NLP Fall 22 Project Report - *Logical Reasoning*

Raviram Mamidi 122307268
Tejesh Andhavarapu 1225589664
Micah Secrest 1214677828
Akash Kant 1222368576

1 Introduction

This project explores GPT3, OpenAI's latest natural language processing model and BERT, for logical reasoning. GPT3 is a large-scale, deep-learning, natural language text model. This model excels in text production and question answering. This study intends to obtain insights about GPT3's capabilities and how it might be applied to real-world challenges by assessing its performance on logical reasoning tasks. Logical reasoning is crucial for problem-solving, decision-making, and other higher-level thinking skills. Logical reasoning is a vital component of intelligence and key to success in many disciplines. Assessing GPT3's logical thinking is crucial.

We'll develop a series of tasks to evaluate the model's capacity to form inferences, draw conclusions, and solve issues. Each challenge will test the model's reasoning. We'll also test the model's natural language processing and text understanding. We'll compare GPT3 to other language models on these tasks. After evaluating GPT3's logical reasoning ability, we'll train a BERT model for label prediction. Google's BERT is a deep learning model created for NLP. Training a BERT model for label prediction will help us investigate GPT3's capabilities and evaluate if it can reliably predict labels from natural language inputs. By testing GPT3 on logical reasoning tasks and training a BERT model for label prediction, we can determine its potential and how it can be implemented.

2 Methods

2.1 Manual Data Generation

We created 5 samples for each of the logic. Upon cross-referencing and feedback from the professor, we updated and corrected some of them. This had to be stored in a simple JSON, structured with the Type, Subtype, Premise, Hypothesis, and Label fields. The simple followed JSON format is given below.

2.2 Data Creation using GPT-3 (Davinci - 2)

Each group member used GPT-3 to generate more data samples, roughly 10 times the size of manual samples. This exercise was to leverage prompt engineering and GPT3's learning capability. Since it can handle reading comprehension tasks, it will understand the pattern of the texts given to it. This is the prompt we used for generating the data. "Create 10 new reasoning examples similar to the samples below. Use different topics and also generate a few false labels. Do not repeat examples. Hypothesis and Premise cannot be the same." For the most part, GPT3 was very effective at generating additional examples, but occasionally it would repeat data points or create examples that are very similar to existing ones. When this occurred, we introduced a few tactics: Variance Prompting: Adding "Do not repeat the topics" or "Do not repeat the premises" will encourage the model to generate new examples Presence Penalty: Adding a presence penalty discourages GPT3 from generating words that already exist, increasing the variety

3 Experiments - Models Used

3.1 GPT-3 (Davinci 2 and 3)

GPT-3 is an autoregressive generative language model developed by OpenAI, which incorporates only the decoder portion of the transformer model. There are a few versions of GPT-3, of varying speed and power, but in this project, we incorporated Davinci 2 and 3. Davinci is the most powerful line of GPT-3 models and they should be able to handle any task that the others can handle. Davinci 2 was used for data generation, as Davinci 3 was not out yet. We used Davinci 3 for the evaluation of GPT-3 because we wanted to see the best GPT-3 could offer us.

3.2 Bidirectional Encoder Representations from Transformers

- BERT Base Uncased: Bert-base-uncased is a Google-developed pre-trained model that uses a deep bidirectional transformer pre-trained on a huge corpus of lowercase (uncased) text. This model can be utilized to rapidly generate a model for natural languages processing tasks like text classification, question answering, and language comprehension. It is trained on a variety of tasks and datasets to generate a strong and potent general-purpose model.

- BERT Large Uncased: The BERT Large Uncased model is an upgrade of the Bert base uncased model. It is a sophisticated, bidirectional transformer that has been trained on an even larger corpus of uncased text. It is aimed to capture more complicated word-sentence interactions in natural language processing tasks. It is trained on a variety of tasks and datasets, resulting in a model that is more powerful and robust than its predecessor.

