Jack McMorrow
DATS 6312 - Natural Language Processing for Data Science
12/11/2023

Individual Report

Introduction

Our project aimed to tackle the challenging task of using natural language processing and transformers to translate Math Word Problems (MWP) used in grade schools into their corresponding mathematical equations and answers. Given the wide range of ambiguity in these types of problems, trying to translate mathematical linguistics into actual numbers would be difficult to achieve in a few weeks.

In order to succeed, we made sure to ensure many aspects of our project were selected with care and consideration. This included sufficient data and transformer architecture, as well as different ways to evaluate the performance of the model. By fine-tuning a pretrained model, Flan-T5 from Hugging Face, we were able to produce a model that translated the words to numerical equations. While these answers were not always accurate, we were still able to demonstrate the power of transformers when it comes to understanding mathematical problems that are written in natural language.

In our project we used two primary datasets for training and testing of our model, MAWPS (A Math Word Problem Repository) and SVAMP (Simple Variation on Arithmetic Math Word Problems). Previous research papers had documented shortcomings of existing datasets and recommended these for a project like this to encapsulate the wide variety of language used in MWPs. (Patel et al., 2021). Once we found sufficient data, we were ready to begin our work.

Description of Individual Work

The project began by each of us coming up with different ideas for our project and what we wanted to achieve. We discussed our different ideas and ended up selecting Akshay's idea of translating mathematical word problems into their corresponding equations. We took some time

to familiarize ourselves with the data and do some research, as we had not had the lectures for transformers yet. After those lectures, we felt more familiar with how to approach this problem.

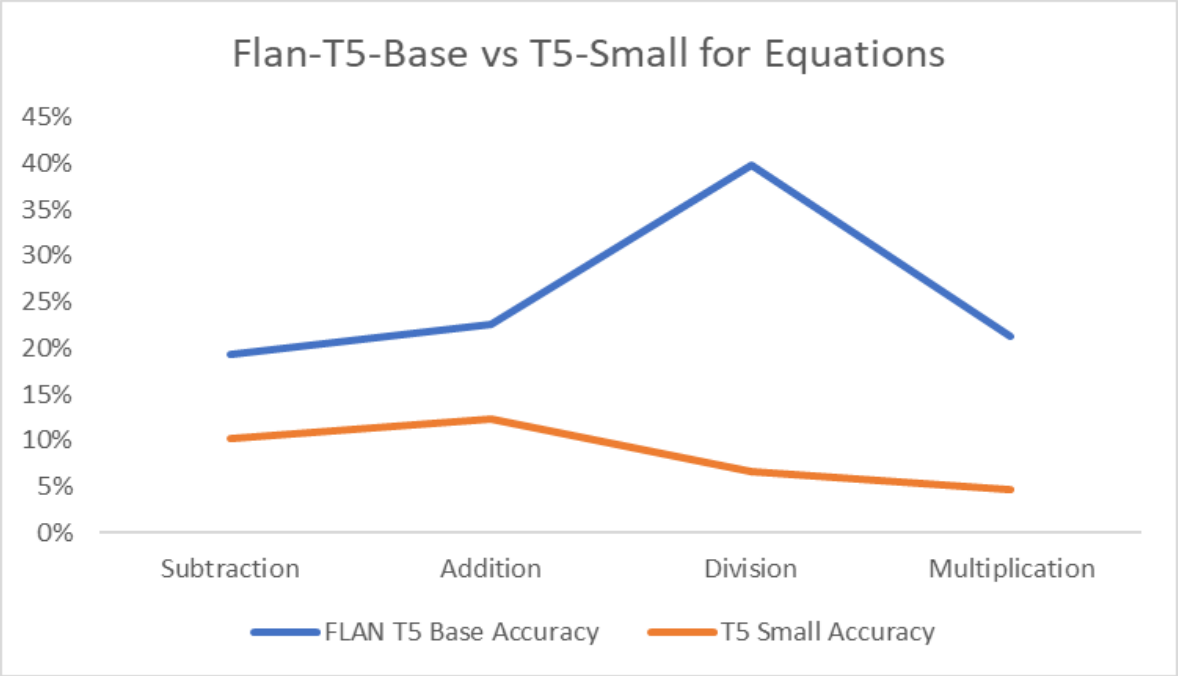
Next, we developed a few different transformer models and compared their performance before making a final selection. We found that we did not have a lot of data to benign with, so that required us to search for some more, which was found. Once we had a model that performed well we focused on fine tuning it and getting metrics for its performance. Our frequent meetings and communication via text definitely facilitated our ability to complete the project efficiently.

