Interim Report
Master Blasters – Group 12

Adish Golechha – ag2384
Sarvesh Kharche – sk2907
Sneh Bhandari – sb2499

Introduction

Problem
The project addresses the challenge of generating flashcards from educational documents, specifically focusing on science materials, to help students test their understanding of topics covered in these documents.

Solution
We are using a T5 Transformer model developed by Google which contains around 770 million parameters, fine-tuned to generate question-answer pairs from various educational documents, such as presentations or PDFs. By transforming the content into question-answer pairs, students can create flashcards for self-assessment and better retention of the material.

Target Audience
This tool is especially useful for students who want to test their understanding of concepts directly from their study materials.

Progress Overview

I. Model Selection

We use two pre-trained T5 models to generate q-a pairs:

1. **T5-large-generation-squad-QuestionAnswer** – This model has been trained on SQUAD dataset which has extractive data. **S**tanford **Qu**estion **A**nswering **D**ataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable.
2. **T5-large-generation-race-QuestionAnsw**er – This model has been trained on RACE dataset which has abstractive data. Race is a large-scale reading comprehension dataset with more than 28,000 passages and nearly 100,000 questions. The dataset is collected from English examinations in China, which are designed for middle school and high school students. The dataset can be served as the training and test sets for machine comprehension.

While these models could generate question-answer pairs, their answers were not always accurate or relevant when applied to the documents. Therefore, we decided to fine-tune the models with science-specific dataset to improve the quality of generated content.

II. Fine-Tuning the Model

Fine-tuning the model requires retraining 770 million parameters. This takes too much time and computation power. To deal with it more efficiently, we train only a subset of the parameters. We use Parameter Efficient Fine Tuning for efficient memory usage. We used the technique called LORA, which reduced the number of parameters to train from 770 million to 4 million (0.6% of total parameters).

Low-Rank Adaptation (LoRA) is a technique that efficiently fine-tunes large pre-trained models by modifying only a small subset of parameters. Instead of updating the entire weight matrix in transformer models, LoRA decomposes it into two smaller, low-rank matrices. These matrices are trained during fine-tuning, while the original model weights remain frozen. This reduces computational and memory costs, making it more efficient than traditional fine-tuning. LoRA allows large models to be adapted to specific tasks with fewer parameters, enabling faster fine-tuning. It is particularly useful for tasks like question answering, text generation, and sentiment analysis, offering a scalable solution for leveraging large models with limited resources.

Steps involved in fine-tuning:

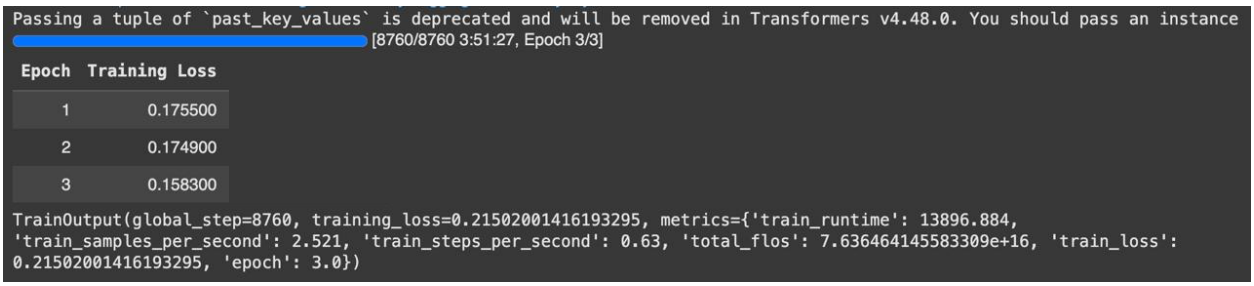
1. **Dataset Selection**: We chose the **SciQ** dataset, which contains question-answer pairs related to scientific content. This dataset serves as the training data for fine-tuning the model. The data includes context (supporting information), questions, and correct answers. The SciQ dataset contains 13,679 crowdsourced science exam questions about Physics, Chemistry and Biology, among others. The questions are in multiple-choice format with 4 answer options each. For the majority of the questions, an additional paragraph with supporting evidence for the correct answer is provided.

2. **Model Fine-Tuning**: We fine-tuned the pre-trained T5 model using the SciQ dataset. We applied LoRA to the attention layers of the T5 model to efficiently adapt it to the task of question-answer generation. This method ensures minimal disruption to the original model while enhancing its ability to generate high-quality question-answer pairs.

3. **Training Configuration**: The fine-tuning was conducted with a set of carefully defined training parameters, including a learning rate, batch size, and the number of epochs. The model was trained on both the training and validation sets of the SciQ dataset to ensure it can generalize well to new unseen content.

