Subjective Question Generation and Answer Evaluation Using NLP

by

Ahmed Symum Swapno
20101308
Mohammad Rafid Hamid
20101491
Safwan Shaheer
22241148
Yaseen Nur
22241147

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
Fall 2023

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Ahmed Symum Swapno
Ahmed Symum Swapno
20101308

Mohammad Rafid Hamid
20101491

Safwan
Safwan Shaheer
22241148

Yaseen Nur
22241147

Approval

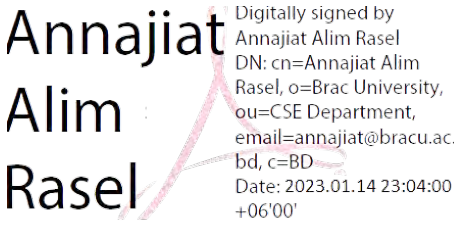
The thesis titled "Subjective Question Generation and Answer Evaluation Using NLP" submitted by

1. Ahmed Symum Swapno (20101308)
2. Mohammad Rafid Hamid (20101491)
3. Safwan Shaheer (22241148)
4. Yaseen Nur (22241147)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on May 22nd, 2023.

Examining Committee:

Supervisor:



(Member)

Mr. Annajiat Alim Rasel
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Natural Language Processing (NLP) is one of the most revolutionary technologies today. It uses artificial intelligence to understand human text and spoken words. It is used for text summarization, grammar checking, sentiment analysis, and advanced chatbots and has many more potential use cases. Furthermore, it has also made its mark on the education sector. Much research and advancements have already been conducted on objective question generation; however, automated subjective question generation and answer evaluation are still in progress. An automated system to generate subjective questions and evaluate the answers can help teachers in assessing student work and enhance the learning experience of the students by allowing them to self-assess their understanding after reading an article or a chapter of a book. This research aims to improve current NLP models or make a novel one for automated subjective question generation and answer evaluation from text input.

Keywords: Question Generation; Subjective Question Generation; Answer Evaluation; Automatic Short Answer Grading; NLP; Machine Learning;

Acknowledgement

First and foremost, Alhamdulillah, all praises belong to Allah the Almighty, the Most Gracious, and the Most Merciful for His blessings, guidance, and mercy that have been bestowed upon us throughout our research journey. We are immensely grateful for His divine providence, enabling us to complete this thesis. We would like to thank Mr. Annajiat Alim Rasel for their guidance and support throughout the research process. We are also extremely grateful to our family and friends for their input, suggestions, and general support.

Table of Contents

Declaration i

Approval ii

Abstract iii

Acknowledgment iv

Table of Contents v

List of Figures vii

Nomenclature viii

1 Introduction 1
- 1.1 Research Challenges . . . 2
- 1.2 Research Objectives . . . 3

2 Related Work 5
- 2.1 Question Generation . . . 5
- 2.2 Answer Evaluation . . . 7
- 2.3 LLMs . . . 8

3 Data Generation 11
- 3.1 Dataset Overview . . . 11
- 3.2 Question Generation Dataset . . . 11
- 3.3 Answer Evaluation Dataset . . . 12
- 3.4 Dataset Generation Process . . . 13
  - 3.4.1 Question Generation Prompt . . . 14
  - 3.4.2 Answer Evaluation Prompt . . . 14

4 Methodology 21
- 4.1 Instruct tuning Model . . . 21
- 4.2 Importance of Instruct Tuning Model . . . 21
- 4.3 LoRA (Low-Rank Adaptation) . . . 21
  - 4.3.1 Quantized Low-Rank Adaptation (QLoRA) . . . 23
  - 4.3.2 4-bit NormalFloat Quantization . . . 23
  - 4.3.3 Bfloat16 Floating Point Precision . . . 23
- 4.4 Model Training . . . 24
  - 4.4.1 Trained Models . . . 24

4.4.2 GPU Specifications 25
4.4.3 Hyperparameters 25
4.4.4 Training Settings 26
4.4.5 Training Prompts 26
4.4.6 Training Process 27
4.4.7 Training and Validation Loss (Answer Evaluation) 28
4.4.8 Training and Validation Loss (Question Generation) 28

5 Result Analysis **29**
5.1 Model Evaluation Metric 29
5.2 Question Generation 29
5.3 Answer Evaluation 32

6 Conclusion **36**
6.1 Future Work 36

Bibliography **39**

List of Figures

4.1 Full Workflow 22
4.2 Train Loss Graph 28
4.3 Validation Loss Graph 28
4.4 Train Loss Graph 28
4.5 Validation Loss Graph 28

5.1 Question Generation Model Ranking Heatmap. Rows denote models and the columns denotes the ranks. Each square shows the percentage of times the model of its row placed in the rank of its column. 30
5.2 Distribution of Question Generation Model Placement for Each Rank 31
5.3 Distribution of Ranking for Each Question Generation Models 31
5.4 Answer Evaluation Model Ranking Heatmap. Rows denote models and the columns denotes the ranks. Each square shows the percentage of times the model of its row placed in the rank of its column. 33
5.5 Distribution of Answer Evaluation Model Placement for Each Rank . 34
5.6 Distribution of Ranking for Each Answer Evaluation Models 34

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$ASAG$ Automatic Short Answer Grading

ASR Automatic Speech Recognition

$BLEU$ Bilingual Evaluation Understudy

GPT Generative Pretrained Transformer

LLM Large Language Model

$LSTM$ Long Short Term Memory

MCQ Multiple Choice Question

NLP Natural Language Processing

QED Quod Erat Demonstrandum

$RLHF$ Reinforcement learning with Human Feedback

RNN Recurrent Neural Network

$ROGUE$ Recall-Oriented Understudy for Gisting Evaluation

SQG Subjective Question Generation

$SQuAD$ Stanford Question Answering Data set

SVM Support Vector Machine

Chapter 1

Introduction

Natural Language Processing is a rapidly growing field that has significantly contributed to various industries, including education. The field of NLP deals with the analysis and generation of human language. The technology can process and understand natural language input and output, providing significant opportunities for innovation in various sectors. The education sector is no exception, as NLP is used to create innovative Education Technology (EdTech) solutions that can enhance how we learn, teach, and evaluate.