We spent much of the last work working on the streamlit user interface and written report. I contributed most to the development of the UI to ensure a good presentation, as well as some final research into how the T5 model works and how to approach different metric evaluations for our report and presentation. As a whole, our team excelled at communicating our results which helped us all understand the status of our project which allowed us all to excel in our individual endeavors.

Results

Our work resulted in two different models that translated the math problems into their equations. The Flan-T5 model performed better with an overall accuracy of 23.6%. When evaluated at an operation level, the model correctly answers 19.4% (103 out of 531) of Subtraction questions, 22.6% (44 out of 195) of Addition questions, 39.8% (66 out of 166) of Common-Division questions, and 21.3% (23 out of 108) of Multiplication questions. The T5-small model, on the other hand, had a lower performance when compared to the Flan-T5-base model, which had a lower accuracy at 9.4%. It also underperformed on all of the operations, which is shown in the figure below.

Figure 1. Accuracy Across Different Models by Operation



In addition to accuracy, we incorporate ROUGE scores as supplementary metrics for checking the quality of generated equations. ROUGE scores measure the overlap and similarity between the model generated and reference equations by comparing the number of matching n-grams. Here we report ROUGE-1, which analyzes matching unigrams, ROUGE-2, which analyzes matching bigrams. ROUGE-L, on the other hand, matches the Longest Common Subsequence, which is the longest subsequence that occurs in both the generated output and the reference equation. The ROUGE scores for the Flan-T5 model had the following metrics: ROUGE-1 (0.605), ROUGE-2 (0.287), and ROUGE-L (0.605) The rouge score was also much lower for the T5-small model, with ROUGE-1 at 0.573, ROUGE-2 at 0.144, and ROGUE-L at 0.673.

Figure 2. Rouge Scores for Flan-T5-Base model.

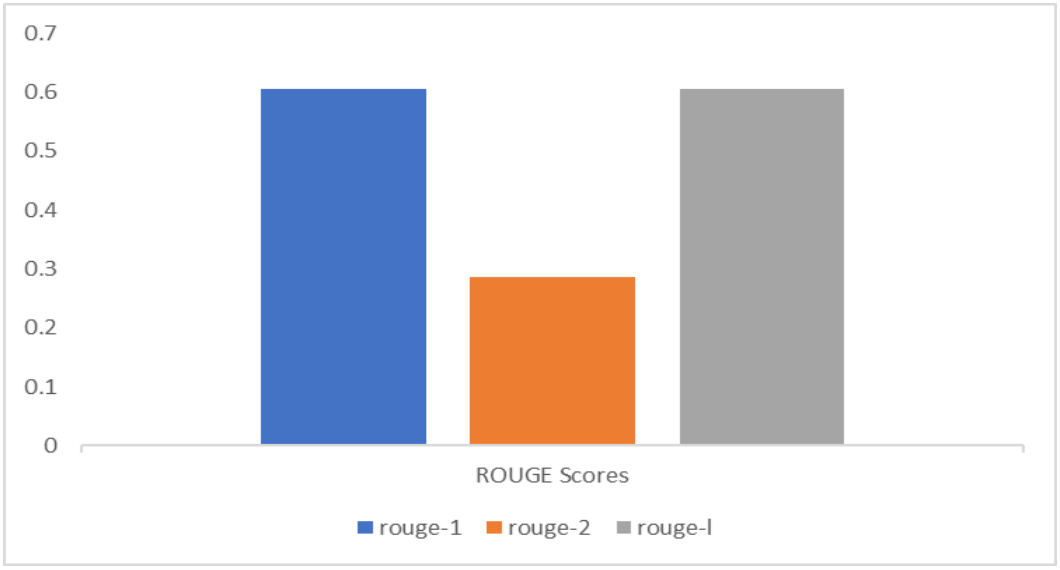
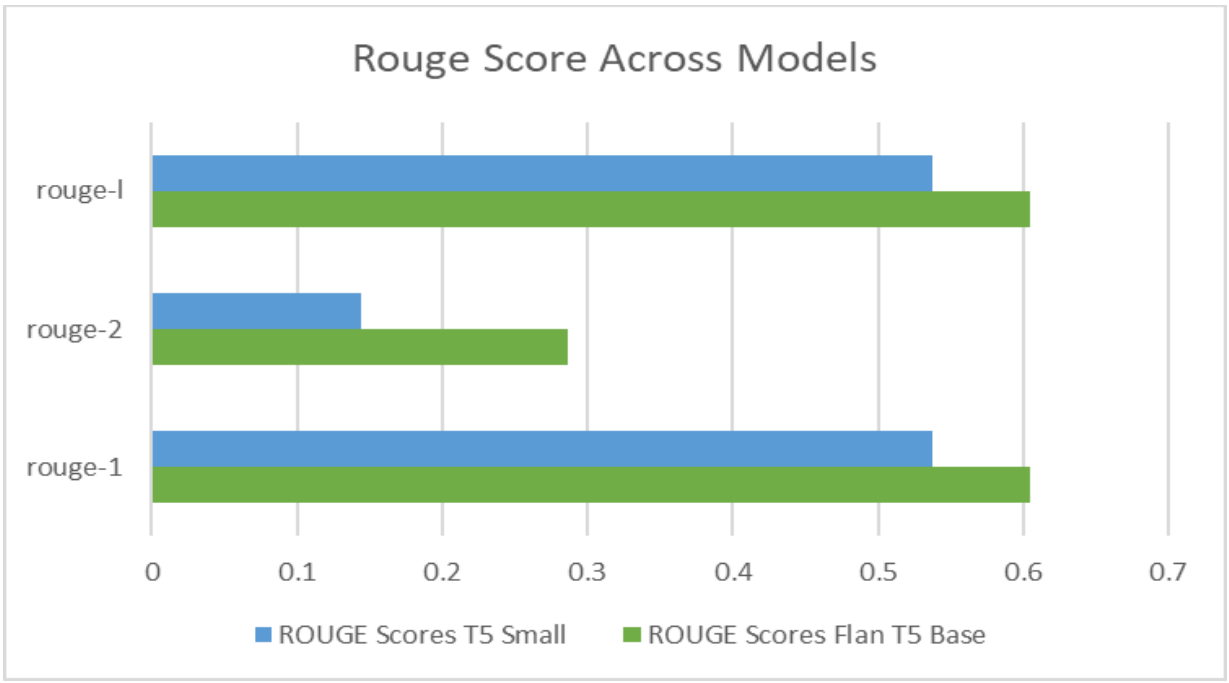


