the k cases (Si) with the highest cosine similarity scores to the new argument (C). Finally, the new argument is concatenated with these retrieved similar cases to form a combined input for the next stage.

2. **Adapter:**

   The adapter component refines the retrieved similar cases (Si) to better match the context of the new argument (C). It consists of two main parts:

   (a) **Encoder:**

   This is a linear model that takes both the new argument (C) and the retrieved similar cases (Si) as inputs. It processes them and generates their corresponding embedding representations, denoted as EC and ES, respectively. These embeddings capture the semantic meaning of the arguments.

(b) **Attention Mechanism:**

This mechanism helps the model focus on the most relevant information within the retrieved similar cases (Si) based on the specific context of the new argument (C). Imagine the retrieved cases are like similar past experiences, and the new argument is a new situation. The attention mechanism helps identify which parts of those past experiences are most relevant to understanding the new situation.

In simpler terms, the attention mechanism analyzes both the new argument (C) and the retrieved similar cases (Si). It then creates a representation (A) of the retrieved cases, highlighting the parts that are most important in understanding the new argument.

3. **Classifier:**

The classifier takes the adapted representation (A) generated by the adapter as input and predicts the final fallacy class for the new argument (C). It utilizes a fully connected neural network layer with a depth (d) and a specific activation function (e.g., softmax) to convert the internal representation into probabilities for each possible fallacy class. The softmax function ensures the output probabilities sum to 1, allowing the model to interpret the output as class probabilities.

During training, the weights of the retriever component are frozen. This is because the retrieval stage focuses on finding similar cases based on their overall similarity, and its effectiveness is less dependent on continuous weight updates compared to the adapter and classifier which learn the nuanced relationships between features extracted from the arguments and their corresponding fallacy classes. The adapter and classifier, on the other hand, undergo end-to-end training, where their weights are continuously adjusted to minimize the overall classification error.

3 Experimental Setup

This section details the experimental setup used to evaluate the effectiveness of our Case-Based Reasoning (CBR)[4] approach with ELECTRA[5] as the base Language Model (LM) for classifying logical fallacies.

We utilize the LOGIC dataset[6] for our experiments. This dataset consists of arguments annotated with thirteen different logical fallacy types, including common fallacies such as:

1. **Ad Hominem:** attacking the person rather than the argument,
2. **Ad Populum:** appeal to popularity,
3. **Appeal to Emotion** using emotions to persuade rather than evidence,
4. **Circular Resoning:** restating the conclusion as a premise,
5. **Equivocation:** shifting the meaning of a word or phrase,
6. **Fallacy of Credibility:** appealing to an unreliable source,
7. **Fallacy of Extension:** assuming something true in a specific case applies generally,
8. **Fallacy of Logic:** formal errors in reasoning structure,
9. **Fallacy of Relevance:** introducing irrelevant information,
10. **False Causality:** mistaking correlation for causation,
11. **False Dilemma:** presenting only two options when more exist,
12. **Faulty generalization:** drawing broad conclusions from limited evidence,
13. **Intentional:** misrepresenting someone's argument to make it easier to attack.

The LOGIC dataset[6] provides a rich resource for training and evaluating our model's ability to identify various fallacy types. To ensure the robustness of our evaluation, it's important to consider the size and distribution of the data. Ideally, the dataset should contain a sufficient number of examples for each fallacy category to prevent the model from overfitting on specific types of fallacies. Additionally, a balanced distribution of examples across categories is desirable to avoid biasing the model towards more frequent fallacies.

3.1 Case representation:

1. **Counterarguments:**

   Frequently used in persuasive writing to clarify why one's perspective is superior to the counterargument and to head off any uncertainty regarding the argument.

2. **Goals:**

   Studies of argumentation frequently concentrate on the interaction between the writer's arguments and the goals they are trying to achieve. Thus, we anticipate that it is helpful to consider the arguments' purposes when categorizing logical fallacies.

3. **Explanations:**

   We hope to enrich the arguments with a wider concept of information that might be helpful for categorizing logical fallacies but is not already included in the original argument by using explanations about logically fallacious arguments. An example of this would be the steps taken in reasoning an argument's way from its premises to its conclusions.

4. **Structure:**

   Higher-order relation comprehension is required for tasks like logical fallacy categorization, which are frequently focused more on the argument's structure than its content.

3.2 Models and Baselines:

1. **Our Model:**

   We employ a CBR[4] approach built upon the pre-trained ELECTRA[5] language model. The CBR[4] system retrieves similar cases from the dataset based on cosine similarity and utilizes an attention mechanism to adapt these retrieved cases to the specific context of the new argument being classified.

2. **Baseline:**

   For comparison purposes, we establish a baseline using vanilla LMs without the CBR[4] extension. We will also consider SimCSE[7], a simple contrastive learning framework, as an additional baseline. SimCSE[7] is optimized for capturing overall sentence similarity and can be used to assess the effectiveness of our more elaborate CBR[4] approach.

3.3 Hyperparameters and Training Details:

We set the number of heads in the multi-headed attention component of the adapter module to 8. The classifier utilizes a fully connected neural network layer with a depth of 2 and a GELU activation function[8]. Briefly describe any additional training details here, such as the training-validation split, batch size, or optimization algorithm used.

