FAQ Generation using Language Models

Dheeraj P Anikar
University of Southern California
Department of Electrical Engineering
anikar@usc.edu

Sudarshana S Rao
University of Southern California
Department of Electrical Engineering
ssudheen@usc.edu

Rbhu Gandhi
University of Southern California
Department of Computer Science
rbhugand@usc.edu

Abstract

Large Language Models (LLMs) have become a household word nowadays. LLMs have compelling applications, and this project explores the effectiveness of Large Language Models (LLMs) in generating Frequently Asked Questions (FAQs) solely from graduate school's admission requirements for Master of Science in Computer Science. T5-large, BART-large, LLaMA-2, and LLaMA-3 performances were investigated, employing a tailored dataset to examine their capabilities in an FAQ generation context. Each model was fine-tuned with data specifically oriented towards eliciting admission-related inquiries, aligning closely with the real-world information needs of prospective students.

To enhance the adaptation of the LLaMA models to this task Q-LoRA, a parameter-efficient approach for fine-tuning was used, which allowed the training of these huge models on a single GPU. Additionally, the transformation of admission texts into a prompt-style input format was experimented with. This modification was designed to explicitly guide the models in generating relevant questions by framing the input text as a source for potential FAQs. An example of the instruction prompt is, "Below is the admission text of a university; what are the possible FAQs that can be asked?"

The findings of this project indicate variations in performance across the models, with nuanced differences in the quality and relevance of the questions generated. These results contribute to understanding each model's applicational boundaries and strengths and suggest that framing input texts as explicit prompts can significantly influence the output, enhancing the practical utility of LLMs in educational settings. This research advances the field of NLP in an educational setting and provides insights for deploying LLMs to create supportive informational tools for academic institutions.

1 Introduction

The admissions process for graduate programs, particularly in highly competitive fields like Computer Science (CS), can be daunting and complex for prospective students. Navigating each university's specific requirements, deadlines, and expectations often demands extensive research and can lead to numerous FAQs. Traditionally, universities address this need through static FAQ sections on their websites or by contacting dedicated admissions staff for information. However, these resources might not always be exhaustive or easily searchable, leaving students with unresolved questions.

Recent advancements in LLMs present a compelling opportunity to streamline the process of generating common admission-related queries. LLMs, with their remarkable capacity for understanding and generating natural language, have the potential to automate the creation of comprehensive and informative FAQs. This project aims to investigate the effectiveness of different LLMs in generating FAQs directly from graduate CS university admissions texts.

1.1 Problem Statement

Applying to graduate programs, particularly in competitive fields like CS, can frustrate prospective students due to the lack of readily available answers. The need to decipher differing admissions requirements, deadlines, and expectations across universities leads to many FAQs.

1.2 Research Questions

- Can LLMs be effectively fine-tuned to generate relevant and accurate FAQs directly from graduate CS university admissions texts?
- Which LLM architectures demonstrate the highest performance in this specific FAQ generation task?

- Does framing the input texts as explicit prompts for FAQ generation improve the quality and relevance of the generated questions?

1.3 Motivation

This research aims to streamline the admissions process, reduce the workload for admissions staff, and ensure information consistency. Automating the creation of comprehensive FAQs with LLMs has the potential to reduce the cognitive burden placed on prospective CS students, making the admissions process more transparent. Additionally, LLM-powered FAQ systems could alleviate some of the workload for admissions staff, allowing them to focus on more complex student inquiries. Finally, universities can promote consistent and accurate information for all applicants by centralizing and automating FAQ generation with LLMs.

1.4 Related Work

In order to better understand the fundamental principles involving question generation, it is imperative to look into some of the researcher's work such that can develop a baseline of metrics and procedural steps that can be included in this work:

- In the article "A deep learning based end-to-end system (F-Gen) for automated email FAQ generation" by S. Jeyaraj and R. T. mentioned summarization techniques of the text that can be useful in determining the context of the dataset, which we firmly believe will be useful in our process of FAQs for admission process.[1] Moreover, in the paper, it was also found that for the evaluation of the model, the metrics used were ROUGE scores, which were implemented in the evaluation process as well.