4 Experiments

We performed a series of experiments evaluating the performance of GPT-3 and BERT on labeling hypotheses in our dataset as True, False, or Undetermined.

4.1 GPT-3

Because GPT-3 is a generative model, a prompt must be used to coerce it into performing classification. For the purpose of our experiment, we used the following prompt:

Given a set of premises and a hypothesis, label the hypothesis as True, False, or Undefined. Premises: PREMISES Hypothesis: HYPOTHESIS Label:

Above, "PREMISES" and "HYPOTHESIS" is replaced by a given data point's list of premises and its hypothesis respectively. The space after "Label:" is left blank because this is where GPT-3 is expected to fill in its prediction.

For this experiment, we used the entirety of our dataset in order to ensure that the evaluation covered a wide variety of examples. This is opposed to the BERT experiments where much of the data needs to be split into training and testing datasets. In order to automate this testing process, we wrote a python script that iterates through the dataset and uses the OPENAI python library to call GPT-3 using the prompt specified above for each data point. After receiving a response for a given data point, any whitespace is stripped from the output and it is compared to the ground truth label. We anticipated that GPT-3 may generate unexpected labels and that we would need to have an "other" category for these, but GPT-3 stuck to the prompt and only used True, False, and Undetermined.

We also included a couple of parameters in our call to GPT-3. The first of these was a temperature of 0, specifying no randomness. This is good because we don't need any creativity or variety for this task, we just need to receive the best fit. It also should help to make the results reproducible. The other parameter we included was max tokens of 7. While "True", "False", and "Undetermined" are single tokens, it is helpful to ensure that GPT-3 can add some whitespace tokens without ruining the results.

4.2 BERT

For our BERT experiments, we split our data into testing and training datasets. We used the training dataset to finetune a pre-trained BERT for Sequence Classification models from Huggingface. After training each model, we evaluated each model against the training dataset. To establish a baseline for our experiment we trained BERT Base Uncased once with 4 epochs and once with 8 epochs.

4.2.1 Adjusted Attention Heads

We also trained and tested BERT Base Uncased and BERT Large Uncased with adjusted numbers of attention heads. We trained and tested BERT Base Uncased once with 24 attention heads and once with 48 attention heads (as compared to the usual 12). We trained and tested BERT Large Uncased once with 24 attention heads.

We then compared the results of our baseline and adjusted models to evaluate their performance.

5 Results and Analysis

5.1 GPT3

The results from the GPT-3 experiment can be seen in Figure 1.

From our results, we can see GPT3 is relatively good at classifying True hypotheses. A precision

of 88.661% means that a value classified as True has an 88.661% of actually being True, which could prove useful in many scenarios. Its F1 score, however, is a bit lower at 79.64%

When it comes to False hypotheses, GPT3 is not nearly as reliable. A precision of 56.127% means that a value classified as True has only a 56.127% of actually being False, which is not very useful. While its recall is a bit higher at 77.278% its F1 score is still quite low at 65.03%.

Our results show that GPT3 is very bad with Undetermined hypotheses. For the Undetermined hypotheses, precision, recall, and F1 score are all less than 20%.

Overall, GPT3 is situationally useful for this task but is not very reliable. If you only care about labeling True samples as True and a 12% error rate is acceptable GPT3 could be used for this task. Performance with False and Undetermined hypotheses, however, is not reliable enough to be useful

The difficulty in predicting Undetermined may be because the word "Undetermined" is much less common than "True" or "False" in GPT-3's training dataset. It is likely that there is some number of True or False questions in GPT-3's training dataset. These are very unlikely to contain Undetermined as an answer. Between True or False questions and Undetermined generally being less used, GPT3 may have a less complete "understanding" of its meaning

