Reproducibility Report Instructions for CSE 481N

Griffin Golias, Alex Dundarov, Charles Immendorf, Eric Yeh
Team 4 - Paul G. Allen's Card
{goliagri, alexd02, chazi, yehe}@uw.edu

April 19, 2023

Project Proposal

Paper: Logical Fallacy Detection [1]

Hypothesis

The paper we are trying to replicate runs tests to demonstrate that introducing structure awareness to a model will improve its ability to detect logical fallacies. Additionally, the paper tries to evaluate if the model trained on the main LOGIC dataset can generalize to the LOGICCLIMATE dataset.

Data Access

The dataset we will train and evaluate is a collection of news articles that may or may not contain a variety of logical fallacy types. The links to these articles can be found on the Github repo.

Implementation

We will mostly work off of the already-implemented Github repo provided here: https://github.com/causalNLP/logical-fallacy

Feasability

The codebase does not have too much documentation, but there are guides on how to run the models for training that we could follow. Additionally, we have gotten in contact with the primary author of the paper, who we can refer to for insights along the way. Overall it seems that as long as we can get the models running, it should be quite feasible, and we can figure out ways to extend upon the model later on.

Minimum Viable Action Plan

As for the minimal viable action plan to reproduce the paper's results, these would be the steps that would need to be taken:

1. We'd need to first familiarize ourselves fully with the paper and its corresponding codebase. This would involve closely reading both and working together to understand both of them fully, so we're able to use their code properly. This can also involve trying to use the codebase, so that we're more familiar with the workflow of the codebase, and to ensure that we can execute the codebase's intended behaviors properly (make sure things work on our end, as well as make sure we can access the large language models that they used in their evaluations). We estimate this step to take about 1.5 to 2 weeks.

2. Then, we'd try to reproduce the paper's results. For this paper, this would involve evaluating the saved models on the paper's datasets, using the same metrics they used. The models that were saved were the best-performing ones (the Electra models); the others weren't saved. We would see if our evaluation results lined up with theirs. Specifically, we want to see if the models that they say are the best perform as well as they said. We estimate this step to take about 1.5 to 2 weeks.

3. Then, if we successfully reproduced the paper's results, then we'd perform an additional experiment not in the paper. One possible experiment would be to see how the model performs on valid arguments (does the model falsely detect a fallacy in a valid argument?). We estimate this step to take about 1.5 to 2 weeks, depending on what exactly the experiment is.

Stretch Goals

1. Perform evaluations on the models that weren't saved (the non-Electra ones), to see if the saved Electra models do perform better on them. This would involve obtaining these models and fine-tuning the ones that were fine-tuned.

2. Try to improve the model's performance via changing the best model's hyperparameters.

3. Do the same for GPT-4.

Progress

We have downloaded the codebase and installed the dependencies necessary to run the code. We are currently further exploring the components of the codebase to understand how they work together to move towards retrieving the original evaluation results.

The next steps for us include the following:

1. Get access to the NLP servers and run on there

2. Try to run the script(s) for training/evaluating the struct aware model

3. Document our work and findings so we can continue to reproduce them

4. Meet with the paper authors again to answer questions on the paper and on the codebase and get ideas on stretch goals

1 Introduction

The paper we are trying to replicate runs tests to demonstrate that introducing structure awareness to a model will improve its ability to detect logical fallacies. Additionally, the paper tries to evaluate if the model trained on the main LOGIC dataset can generalize to the LOGICCLIMATE dataset.

2 Scope of Reproducibility

This paper introduces a structure-aware model called Electra-StructAware and claims that this model classifies logical fallacy arguments better than other models. The reasoning is that the structure of an argument is more likely to determine the logical fallacy type than the content words, therefore it would be ideal to mask the content words out to simplify classification. To corroborate the claim, the paper ran experiments using the Electra-StructAware model and compared the results against other zero-shot and fine-tuned models and outperforms the best one, giving an F1 score of 58.77%. Since the other models are no longer available, we will aim to solely reproduce the results of the Electra scores to determine if the claim holds.

3 Methodology

3.1 Datasets

The dataset we will train and evaluate is a collection of news articles that may or may not contain a variety of logical fallacy types. The links to these articles can be found on the Github repo.

Communication with Original Authors

We have contacted the lead author of the paper, Zhijing Jin, and the main contributor to the Github repo, Abhinav Lalwani. We have learned that the reproducibility issue on the Github repo has not yet been resolved, and they encourage us to work towards getting it working properly.

Questions to Ask:

1. Explanation of the results, and why some of the accuracies are 0s?
2. More context on how the model can be trained and evaluated, which scripts to run (including what commands to pass in)
3. What further work can be done to build on the results from the paper?

References

[1] 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*, pages 7180–7198, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. 1