TriviaAI

Alyssa Yang
University of Michigan
Ann Arbor, US
awyang@umich.edu

Abstract—Trivia has always been a go-to activity for people whether it be recreationally or by watching game shows like Jeopardy! or Who Wants to be a Millionaire?. In either case, it offers a competitive and engaging way to test knowledge and learn new facts about a variety of topics. This project explores the fine-tuning of a T5-based transformer model to improve performance on a trivia question-answering task using an open trivia database. Overall, this work underscores the complexity of adapting pre-trained language models to specific tasks and emphasizes the importance of data quality and augmentation strategies in achieving performance improvements.

I. Introduction

In recent years, increased interest in natural language processing (NLP) has led to advancements in the field of machine learning by allowing computers to understand and generate human-like texts and structures. Its numerous applications have allowed experimentation not just within the traditional tasks such as sentiment analysis or translation, but for interactive entertainment as well.

The motivation for this project stems from the desire to explore how pre-trained NLP models can be utilized to create applications within the entertainment field, specifically for trivia. It allows us to showcase how these advancements in NLP models can be applied beyond traditional use cases, such as with question-answering systems, and highlight the potential for personalization in AI-driven games, providing another avenue for more human-computer interactions.

One of the most significant breakthroughs in the field of natural language processing has been the creation of transformer-based models. By introducing self-attention mechanisms, they overcome the limitations of the existing recurrent and convolutional neural networks. These mechanisms allow for the processing of tokens simultaneously which, combined with the usage of modern hardwares such as GPUs and TPUs, have massive computational and speed advantages through this use of parallel processing. Along with this, the creation of these models has allowed transfer learning to gain more traction and practical implementations as well. By allowing for pre-trained models trained on massive amounts of data to be fine-tuned on smaller and more task-specific datasets, it reduces computational requirements, and thus resources needed, and improves performance.

Through these developments in natural language processing, conversational AI and chatbots have also grown increasingly popular. More specifically, ChatGPT (in conjunction with GPT3) has allowed users to communicate in real time with a bot by understanding and generating natural language autonomously. By using supervised and reinforcement learning, the algorithm continuously updates and gains information through more interactions with users, and has become a forefront of how AI and NLP have shaped how people learn and interact, and how its performance will continue to improve over time.

The goal of this project is to leverage these NLP and machine learning advancements, building upon a pre-trained language model in order to improve performance and build real-world applications. Using them in a dynamic setting will provide an opportunity to not only experiment with these models but also bridge the gap between theory and practice.

II. Method

The pre-trained model I will be using is a T5 model that can answer questions without any context, and it is a transformer-based language model with around 220 million parameters [1]. The main goal of this model is to be able to implement a solution to the question-answering problem, more specifically about closed-book question answering. This type of question loads information about the context into the parameters of the language model and then queries it with no context; this trivia model was specifically trained on a dataset with no contexts to its questions. Overall, the T5 model makes it easier to fine-tune transformers for NLP problems, and it is able to implicitly store knowledge during training, making it a good choice for this type of trivia problem.

The trivia model has been trained on around 95,000 questions and answers from the TriviaQA reading comprehension dataset with no context. It was trained for 80 epochs and obtained an EM score of 17 and a subset match score of 24—the model outputted the exact correct answer 17% of the time, and outputted a subset of the correct answer 24% of the time. This suggests that even after extensive training, the model still does not have the highest accuracy, and this highlights just how difficult and extensive natural language processing tasks can be.

In order to build upon this model, I first imported it from HuggingFace and also imported the new data from the open trivia database [2]. In order to reduce complexity and the computational limitations that come with this task, I opted to only include questions under the categories "entertainment", "food and drink", and "science and nature". By training and testing on only these three categories, it will provide a more

accurate estimate for how well the model would perform on all types of questions if I had a more robust computing system.

I filtered the data to only include the questions, answers, and categories, and cleaned the data by converting answers from their original list format and removing all fill-in-the-blank style questions as the original T5 model was not trained on these types of questions. The size of this dataset was quite small after restricting the categories and cleaning the questions, so I augmented the data by adding two more rephrased questions for each original one - thus, there would be 3 versions of each question to improve generalization ability. In total, I had 2,262 questions in the Entertainment category, 4,989 in the Science and Nature category, and 2,865 in the Food category.

To conduct training, I split up the dataset using a stratified split in order to keep proportions of categories the same throughout each set. I used a 80-10-10 training-validation-testing split in order to train on as many samples as possible and improve performance. After this, I preprocessed the data by tokenizing the inputs and making the answers the target variable to learn. Then, I trained the model using a learning rate of 2e-5, a batch size of 32, and a weight decay of 0.01 over 4 epochs. I chose these numbers for the training parameters through trial and error and evaluated them using resulting accuracies on the test set.