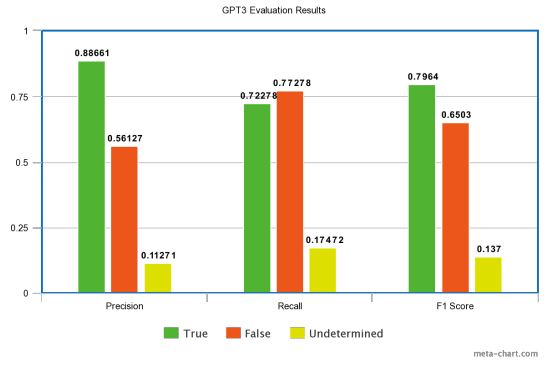

Figure 1: GPT3

5.2 BERT

The BERT base model has shown good results, with an overall accuracy of 76%, with good precision for the True and False hypotheses (77.04% and 71.81%, respectively). However, the precision for Undetermined hypotheses was 100%, but the recall was only 3.17%, indicating that the model was only able to return a few correct results. The F1 scores for True and False hypotheses were 84.86% and 62.29%, respectively, while the F1 score for Undetermined was 6.15

By increasing the number of epochs to 8, the BERT model was able to improve its performance. However, increasing the number of attention heads to 24 or 48 reduced the overall accuracy. The BERT large model was able to achieve an overall accuracy of 80.16%, which is an improvement over the base model. It had good precision rates for True (84.32%), False (68.38%), and Undetermined (71.64%) hypotheses. The F1 scores for True, False, and Undetermined hypotheses were 87.58%, 61.38%, and 67.60%, respectively.

The results for each BERT model can be seen in the graphs below.

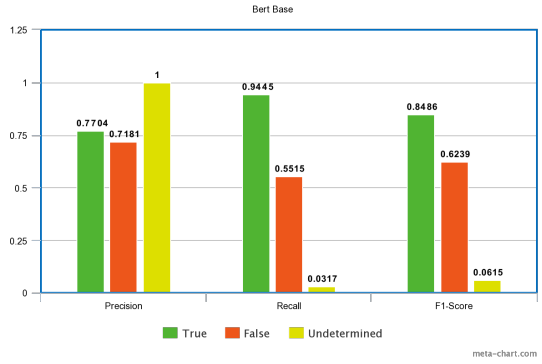

Figure 2: BERT Base

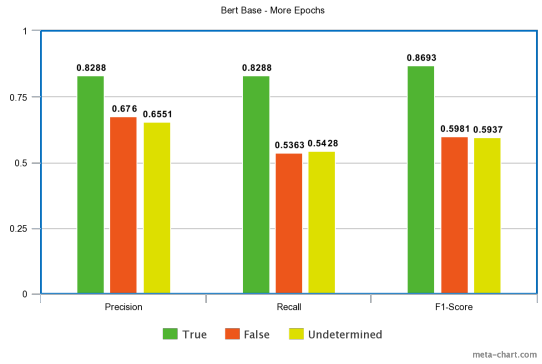

Figure 3: BERT Base with more Epochs

6 Individual Contributions

6.1 Akash Kant

My primary contribution to this project was creating the manual and gpt3 created 10 times large dataset. As in logical reasoning, there are not a lot

