Automated Flashcard Generation Using LoRA-Enhanced Transformer Models

16:954:577:01 Statistical Software
Final Project Report

Team 12:

- Adish Golechha
- Sarvesh Kharche
- Sneh Bhandari

Introduction

Students often struggle to retain key concepts from extensive academic notes. Traditional methods of creating flashcards are time-consuming and may not cover all critical areas, leading to inefficient study practices. The goal of this project was to develop an automated tool for generating high-quality, contextually accurate flashcards from academic notes. This tool aims to help students efficiently test and reinforce their understanding of key concepts, making study sessions more effective and targeted.

This was a **text-generation task**, not a classification problem. To solve this, we started with a baseline model using Recurrent Neural Networks (RNNs). For the primary architecture, we implemented a fine-tuned **T5 (Text-to-Text Transfer Transformer)** model, further enhanced with **Low-Rank Adaptation (LoRA)** for fine-tuning efficiency. We also experimented with integrating **Part-of-Speech (POS) tagging** to improve accuracy and contextual understanding.

The fine-tuned T5 model significantly outperformed the RNN baseline, generating flashcards that demonstrated superior contextual understanding and accuracy. Although the incorporation of POS tagging provided additional insights, it did not lead to substantial improvements in model performance. Overall, the enhanced T5 model achieved the objective of producing high-quality flashcards effectively, marking a notable step forward in automating educational tools achieving high scores for both BertScore and ROUGE-L. Both scores give a complete overlook of the text output quality.

Related Work

We reviewed the paper *"Generating Diverse and Consistent QA Pairs from Contexts with Information-Maximizing Hierarchical Conditional VAEs"* by Lee et al. (2020). The authors tackle the challenge of generating question-answer (QA) pairs from unstructured texts, which is crucial for tasks such as question generation and training QA systems in resource-scarce environments. They introduce a novel framework called Information Maximizing Hierarchical Conditional Variational AutoEncoder (Info-HCVAE), which generates diverse and semantically consistent QA pairs by maximizing mutual information between the questions and answers.

The key contributions of the paper include:

- A hierarchical conditional VAE framework with separate latent spaces for questions and answers, improving diversity by focusing on different parts of a given context.
- An InfoMax regularizer to ensure consistency between QA pairs, making sure the generated questions are answerable from both the provided answers and context.
- Introduction of the Reverse QA-based Evaluation (R-QAE) metric, designed to assess the novelty and diversity of the generated QA pairs.

Experimental results on benchmark datasets such as SQuAD, Natural Questions, and TriviaQA showed that Info-HCVAE outperformed state-of-the-art baselines, achieving high QA-based evaluation scores while maintaining low R-QAE, which indicated both high quality and diversity in the generated QA pairs. This research highlights the potential of probabilistic generative models to enhance QA systems and inspired key aspects of our approach to fine-tuning large models for generating educational content.

Data Collection

The data for this project was collected from the SciQ dataset available on Kaggle. The SciQ dataset comprises 13,679 crowdsourced science exam questions covering subjects like Physics, Chemistry, Biology, and more. Each question is presented in a multiple-choice format with four answer options. In most cases, an additional paragraph is provided as supporting evidence for the correct answer, enhancing the dataset's educational value. The dataset includes the following columns:

- Question: The text of the question.
- distractor1, distractor2, distractor3: Three incorrect answer options for the question.
- correct_answer: The correct answer for the question.
- support: The supporting text that justifies the correct answer.

Before training, the following data cleaning steps were applied:

1. Questions missing answers or supporting evidence were discarded.
2. Applied preprocessing such as lowercasing, removing extraneous spaces, and fixing encoding issues.
3. Reducing Dataset to only using the necessary columns - Question, Correct_Answer, Support.

This rich and well-structured dataset facilitates the development of models aimed at generating high-quality, contextually accurate flashcards by leveraging the provided questions and supporting information. The data was divided into train and test sets using an 80/20 split.