III. RESULTS

For evaluation, I generated predicted answers for the trivia questions in the test set for both the original model and my model. Due to the nuances in how questions can be answered, I evaluated accuracy by calculating the amount of overlap of main words within the answers. To do this, I removed all punctuation, put all words in lowercase, split the answers into separate words, and then excluded all English stopwords using the NLTK library. From there, if at least half of the main words in the answers matched, the predicted answer would qualify as correct.

Due to the randomization within the stratified splitting, the resulting accuracies will not always be the same, but in my run-through, the original model without any additional training achieved an accuracy of 10.67%. Interestingly, after training, the model achieved the exact same accuracy of 10.67%. Thus, after training, my model neither improved nor performed worse compared to the original model.

(a) Accuracy of the original model
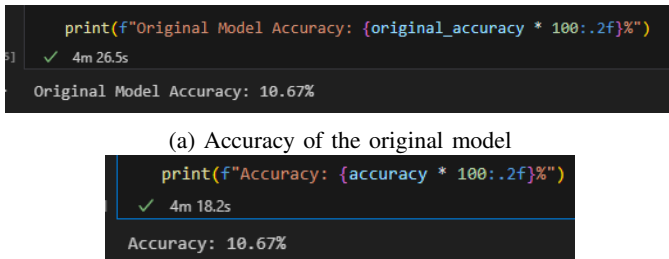

(b) Accuracy of the final model

Fig. 1: Comparison of model accuracies.

Because of this, I also looked at other evaluation metrics including BLEU, F1, and ROUGE scores as shown in Table I below. Except for the BLEU scores, the original model and my final model had very similar scores, meaning that the recalls and overlap between both models' predictions with the actual answer were almost the same, with the original model only slightly outperforming my model. However, my model's BLEU score was over twice as high as the original model's, indicating that after additional training, my model captured many more of the n-grams in the answer than the original. Thus, doing additional training did not impact the accuracy or recall of the model, but made it more precise.

	BLEU	F1	Rouge1	Rouge2	RougeL
Original	0.228	0.089	0.108	0.020	0.108
Final	0.095	0.091	0.114	0.026	0.114

TABLE I: Other evaluation metrics

I also wanted to incorporate accuracy in an interactive way by creating a rudimentary trivia game where the user can compete against my trained model. The user is able to select one of the 3 trained categories (entertainment, science and nature, and food and drink) in which the model will randomly select a question from the open trivia dataset. It will use the model to generate a prediction, take in the user's answer, and compare both with the correct answer listed in the dataset. I used the same accuracy method as described above as a scoring method, keeping track of how many points the user and model have. It allows the user to keep playing as long as they want, and when they are finished, it will print out the final scores along with who won. For example output for how the game is played, refer to Figure 2 in the Appendix.

IV. CONCLUSION

The field of natural language processing continues to advance at a rapid pace, providing breakthroughs in how machines understand and generate human language. However, as evidenced by my exploration with the trivia model, it remains a computationally intensive and complex task. Even with a small subset of data, training the model took approximately 40 minutes to complete, and even then, it's performance was not higher than that of the pre-trained baseline.

From this, we can conclude that the utilization of pre-trained models can be greatly beneficial for large datasets or computationally intensive tasks such as NLP. However, when fine-tuning these models, still a large dataset is required to achieve competitive accuracy as shown by my results above. While pre-trained models offer a powerful starting point, achieving substantial improvements often requires precise optimization, access to larger datasets, and advanced computational resources. Future work could focus on incorporating data augmentation, exploring transfer learning techniques, or using larger datasets to enhance the model's performance in domain-specific tasks such as trivia generation.

REFERENCES

[1] P. Dwivedi, Closed Book Trivia-QA T5 base
https://huggingface.co/deep-learning-analytics/triviaqa-t5-base

[2] C. Sutton and M. Tancoigne, Open trivia database
https://github.com/el-cms/Open-trivia-database

APPENDIX


```
# Play the trivia game
trivia_game(trivia_data, model, tokenizer)
✓ 1m 11.6s

Welcome to the Trivia Game! It's you vs the AI model!

Available categories:
1. entertainment
2. science
3. food

Question: What is the name of the whale that swallowed Pinocchio.

Your Answer: monstro
Model's Answer: Agusta
Correct Answer: Monstro

You got it right!
The model got it wrong.

Current Scores:
You: 1
Model: 0


Available categories:
1. entertainment
2. science
3. food

Question: What Element Is Used In The Process Of Galvanisation

Your Answer: zinc

You: 2
Model: 0


Available categories:
1. entertainment
2. science
3. food

Question: Which country would you associate with the dish Couscous?

Your Answer: morocco
Model's Answer: Tunisia
Correct Answer: Tunisia

You got it wrong.
The model got it right!

Current Scores:
You: 2
Model: 1


Game Over! Final Scores:
You: 2
Model: 1
Congratulations! You beat the AI!
```

Fig. 2: Example trivia game

Link to the GitHub repository:
https://github.com/alyssawyang/stats507-final-project