Progress

The following is the code for our first run for SQUAD dataset.



```
Passing a tuple of `past_key_values` is deprecated and will be removed in Transformers v4.48.0. You should pass an instance
[8760/8760 3:51:27, Epoch 3/3]
```

Epoch	Training Loss
1	0.175500
2	0.174900
3	0.158300

```
TrainOutput(global_step=8760, training_loss=0.21502001416193295, metrics={'train_runtime': 13896.884,
'train_samples_per_second': 2.521, 'train_steps_per_second': 0.63, 'total_flos': 7.636464145583309e+16, 'train_loss':
0.21502001416193295, 'epoch': 3.0})
```

The following are the hyperparameters we used for the above

```python
training_args = TrainingArguments(
    output_dir="./t5_finetuned_sciq",
    evaluation_strategy="epoch",
    learning_rate=3e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    save_strategy="epoch",
    weight_decay=0.01,
    logging_dir="./logs",
    logging_steps=100,
    report_to="none"
)
```

```python
lora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    target_modules=["q", "v"],
    lora_dropout=0.1,
    bias="none"
)
```

Evaluation Metrics:

BLEU (Bilingual Evaluation Understudy) is a metric used to evaluate the quality of machine-generated text by comparing it to reference texts, primarily focusing on precision by matching n-grams between the generated and reference sentences. It helps assess the fluency and accuracy of questions generated by the model.

BERTScore evaluates semantic relevance by using contextual embeddings from BERT-based models to compute similarity between generated and reference text. It captures more nuanced semantic matches compared to BLEU.

These metrics were chosen for evaluating our T5 model because BLEU ensures that the generated questions are linguistically similar to reference questions, while BERTScore

ensures that the generated answers are semantically aligned with the context, ensuring both quality and relevance in the question-answer pairs.

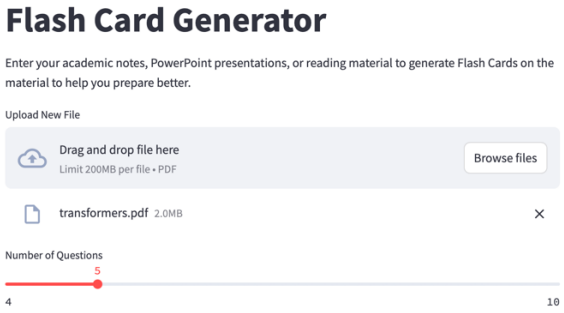
Base model
We will be using a logistic regression model as a baseline model for model comparison. Logistic regression is simple, interpretable, and computationally efficient, making it a strong baseline model for classification tasks. Comparing complex models like T5 against it helps evaluate the performance gains from added complexity.

Web Application Development

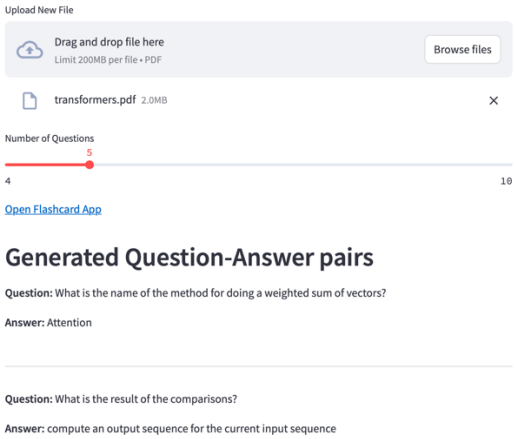
To make the fine-tuned model accessible to users, we developed a web application using Streamlit to create the webapp and Dash for creating the GUI. The application allows users to:

1. **Upload PDF Files:** Users can upload PDF documents (less than 200 MB) for processing.
2. **Generate Question-Answer Pairs:** The uploaded document is processed by the fine-tuned model to generate a set of relevant question-answer pairs.
3. **Create Flashcards:** Users can then create flashcards from the generated questions to test their understanding of the content.

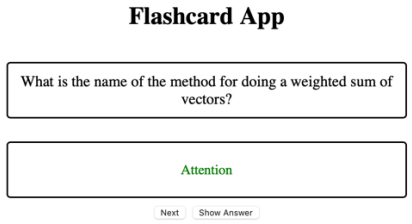
This is how our web-application looks currently. When a file is uploaded, a slider appears for the number of question-answer pairs the user wants to generate.



The following is the generated question-answer pairs when the Transformers.pdf used in this course was uploaded.



This is how the flashcards are seen. The questions are randomly chosen. The answers are only displayed when the **show answer** button is created.



Limitations

1. **Input Size Limit:** The model has a limitation on the length of text – 512 tokens that can be processed at once due to the input size constraints of the T5 model.
2. **File Size:** Users can currently upload PDFs that are up to 200 MB. Larger files are not supported at this stage.
3. **Computational Cost:** Even though using LORA reduced the number of parameters to train from 770 Million to 4 Million, it still takes around 4 hours to fine tune the models on the SciQ dataset which has 11 Thousand datapoints.

Future Plans

We plan to enhance the user interface (UI) to make it more intuitive and user-friendly by integrating features such as drag-and-drop file upload, and progress indicators. We also aim to incorporate **Named Entity Recognition (NER)** and **Part-of-Speech (POS)** tagging to improve the model's ability to generate contextually accurate and grammatically correct question-answer pairs. Furthermore, we plan to expand the system to support a wider range of input formats beyond PDFs and increase file size limits to handle larger documents. Lastly, we are exploring integration with other platforms for broader accessibility and providing real-time feedback on the quality of generated content.