Figure 3. Rouge Scores for Flan-T5-Base and T5-Small



Summary and Conclusions

The results from our models are far from perfect, but they do demonstrate the capabilities of transformer models when it comes to understanding mathematical linguistics. With more fine tuning and advancements in transformer technology, the results from an experiment like this could significantly improve. Overall, the results of this project emphasize the robustness and power of transformers and LLMs in their ability to translate word constructed math problems into their underlying equations. After finding sufficient data and experimenting with different models, we were able to utilize two different T5 models in order to achieve a desired result.The Flan T5 base model performed better across the different ROUGE score, which measure the similarity between the desired target and the output of the transformer, as well as the overall accuracy of the results, which stood at 23.6%. The T5 small model had consistent accuracy across different operations, while the Flan T5 model performed the best on division operations.

Despite the satisfactory results, an accuracy of just twenty-four percent can definitely be improved upon. While the Flan T5 model has impressive architecture and has been proved to perform well on many different tasks, the final model we trained tends to fall short for more complex mathematical problems. The model performance begins to plateau, signifying an underlying limitation in its performance. In order for the model to improve, the capabilities of the model would need to be enhanced for our accuracy to increase. As mentioned during our presentation, changing the learning rate of the T5 model could have improved the accuracy of its final performance. We could also explore other types of transformers and assess their performance for a problem like this. Regardless, the Flan T5 model's ability to assess these mathematical problems was impressive and shows the powers that Natural Language Processing has when it comes to translating MWPs into equations.

Percentage of Code

About 30% of my final code was taken from online sources or improved upon from previous projects. Note: I had trouble configuring my git settings so some of my commits may have been made under "Ubunutu" rather than "jmcmorrow".

References

Doe, P. (2021). Rouge Your NLP Results. Medium. *https://medium.com/@priyankads/rouge-your-nlp-results-b2feba61053a*

Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., & Hajishirzi, H. (2016, June). MAWPS: A math word problem repository. In Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies (pp. 1152-1157).

Hugging Face. (2023). T5-small: Text-To-Text Transfer Transformer. Hugging Face. *https://huggingface.co/t5-small*

Hugging Face. (2023). T5-Flan-Base: Text-To-Text Transfer Transformer. Hugging Face. *https://huggingface.co/google/flan-t5-base*

Jacob, John. (2023). What is FLAN-T5? Exemplary AI Blog. *https://exemplary.ai/blog/flan-t5*

Patel, A. (2022). SVAMP: Structural Variant Annotation, Mapping, and Prediction. GitHub. *https://github.com/arkilpatel/SVAMP*

Patel, A., Bhattamishra, S., & Goyal, N. (2021). Are NLP models really able to solve simple math word problems?. arXiv preprint arXiv:2103.07191.

MathBot. (2023). MAWPS_Augmented.pkl. GitHub. *https://github.com/Starscream-11813/MathBot/blob/main/MAWPS_Augmented.pkl*

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21(140), 1-67.