Column name	Description
question	The question text. (String)
distractor3	One of the distractors for the question. (String)
distractor1	One of the distractors for the question. (String)
distractor2	One of the distractors for the question. (String)
correct_answer	The correct answer for the question. (String)
support	The supporting text for the question. (String)

question The question text. (String)	distractor3 One of the distractors for the question. (String)	distractor1 One of the distractors for the question. (String)	distractor2 One of the distractors for the question. (String)	correct_answer The correct answer for the question. (String)	support The supporting text for the question. (String)
11609 unique values	6960 unique values	6991 unique values	6900 unique values	5896 unique values	[null] 10% Coefficients are u... 0% Other (10479) 90%
What type of organism is commonly used in preparation of foods such as cheese and yogurt?	viruses	protozoa	gymnosperms	mesophilic organisms	Mesophiles grow best in moderate temperature, typically between 25°C and 40°C (77°F and 104°F). Meso...
What phenomenon makes global winds blow northeast to southwest or the reverse in the northern hemisp...	tropical effect	muon effect	centrifugal effect	coriolis effect	Without Coriolis Effect the global winds would blow north to south or south to north. But Coriolis m...

Overview of SciQ Dataset

We also leveraged additional datasets to enhance the model's performance by using pre trained T5 models on the following:

- SQuAD Dataset: Wikipedia's extractive Q&A pairs for generating context-aware questions and answers.
- RACE Dataset: Abstractive Q&A pairs from middle and high school English exams, enhancing question diversity and complexity.

Experiments

Baseline Model

The baseline architecture used in this project was a Recurrent Neural Network (RNN), which captures sequential dependencies in data using internal memory. The RNN model included an Embedding Layer, Simple RNN Layer, and Dense Layer for output prediction. RNNs were chosen as a baseline to compare against the more complex transformer model, T5, to highlight the performance improvements transformers offer for generating question-answer pairs from educational content.

The parameters for the RNN model were as follows:

- Optimizer: Adam
- RNN Units: 128
- Loss function: Sparse Categorical Cross Entropy
- Epochs: 10
- Batch Size: 64
- Neural Network Layers:
    - Embedding Layer
        - input_dim: vocab_size (size of the vocabulary)
        - output_dim: 100 (embedding dimension)
        - input_length: 100 (maximum length of sequences)
    - SimpleRNN Layer
        - units: 128
        - return_sequences: True
    - Dense Layer
        - units: vocab_size (size of the vocabulary)
        - activation: 'softmax'

The RNN model struggled with:

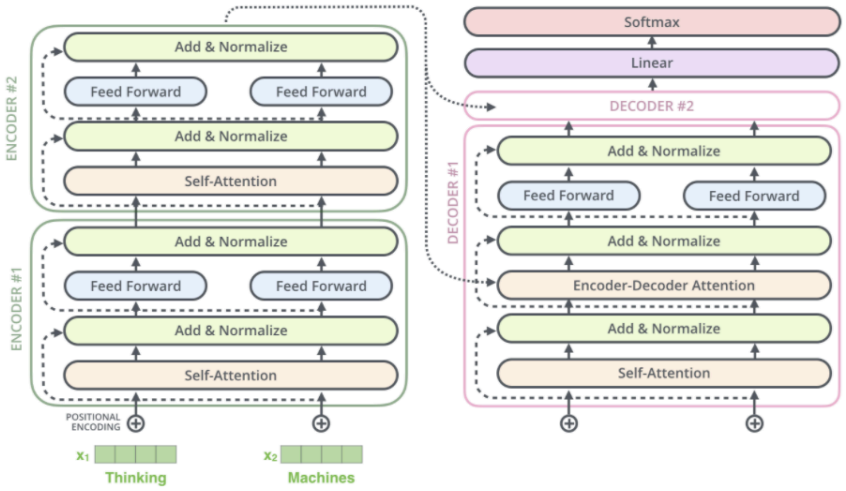
1. Capturing long-range dependencies, a limitation of its architecture.
2. Overfitting due to its simplicity and lack of regularization.
3. Performance, which was suboptimal compared to the transformer-based architecture.

Primary Architecture

For the primary architecture, the T5 (Text-to-Text Transfer Transformer) model was used. The pre-trained T5 models were obtained from Hugging Face (T5 Trained on SQuAD, T5 trained on RACE) and fine-tuned on SciQ. The fine-tuning process involved training the model on our dataset to generate meaningful question-answer pairs directly from educational texts. Additionally, Low-Rank Adaptation (LoRA) was employed to reduce the number of trainable parameters, making fine-tuning more memory and computationally efficient.