- In the research process of finding a dataset of universities, it was found that there is no dataset available online or in any database to suffice the need for admission text and FAQs. So, the authors used a method adapted from "Generative Language Models for Paragraph-Level Question Generation" by Asahi Ushio et al., in which datasets like SQuAD are used. Thereby, it was decided to look into SQuAD and decided to build a dataset in a similar format.[2]

- It was researched that the Large Language Models have a wide variety of hyperparameters that impact the results in the generation process. Given this approach, the authors anticipated that they would run into issues of certain discrepancies, which they believe can be minimized by using fine-tuning techniques.[4]

- After the progress report, the authors of this article chose to test the efficacy of our approach by using Large Language Models such as Llama-2 & Llama-3. However, the training process is highly computationally intensive and will not be able to withstand GPU computing power itself. To resolve this, an article talks about Q-LoRa, which is an example of PEFT(Parameter Efficient Fine-Tuning Techniques), which are mainly used for these large models.[5]

2 Dataset

A list of the top 150 US universities for the CS program based on USNews rankings was compiled. The university names and website URLs were stored in a CSV file. Using the BeautifulSoup library, web scraping on each URL was performed, extracting admission requirements and existing FAQs. This collected data was then structured into a JSON file for compatibility with the language model training process. The dataset was split into training, validation, and testing sets as per the ratio 80%:10%:10%. The JSON file format is shown as follows.

"Stanford University": { "FAQs": ["Do you have to be a Computer Science undergraduate major to apply?", "If I already have an M.S. or PhD degree in Computer Science from another institution, may I apply to the M.S. or PhD program at Stanford?", ...],

"AdmissionText": ["Email forwarding for @cs.stanford.edu is changing on Feb 1, 2024. More details here", "The MS program is excellent preparation for a career as a computer professional or for future entry into a Ph.D. program at Stanford or elsewhere. Individual programs can be structured to consist entirely of coursework or to involve some research. Students who are more interested in research may pursue a M̈.S. degree with distinction in research.̈ See the R̈esearch or Project Requirementsection of the Stanford Bulletin for more information." .....] },

3 Data Pre-Processing

Data pre-processing is critical in any data-driven project, significantly when leveraging large lan-

guage models (LLMs) for generating FAQs. Preprocessing aims to clean and organize raw data into a structured format that enhances the quality of the inputs for model training or analysis. This step involves dealing with missing data, transforming text data, and aggregating information to align with this project's objectives.

Two ways of data structuring were explored, which are the inputs to fine-tuning the Language models. The two data pre-processing techniques are called Method-1 and Method-2.

3.1 Method-1

This method is minimal and only does the fundamental transformations to the dataset. The steps followed are as follows.

- **Loading the data**: Import the dataset from a CSV file into a pandas DataFrame, providing a flexible structure for tabular data manipulation.
- **Handling missing values**: Replace all missing values in the dataset with the string 'No data available' to ensure no entries are left empty and to indicate missing information.
- **Transforming the FAQs**: Modify the FAQ entries, initially separated by "|", by splitting each string on this delimiter and joining the results into a continuous string. This prepares the text for better processing as a cohesive block.
- **Aggregating data by university**: Group the dataset by 'University Name' and aggregate it to consolidate each university's data

3.2 Method-2

Method-2 is more comprehensive, extends on the data pre-processing done in Method-1, and does much more in cleaning and processing the scrapped data than Method-1. This method integrates advanced text processing and summarization techniques using GPT-3.5 to pre-process university admission texts and FAQs. The goal is to generate clean, well-structured inputs that enhance the training and performance of a language model tasked with FAQ generation. The below steps were followed.