3.4 Evaluation Metrics:

We will evaluate the performance of our model and baselines using standard classification metrics, including accuracy, precision, recall, and F1-score. These metrics will provide a comprehensive assessment of how well each method identifies the correct fallacy categories for unseen arguments.

4 Result:

Our model achieved promising results in detecting various logical fallacies within textual arguments. We observed a positive trend in performance across different stages of the analysis process for the fallacy types we investigated. The performance of our approach on four different fallacy types is given below:

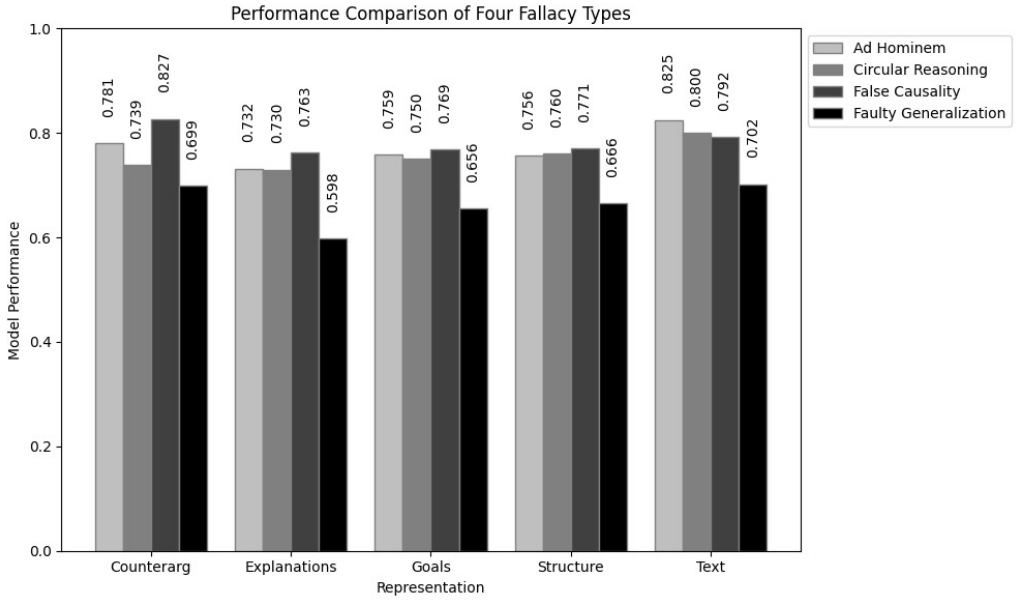

Figure 2: Performance of four fallacy types

We evaluated the performance of our model against baseline models, their F1 scores, Recall and Precision have been visualized below as bar graphs.

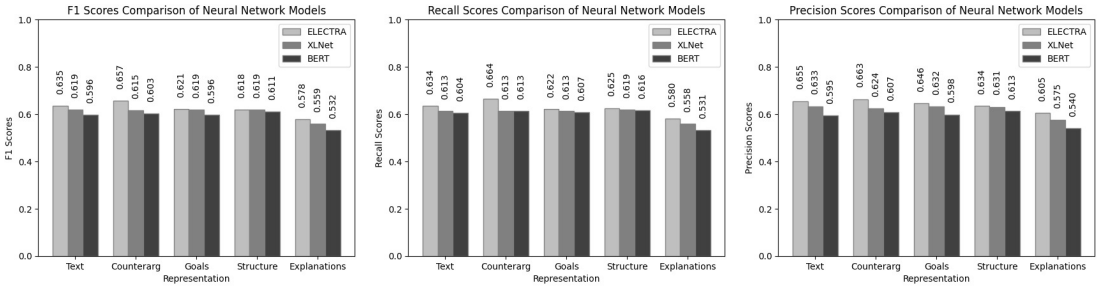

Figure 3: (a) F1 Score (b) Recall (c) Precision

5 Discussion/Conclusion

Electra Transformer consistently performed better. Overall augmenting data with counter argument and Structure gave better performance. Retrieving just a single case instead of many cases worked the best.

6 Team Contributions

Navin led the charge in model preprocessing, wielding a profound understanding to refine data effectively. Aswatth skillfully orchestrated testing and reporting, showcasing a nuanced mastery of methodologies. Anju meticulously fine-tuned and evaluated, revealing a keen comprehension of the intricacies involved. Nikita diligently maintained data readiness and model stability, ensuring a solid foundation for success.

7 Link to code repository

`https://github.com/NavinPrasath14/Logical-Fallacy-Classification`

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[3] H. Allcott, M. Gentzkow, and C. Yu, "Trends in the diffusion of misinformation on social media," *Research and Politics*, vol. 6, p. 205316801984855, 04 2019.

[4] Z. Sourati, F. Ilievski, H. Ân Sandlin, and A. Mermoud, "Case-based reasoning with language models for classification of logical fallacies," 2023.

[5] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," 2020.

[6] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, and B. Schoelkopf, "Logical fallacy detection," in *Findings of the Association for Computational Linguistics: EMNLP 2022* (Y. Goldberg, Z. Kozareva, and Y. Zhang, eds.), (Abu Dhabi, United Arab Emirates), pp. 7180–7198, Association for Computational Linguistics, Dec. 2022.

[7] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, eds.), (Online and Punta Cana, Dominican Republic), pp. 6894–6910, Association for Computational Linguistics, Nov. 2021.

[8] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2023.