Model Details

The **T5 (Text-to-Text Transfer Transformer)** model is a versatile and powerful architecture in the family of transformer-based models, designed by Google Research. The key feature of T5 is that it frames every NLP task as a **text-to-text problem**, meaning both the inputs and outputs are always text. The model was pre-trained on massive datasets like SQuAD (extractive Q&A) and RACE (abstractive Q&A), allowing it to capture diverse linguistic and contextual nuances.



T5 Model Architecture

Parameters

- T5 Large: 770 million parameters
- Model Layers: 24 transformer layers
- Attention Heads: 16 attention heads per layer
- Learning Rate: 3e-4
- Batch Size: 4

- Epochs: 6
- Weight Decay: 0.01
- Warmup Steps: 200
- Learning Rate Scheduler: Linear

Fine-Tuning with LoRA

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. Using LoRA, we reduced the number of trainable parameters from 770 million to approximately 4 million, or 0.6% of the total parameters.

The fine-tuning process included the following steps:

1. The SciQ dataset was chosen for fine-tuning. This dataset includes context (supporting information), questions, and correct answers, making it well-suited for our task.
2. LoRA was applied to the attention layers of the T5 model, ensuring minimal disruption to the original model while enhancing its ability to generate high-quality question-answer pairs.
3. Key training parameters included learning rate, batch size, and number of epochs. The model was trained on both training and validation sets to ensure generalization to unseen content.

LoRA Config:

- r: 16
- lora_alpha: 32
- target_modules: ["q", "v"]
- lora_dropout: 0.1
- bias: none
- Trainable Parameters: 4,718,592 (0.64% of Total)

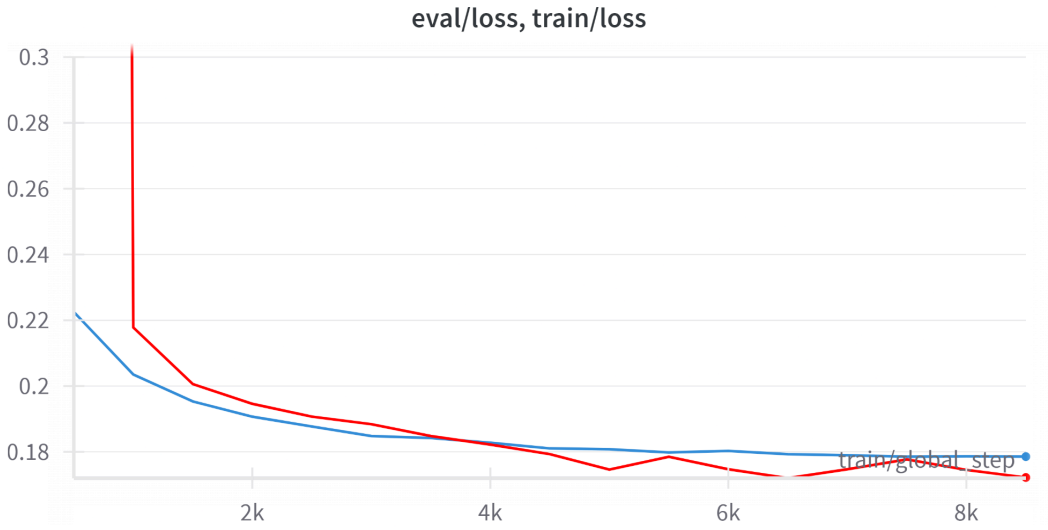
Training

Fine-tuning the T5 model involved using the T5 tokenizer to convert input text into tokenized sequences and generate word embeddings. We fine-tuned the model on the SciQ dataset using Low-Rank Adaptation (LoRA) to reduce the number of trainable parameters and improve memory efficiency. The training took approximately 6 hours, and the loss function was reduced to 0.16, indicating effective learning.

[8760/8760 5:56:20, Epoch 6/6]

Epoch	Training Loss	Validation Loss
1	0.180600	0.174946
2	0.171400	0.171466
3	0.156100	0.169538
4	0.150400	0.169569
5	0.153200	0.168435
6	0.134400	0.169165

Loss Table and Graph (Red- Train, Blue- Eval)



Test Set Metrics

```
{'precision': 0.9078778445720672, 'recall': 0.8525043427944183, 'f1': 0.8787162780761719}
```