- **Text summarization with GPT-3.5**: GPT 3.5 was leveraged to summarize the lengthy admission texts. The summarization focuses on retaining content relevant to frequently asked questions, thus eliminating unnecessary details and reducing data size. This results in a distilled version of the admission texts, referred to as the "Summary" column, which contains the essence of the information needed for effective FAQ generation. Figure 1 illustrates the process of text summarization.

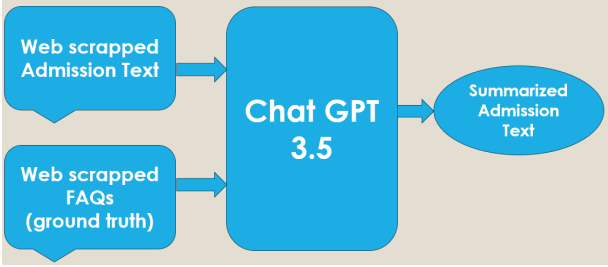

Figure 1: Admission Text Summarization

- **Text cleaning**: The following steps were involved in data cleaning:
  - **Removal of HTML tags**: Clean HTML content using BeautifulSoup, ensuring that only text content is retained.
  - **Normalize unicode characters**: Convert characters to their closest ASCII representation to maintain text uniformity and avoid encoding issues.
  - **URLs and emails**: Replace URLs and email addresses with placeholders to prevent model confusion and protect privacy.
  - **Non-ASCII character removal**: Strip out non-ASCII characters to standardize text for model processing.
  - **Punctuation handling and whitespace normalization**: Clean up the text by removing unnecessary punctuation and reducing multiple spaces to a single space, creating cleaner, more readable text.
- **FAQs parsing and formatting**: Modify the FAQ entries, initially separated by "|", by splitting each string on this delimiter and joining the results into a continuous string. This prepares the text for better processing as a cohesive block.
- **Training prompt generation**: Combine the cleaned, summarized admission text and the structured FAQs to complete the training

prompt. This is done to give the model more direction as to what to do for the given input text.

Figure 2 depicts the text summarization by GPT 3.5 and example prompt instruction.

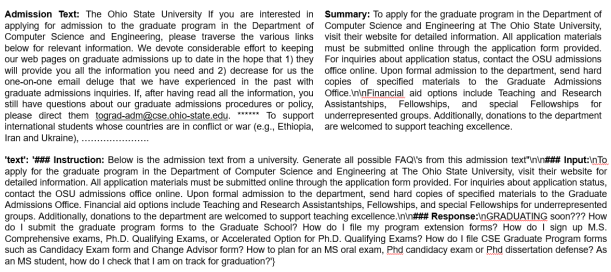

Figure 2: Data Pre-Processing Method-2

4 Models

A total of four LLMs were experimented in this project. Out of which, two were encoder-decoder architectures, and the other two were decoder architectures.

4.1 Encoder-decoder Models

Encoder-decoder models, a type of sequence-to-sequence neural network architecture, excel at tasks where the input and output have different lengths or structures. They are a powerful tool for generating new FAQs directly from a knowledge base or raw text data. The encoder analyzes the existing text corpus and relationships and identifies patterns. The decoder then leverages this encoded information to generate well-structured FAQs. In this regard, T5-large and BART-large models were chosen to generate FAQs.

T5 (Text-to-Text Transfer Transformer) has 770 million parameters. A key distinguishing feature of T5 is its text-to-text framework, which reframes the FAQ generation task as a single text input and output problem.

BART (Bidirectional and Auto-Regressive Transformer) has 406 million parameters. Built on the Transformer architecture, it's designed explicitly as a denoising autoencoder. T5 and BART have been pre-trained on a massive text corpus and can be fine-tuned for tasks like summarization, question answering, and text generation.

4.2 Decoder Models

While encoder-decoder models are highly effective in tasks where a transformation of information from one form to another is required, they may not always excel in pure generation tasks to the same extent as decoder models. Thus, decoder models were decided to explore, such as Llama-2 and Llama-3. The autoregressive nature of decoder-only models, where each token is predicted sequentially based on all previous tokens, allows for a more focused generation process. This characteristic is particularly beneficial for question generation.