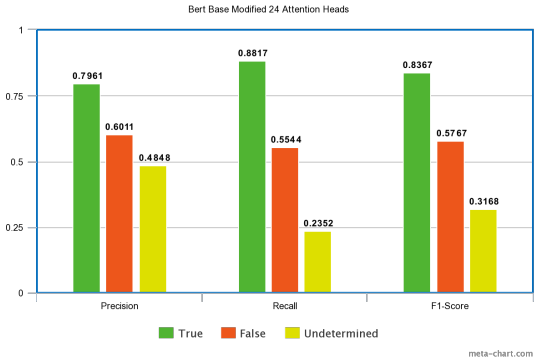

Figure 4: BERT Base with 24 attention heads

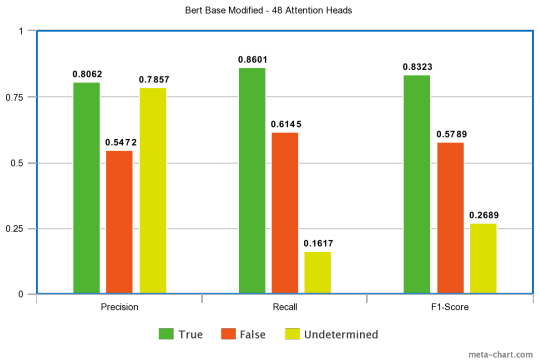

Figure 5: BERT Base with 48 attention heads

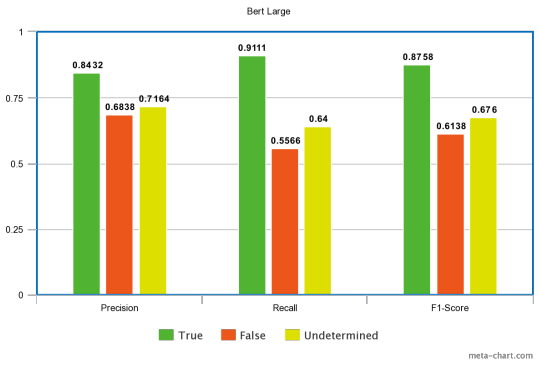

Figure 6: BERT Large

of collaborated datasets. With 12 people collaborating over multiple weeks we made a large sample of the dataset. Created the manual and GPT 3 for the data creation. Ran the BERT base model and helped in plotting graphs for the results. Aided in building the report in Latex.

6.2 Micah Secrest

My primary contribution to this project was the evaluation of GPT-3. I wrote the python script which retrieves and evaluates the results from GPT-3 and ran it, ensuring that there were no issues in the final run. During this process, I ran into GPT-3's rate limit, which I had to first circumvent by limiting my script's query rate, but eventually, I caved and made a paid account to bypass this to speed up the testing process. Additionally, this process led to me finding a handful of data quality issues that we fixed as part of preprocessing, or that I worked around in my script.

In addition to the evaluation of GPT-3, I contributed to the data generation process like the rest of the team members, and I contributed to the discussions surrounding BERT's training and evaluation process. Finally, I contributed to each of the required reports and presentations for the team.

6.3 Raviram Mamidi

My contributions include exploring hyperparameters like temperature and frequency penalty of GPT3 for ten-time data generation, using these two parameters we were able to save the time of rerunning gpt3 for sample generation and reprompting. I helped in running the Bert-Base model and aid in building the code surrounding data preprocessing and result in inference. I was also able to incorporate the attention head tweaking mode of Bert that would help the team to build a variant model that has helped in creating new data points for comparison with other models in the report.

6.4 Tejesh Andhavarapu

My primary contribution to the project was creating the manual dataset and using GPT3 to make it 10X. Then I went over all the GPT3 generated data to validate the data and make some minor corrections to the generated data. I ran the BERT base model, BERT base model with 24 heads, and BERT large model and then summarized the results. In addition, I contributed to the final report and the presentation slides.

7 Conclusions

Overall, this project has served as a comparison between a handful of approaches to logical reasoning in natural language processing. BERT models tend to outperform GPT-3, but their downside is that they require training. GPT-3 on the other hand performs surprisingly well out of the box on these logical reasoning examples, at least on True classifications. For this task, increasing the attention heads didn't seem to have a major impact, and 12 attention heads seems to be enough. In total, the BERT Large model was the best performer due to its increased size. Comparisons between methods like this are important for deciding which type of model is ideal for a given task.

References

GPT3

BERT

GitHub - https://github.com/akashkthkr/CSE-576-Fall22-NLP-Dev-engers

Model and Jupyter Files GDrive - https://drive.google.com/drive/folders/1ijuOLvCjvsEc9ErPguMlExVIDWj4aN25?usp=sharing

DataSet - On GitHub and also managed on Test.json and Train.json on the dataset Folder.

Models - https://drive.google.com/drive/folders/13JUzIdJgDZR6fx9JaNhqKkhdOxdexENUusp=share_link