Test Set Examples

```
Context:
Tree rings, ice cores, and varves indicate the environmental conditions at the time they were made.
Generated QA:
<pad> Tree rings, ice cores, and varves indicate what at the time they were made?<sep> environmental conditions</s>
--------------------------------------------------
Context:
Plant hormones are chemical signals that control different processes in plants.
Generated QA:
<pad> What are chemical signals that control different processes in plants?<sep> plant hormones</s>
--------------------------------------------------
Context:
Gametogenesis (Spermatogenesis and Oogenesis) Gametogenesis, the production of sperm and eggs, involves the process of meiosis.
Generated QA:
<pad> Gametogenesis, the production of sperm and eggs, involves the process of what?<sep> meiosis</s>
--------------------------------------------------
Context:
Figure 44.18 Deciduous trees are the dominant plant in the temperate forest. (credit: Oliver Herold).
Generated QA:
<pad> What is the dominant plant in the temperate forest?<sep> deciduous trees</s>
```

Challenges Faced

1. Training the T5 model required significant computational resources, mitigated by using Low Rank Adaption.
2. Managing lengthy input paragraphs required careful truncation without losing contextual meaning.
3. Fine-tuning even with LoRA demanded efficient resource management (30+ Hours of T4 GPU).

Pipeline Enhancements

To address input token limits, large documents are split into smaller chunks, with chunk size adjustable through the Streamlit web app. Overlapping chunks are used to preserve context between sections. Each chunk is summarized using a BART Large CNN model, ensuring that summaries remain within the 512-token limit. Beam search is employed to generate the most optimal question-answer pairs from the summaries. This method efficiently handles long PDFs (20+ pages).

We also experimented with Named Entity Recognition (NER) and Part-of-Speech (POS) tagging to enhance the quality of the generated output. Re-fine-tuning the T5 model on datasets tagged with these annotations revealed that POS tagging had a more significant impact on improving question-answer quality than NER.

Evaluation Metrics

The following metrics were chosen to evaluate the T5 model:

- ROUGE-L: Measures the longest common subsequence between generated and reference text, assessing syntactic similarity and content overlap.
- BERTScore: Utilizes BERT embeddings to evaluate semantic similarity, ensuring contextual accuracy and meaning alignment.

	RNN	**T5-Race Fine-tuned**	**T5-Squad Fine-tuned**	**T5-Squad Fine-tuned and POS tagging**
ROUGE-L	0.222	0.633	0.699	0.733
BERT Score	0.766	0.917	0.928	0.931

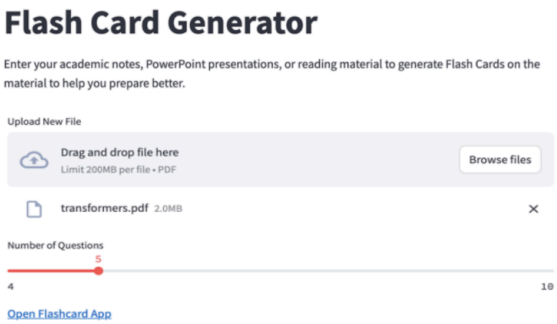
Results Table

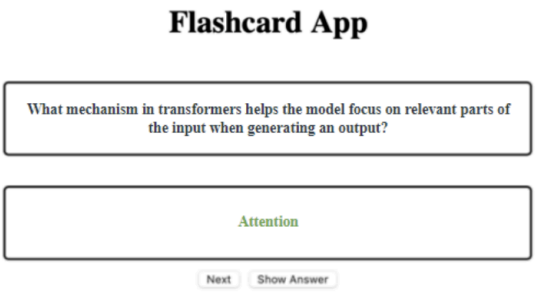
Web Application Development

To make the fine-tuned model accessible to users, a web application was developed using Streamlit for backend logic and Dash for GUI elements. The application includes several features:

- **PDF Upload**: Users can upload PDF documents (up to 200 MB) for processing.
- **Question-Answer Generation**: The uploaded document is processed by the fine-tuned model to generate a set of relevant question-answer pairs.
- **Flashcard Creation**: Users can convert the generated question-answer pairs into flashcards, where the questions are displayed, and the answers are revealed by clicking a button.