Llama 2 is a family of pre-trained and fine-tuned LLMs released by Meta AI in 2023. Llama 2 models can perform various NLP tasks, from text generation to programming code. Llama 2 models offer a context length of 4,096 tokens, double that of LLaMa 1. The model chosen to fine-tune is the Llama-2 7 billion parameter-sized model.

The new 8B and 70B parameter Llama 3 models are a significant leap over Llama 2 and establish a new state-of-the-art for LLM models at these scales. Thanks to improvements in pretraining and post-training, Llama-3 pre-trained and instruction-fine-tuned models are the best today at the 8B and 70B parameter scales.

Input to T5-large and BART-large is the dataset obtained from method-1 pre-processing. Figure 3 shows the input and expected output for T5 and BART.

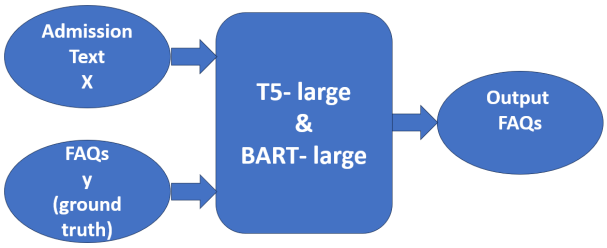

Figure 3: Encoder-Decoder Models

On the other hand, the input to Llama-2 and Llama-3 is the dataset obtained from method-2 pre-processing. Figure 4 shows the input and expected output for Llama-2 and Llama-3.

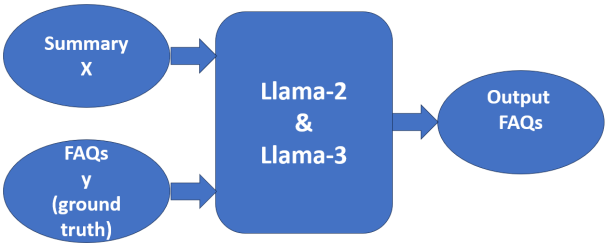

Figure 4: Decoder Models

5 Fine-Tuning

The four models, T5-large, BART-large, Llama-2 7b, and Llama-3 8b, were fine-tuned. The model's hyperparameters were experimented with for optimal results.

5.1 T5 and BART

Table 1 shows the fine-tuned hyperparameter values used for the T5-large model.

Hyperparameters	Values
Learning rate	5e-5
Batch size	2
Epochs	5
Optimizer	AdamW

Table 1: Hyperparameters used for T5

Table 2 shows the fine-tuned hyperparameter values used for the BART-large model.

Hyperparameters	Values
Learning rate	5e-5
Batch size	4
Epochs	5
Optimizer	AdamW

Table 2: Hyperparameters used for BART

5.2 Llama-2 and Llama-3

Fine-tuning larger models like Llama-2 and Llama-3 on a single GPU similar to T5 and BART isn't possible, as the model parameters are huge and take up a lot of memory. The Llama-2 model used in this project has 7 billion parameters. In contrast, the Llama-3 model has 8 billion parameters, which is enormous compared to the million parameter-sized models such as BART and T5. Thus, Q-LoRA (Quantized Low-Rank Adaptation), an extension of LoRA (Low-Rank Adaptation), a Parameter Efficient Fine-Tuning technique, was leveraged.

5.2.1 Parameter Efficient Fine-Tuning

Parameter Efficient Fine-Tuning, also popularly known as PEFT, allows for efficiently fine-tuning large models on consumer hardware. This technique reduces the number of trainable parameters in the model. It achieves this by fine-tuning only a tiny number of (extra) tunable parameters by keeping most of the model parameters fixed (frozen). One of the drawbacks of regular fine-tuning is that by updating the weights, which result from pretraining, the model is used. However, it gets better at the downstream task, which is fine-tuned and might also lead to losing some of the understanding it might have gained in pretraining. This is known as catastrophic forgetting.

PEFT overcomes the problem of catastrophic forgetting, often encountered during regular fine-tuning of large language models. PEFT also works well in low-data regimes. LoRA is the PEFT technique that was used in this project.

5.2.2 LoRA

Low-Rank Adaptation, popularly known as LoRA, is a type of PEFT technique that is based on Low-rank Matrix Factorization, $A_{M*N} = A1_{M*K} * A2_{K*N}$. If the matrix is not full rank, a matrix of dimensions M*N can be factorized into two smaller matrices, M*K and K*N. Here, the dimension 'k' is the rank of the matrix.

After adapting to a downstream task, LLMs were found to have low intrinsic dimensions, according to A. Aghajanyan et.al in "Intrinsic dimensionality explains the effectiveness of language model fine-tuning". This implies that a less-than-full rank matrix can represent the weight parameter space. Thus, the weight matrix can be decomposed into two matrices of M*K and K*N (where K is the rank of the matrix). This K is a hyper-parameter chosen during fine-tuning; the two new decomposed matrices (say W_A and W_B) are learned through the fine-tuning process via backpropagation.

These decomposed matrices W_A and W_B, which are learned via backpropagation, are learned separately, i.e., the total tunable parameters become M*K + K*N, which is much much lesser than M*N (the range for K is around 16-64). This reduces the number of tunable parameters by a factor of at least 1000, if not even more.

As K is the hyper-parameter chosen for fine-tuning, it determines the rank. Low K implies the decomposed matrices are of lower rank, which results in a lower quality fine-tuned model but much fewer computational resources used. Thus, choosing the correct value of K becomes an essential consideration during LoRA.

5.2.3 Quantized - LoRA

Q-LoRA is an extension of LoRA. Typically, the trained model parameters are of 32-bit precision; this method quantizes the precision of weight parameters in the pre-trained LLM to 4-bit precision.

Thus saving memory even further.

After various trials and errors, the hyperparameters that yielded good results from Llama-2 and Llama-2 are shown in table 3.

Hyperparameters	Values
Learning rate	5e-5
Batch size	2
Epochs	4
Optimizer	AdamW
Rank(K)	16
Alpha	64
Temperature	1.1

Table 3: Hyperparameters used for Llama-2 & Llama-3

6 Results

The following subsections outline the four models' evaluation metrics and loss curves.

6.1 T5-large

Table 4 depicts the BERT and ROUGE scores of the T5.

Metric	Values (F1 scores)
BERTScore	-0.086
ROUGE-1	0.10318908708354294
ROUGE-2	0.009371644453146357
ROUGE-L	0.08980406630721782

Table 4: T5 Results

Figure 5 illustrates the training and validation loss curve of the T5 model. Initially, training and validation loss are high, gradually decreasing over epochs.

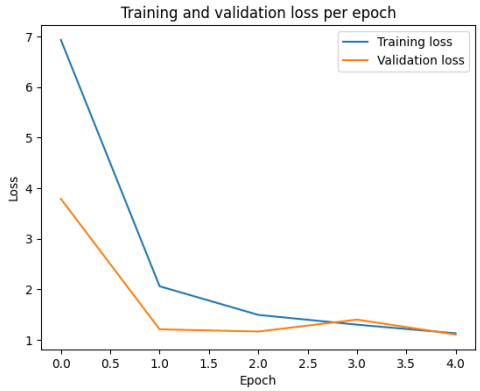

Figure 5: Loss Curve of T5

Figure 6 shows the output of the T5-large model.

Interested in a Ph.D. program in computer science? Are you interested in earning specialized training in the field? Is there grad school for computer scientists? What is the minimum GPA requirement for obtaining baccalaureate degree? How do I apply for the Masterâ€™s in Computer Science program? Can I transfer to another university? Do I need to have CS majors to apply? Does the program require BS or MAT? Will I be admitted to the masterâ€™ s degree program at Illinois Tech? Yes,! Can you apply to earn my application? The program is accredited by the application to get in engineering?