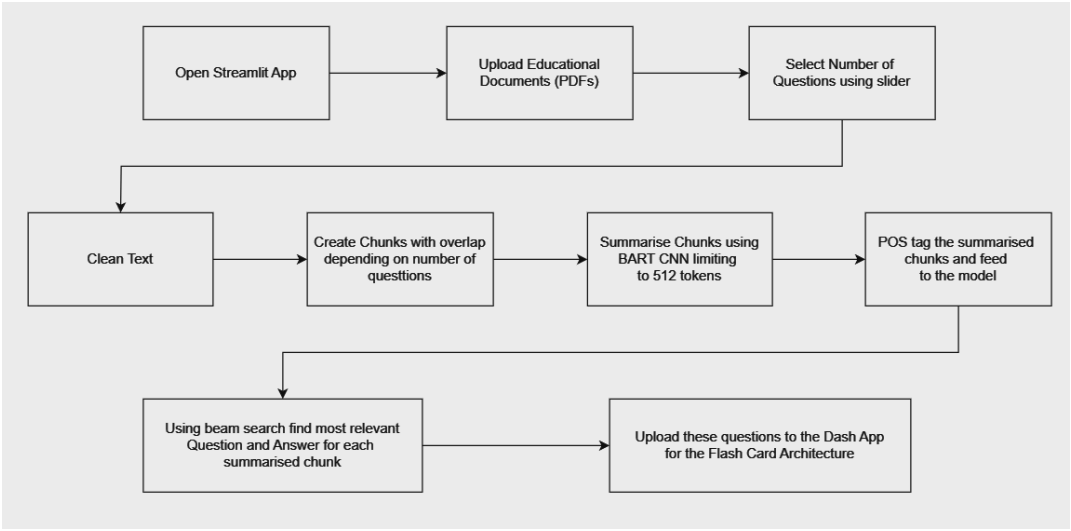
To address the challenge of time-consuming question-answer generation, once the pairs are created, they are stored in a text file for future use. Users can then upload this file to create flashcards without needing to regenerate the pairs. The application interface is designed to be user-friendly, with a slider to adjust the number of question-answer pairs to be generated and intuitive options for navigating through the flashcards.



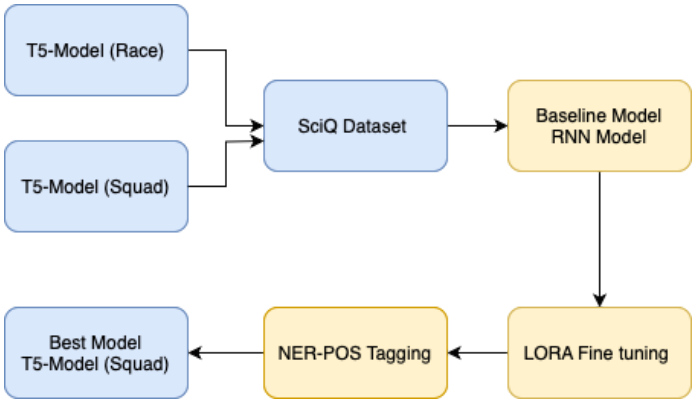


Built on: Streamlit, Dash

User Interface



Pipeline of the Project



Project Workflow

Conclusions

The project successfully demonstrated the potential of leveraging transformer-based models, enhanced with techniques like Low-Rank Adaptation (LoRA), to automate the generation of high-quality and contextually accurate flashcards from academic notes. By comparing the baseline RNN model with the fine-tuned T5 model, we observed significant improvements in the quality of generated question-answer pairs, as evidenced by evaluation metrics such as ROUGE-L and BERTScore.

Key takeaways from the project include:

- The use of pre-trained models like T5, fine-tuned on domain-specific datasets, significantly enhances the contextual understanding and relevance of generated flashcards.
- LoRA proved to be an efficient approach to fine-tuning large transformer models, drastically reducing computational and memory costs without compromising performance.
- The integration of Part-of-Speech (POS) tagging contributed to improved question-answer quality, highlighting the importance of linguistic features in text generation tasks.

The developed web application provides a practical and user-friendly solution for students to create flashcards directly from their study materials, addressing the challenges of traditional flashcard creation methods.

To improve the results, several approaches can be considered. Firstly, expanding the input format support to handle diverse document types, including audio or video content, could broaden the tool's applicability. Secondly, addressing input token constraints and file size restrictions would enable the model to process larger and more complex documents. This could be achieved by improving chunking strategies or adopting more advanced models that can handle longer sequences. Finally, optimizing the system’s performance by exploring more efficient tokenization and summarization techniques could reduce computational costs and improve speed, making the system more accessible and scalable.