Interested in Engineering? - a Master's degree in engineering is required to be admitted to the School of Engineering. â€“ if you are interested in pursuing grad school, you must apply to UC Santa Clara. The School offers the following programs: Bachelor' s Degree (or equivalent) / Master of Science (with physics or chemistry) or Master in Science in Mathematics (MS) with CS or equivalent (M.S.) in any of the above fields. * Applicants must be enrolled

Figure 6: T5 Generated FAQs

6.2 BART-large

Table 5 depicts the BERT and ROUGE scores of the BART.

Metric	Values (F1 scores)
BERTScore	0.225
ROUGE-1	0.29364804888577556
ROUGE-2	0.09333228748923009
ROUGE-L	0.20378157768427876

Table 5: BART Results

Figure 7 illustrates the BART model's training and validation loss curve. Initially, training and validation loss are high, gradually decreasing over epochs.

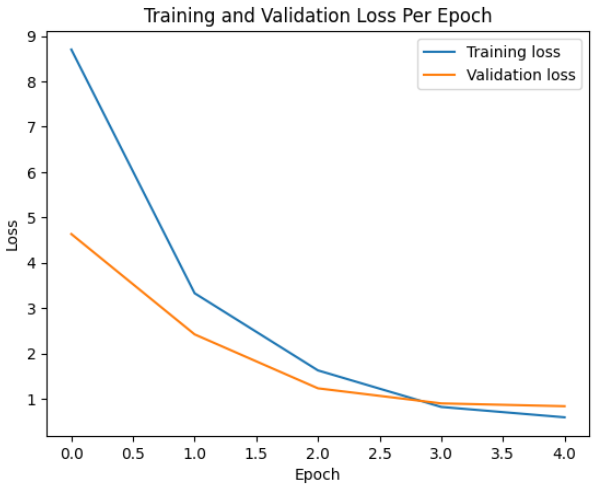

Figure 7: Loss Curve of BART

Figure 8 shows the output of the BART-large model.

How do I apply to the IMS program? What is the difference between the Graduate School and the School of Engineering? Do I need to have a master's degree in order to be considered for admission? Is there a minimum GPA requirement? Can I still apply for a Master's degree if I am not enrolled in a graduate program at Vanderbilt University?

What is the difference between the graduate programs at the University of Oklahoma and the U.S. Department of Education? How do I apply? What are my chances of being admitted? Do I need a bachelor's degree in order to be admitted to the School of Computer Science?

Figure 8: BART Generated FAQs

6.3 Llama-2

Table 6 depicts the BERT and ROUGE scores of the Llama-2.

Metric	Values (F1 scores)
BERTScore	0.83948094
ROUGE-1	0.22865427281133038
ROUGE-2	0.07720491898694384
ROUGE-L	0.16829442985815585

Table 6: Llama-2 Results

Figure 9 illustrates the training and validation loss curve of the Llama-2 model. Initially, training and validation loss are high, gradually decreasing over epochs.

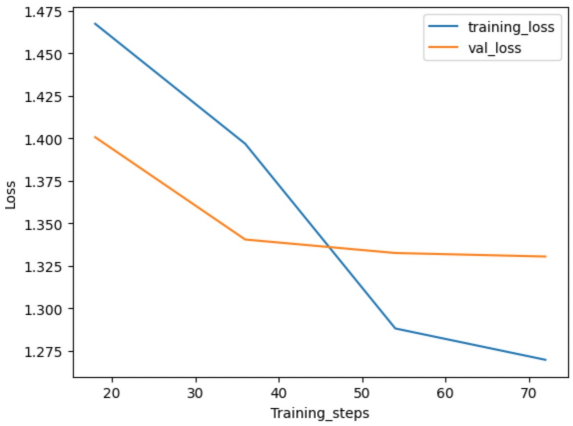

Figure 9: Loss Curve of Llama-2

Figure 10 shows the output of the Llama-2 model.

What is a thesis option? How long does it take to complete the MSCS program? Can I complete the program in less than two semesters? Can I transfer credits from another institution? How many credits do I need to graduate? Can I take courses as a non-degree student? Can I enroll in courses without being admitted to a degree program? Do I have to be a full-time student to be eligible for financial aid? Can I apply for financial aid if I am not a U.S. citizen or permanent resident? How much is tuition and what are the fees? Are there scholarships or fellowships available for graduate students? Can I work while I am enrolled in the program? Is there on-campus housing available? Can I bring my family to East Lansing? Can I stay in the United States after I graduate? What is my visa status? Can I travel outside of the United States while I am a student at MSU? Can I extend my I-20 or DS-2019? Can I change my status from F-1 or J-1 to H-1B?

Figure 10: Llama-2 Generated FAQs

6.4 Llama-3

Table 7 depicts the BERT and ROUGE scores of the Llama-3.

Metric	Values (F1 scores)
BERTScore	0.83694977
ROUGE-1	0.19798938538277866
ROUGE-2	0.060723281389197734
ROUGE-L	0.16688194707468773

Table 7: Llama-3 Results

Figure 11 illustrates the training and validation loss curve of the Llama-3 model. Initially, training and validation loss are high, gradually decreasing over epochs.

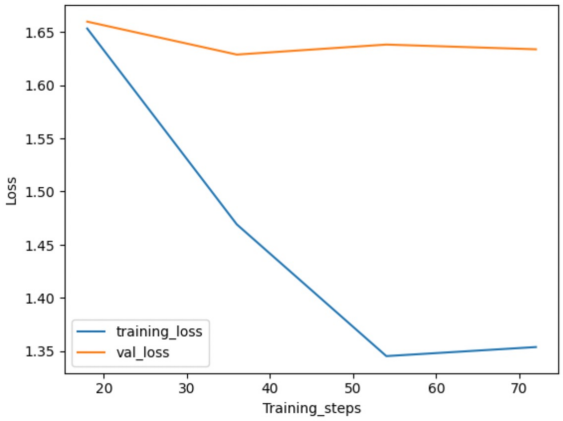

Figure 11: Loss Curve of Llama-3

Figure 12 shows the output of the Llama-3 model.

Do I need a degree in Computer Science to be admitted to the MSU CSE graduate program? What are the prerequisites for the program? How do I apply to the program? When are the application deadlines? What supporting documents do I need to submit with my application? What are my chances of being admitted if I don't have a computer science degree? How are admissions decisions made? I'm an international applicant. Are there any special considerations for me? Are there fellowships or scholarships available for graduate students in the CSE program? If I'm admitted to the program, how do I get reimbursed for the application fee? Where do I send my application materials? Where can I get more information about the program?

Figure 12: Llama-3 Generated FAQs

6.5 Zero-shot prompting of Llama-2 and Llama-3

The results of Llama-2 and Llama-3 obtained by zero-shot prompting were very similar. Hence, only one tabular column containing the BERT and ROUGE scores and one image depicting the generated FAQs are shown.

Table 8 depicts the BERT and ROUGE scores of zero-shot prompting.

Metric	Values (F1 scores)
BERTScore	0.858526
ROUGE-1	0.24188042631490989
ROUGE-2	0.07603239677051529
ROUGE-L	0.21875964758276956

Table 8: Zero-Shot Prompting Results

Figure 13 shows the output of zero-shot prompting.

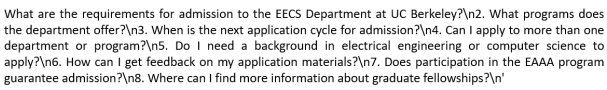

Figure 13: Zero-Shot Prompted FAQs

BERT scores clearly reflect the performance of each model for FAQ generation as it focuses on semantic matching (cosine similarity) rather than lexical matching. ROUGE scores were calculated for each model because every research paper related to this project had computed ROUGE scores. Therefore, BERT and ROUGE scores were decided as the evaluation metrics for this project.

7 Conclusions

The FAQs generated by T5-large and BART-large will be considered for comparison as these two models were fine-tuned using the dataset obtained from method-1's pre-processing technique. Also, the architectures of these two models are similar, and comparing the outputs of these two models would make more sense. Conversely, the FAQs generated by Llama-2 and Llama-3 will be compared with those obtained from zero-shot prompting of Llama-2 and Llama-3.

It is evident that the FAQs generated by the BART-large model are much better than those of the T5-large. Moreover, the BERT and ROUGE scores of BART-large are better than those of the T5-large. BART was able to generate questions (not necessarily FAQs) for every single university in the testing dataset, whereas the T5-large did not generate FAQs for every single university. One reason for the better performance of BART is the fact that BART has a denoising auto-encoder architecture, while the T5 has an encoder-decoder architecture. BART is explicitly trained to generate from noisy input. Conversely, the T5 is found to be useful in generic text-to-text generation. Also, BART is bidirectional, meaning that BART can read the input from both sides of the sentence. Finally, the dataset used in this project is small and since BART is a smaller model compared to the T5, BART was able to generate from a small dataset.

Although good results were achieved by fine-tuning Llama-2 and Llama-3 models, these models also performed comparably even when they were zero-shot prompted. This implies fine-tuning Llama-2 and Llama-3 for this task might be unnecessary. This is possibly due to the large scale of these models, which imply a better general language understanding, and their knowledge cut-off being more recent. However, all four models were able to generate FAQs in general, which were comparable with the FAQs in the ground truths. This shows that LLMs can be fine-tuned for FAQ generation tasks provided with a larger dataset. It is not unreasonable to expect models like BART and T5 to give significantly better results (with a larger dataset) than the results achieved with a small dataset.

To encapsulate, this project was able to answer all the research questions mentioned earlier. LLMs can be effectively fine-tuned to generate relevant and accurate FAQs directly from graduate CS university admission texts. Llama architecture demonstrated the highest performance in FAQ generation. Framing the input admission texts explicitly as instruction prompts does improve the quality and relevance of the generated FAQs.

8 Future Recommendations

- The use case of FAQs can extend way beyond the scope of the admission process to any other industry that requires the formation of FAQs based on the given text. For instance, this project can be incorporated in any service-based industry, which takes a lot of feedback from users in the form of questions and concerns, and based on user-specified metrics, that text can be used to generate FAQs.

- Thus far, the validation of this method has been tested only in English. But it can be used in a multilingual setup as well.

- Moreover, the authors believe it can also be accompanied by the generation of question-and-answer pairs rather than generating just questions themselves.

References

1. S. Jeyaraj and R. T., "A deep learning based end-to-end system (F-Gen) for automated email FAQ generation," Expert Systems With Applications, vol. 187, 2022, https://doi.org/10.1016/j.eswa.2021.115896.

2. AsahiUshio, Fernando Alva-Manchego, and Jose Camacho-Collados, "Generative Language Models for Paragraph-Level Question Generation," Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 670-688 December7-11, 2022, https://aclanthology.org/2022.emnlp-main.42.pdf.

3. Best Computer Science Schools, U.S.News, https://www.usnews.com/best-graduate-schools/top-science-schools/computer-science-rankings (accessed March 3, 2024).

4. A. Aghajanyan, L. Zettlemoyer, and S. Gupta, "Intrinsic dimensionality explains the effectiveness of language model fine-tuning" arXiv.org, https://arxiv.org/abs/2012.13255 (accessed Apr. 30, 2024).

5. T. Dettmers, A. Pagnoni, and L. Zettlemoyer, "QLORA: Efficient Finetuning of Quantized LLMs." https://arxiv.org/pdf/2305.14314 (accessed: April 30, 2024)

A Appendix

The GitHub repository of this project can be found https://github.com/Rbhu376264/FAQGenerator/tree/main.