NLP allows education technology to understand human language, just like humans, and use that understanding to provide a more engaging and personalized learning experience. With NLP, the technology can understand students' queries and respond with personalized feedback, explanations, and guidance, providing students with a more interactive and practical learning experience. NLP can also create customized learning experiences by analyzing students' writing, speech, and behaviour to identify their strengths and weaknesses. With this information, educational systems can recommend relevant resources and activities to support students' learning.

Another vital application of NLP in education is the generation of questions and evaluation of the answers. NLP can help create questions that test students' understanding of a topic and can provide personalized feedback on their answers. This feature enables teachers to assess students' knowledge of a topic more effectively. NLP can also be used to evaluate students' writing and speech, providing insights into their language proficiency and areas of improvement.

Questions play a vital role in the education process, as they serve as a means of assessment, testing students' knowledge and understanding of a particular topic. They can also be used as a tool for self-assessment, helping individuals deepen their understanding of a book or article.

In recent years, there has been a significant amount of research on question generation and evaluation using NLP techniques. However, much of this work has focused on objective questions—True and False, fill in the blanks, MCQ, snippets— which pertain to a specific value or piece of information. On the other hand, Subjective questions are open-ended questions that require personal judgment and interpretation rather than objective facts or information. Subjective questions are more

challenging to answer than objective ones because they require a deeper understanding of the topic and require critical thinking, analysis and personal judgment. These questions can help evaluate students' ability to analyze, compare, interpret and evaluate information, which are essential skills for success in many careers and lifelong learning.

This research will focus on Subjective Questions since they are more important to judge the cognitive abilities of the students. The aim is to assists the teacher in setting the questions for the exams efficiently and conveniently while giving the students a tool to practice and self-evaluate themselves. We plan to use NLP in order to automate the process of subjective question generation and the evaluation of students' answers. In particular, we will be exploring the potential of Large Language Models in these tasks. **This research aims to provide an automated system for generation of Subjective Questions from a given context and evaluation of students' answers to the generated questions by instruct-tuning LLMs.**

In the following sections, we describe the research challenges, the research objectives, a detailed literature review, the data generation process and the description of data, the description of the models, in depth analysis of the results and concluding remarks along with future works.

1.1 Research Challenges

The goal is to generate grammatically and semantically correct questions requiring a subjective or opinion-based answer, as opposed to a factual or objective answer, as well as engaging and relevant to a given topic or context. The input to the model is typically a piece of text or a set of keywords that provide context for the question, and the output is a natural language question. This problem is challenging because it requires the model to understand the meaning of the input text and generate grammatically and semantically correct questions relevant to the context.
Understanding context is crucial when generating subjective questions. Context provides the input text's background information and meaning, which is crucial for generating relevant and appropriate questions. For example, if the model is given the input text, "The new iPhone has a great camera", a relevant question would be, "What do you think of the camera on the new iPhone?". This question is relevant because it is in context with the input text and asks for an opinion about the camera. On the other hand, if the model is given the exact input text and generates the question "What is the color of the new iPhone?", it would not be relevant because it is not in context with the input text. It asks for a factual answer, not an opinion.

The model needs to generate open-ended questions rather than closed-ended ones that can be answered with a simple "yes" or "no". Furthermore, it should generate questions that elicit a subjective or opinion-based answer rather than a factual one. For example, given the input text "The new Star Wars movie was just released", a relevant open-ended question would be "What did you think of the new Star Wars movie?" whereas a closed-ended question would be "Did you like the new Star Wars movie?"

Multiple questions can be generated from a single input text, some of which might be more subjective and open-ended than others. The model should handle this ambiguity and generate a question that is most relevant and appropriate for the given context. Existing ASAG systems need help to grade and understand natural language responses accurately.

There are several challenges associated with this problem:

- **Quality of Scores**: One of the main challenges is that the quality of the scores generated by ASAG systems remains a concern. Even after decades of research, a lot can still be done to make these systems give improved scores.
- **Development Time and Consistency**: Another challenge is the time and resources required to develop an automated solution for ASAG. Additionally, consistency is challenging when using ASAG, as the system's ability to accurately grade responses may vary depending on the question and the responses it receives.
- **Bias and Fairness**: If the model is trained on a dataset that contains bias, it may perpetuate that bias in its predictions and produce unfair results, particularly for minority or underprivileged groups. As a result, the system may grade responses unjustly.
- **Lack of interpretability**: There needs to be more interpretability in existing ASAG systems, making it difficult for educators to understand and evaluate the system's decisions, identify and address biases, and fine-tune the model.
- **Human judgement factor**: The ability of the system to replace human judgement remains a challenge as humans can evaluate the response considering various factors such as context, intonation, and background knowledge and may have a more holistic understanding of the response.

These challenges highlight the need for further research in the field of ASAG to improve the accuracy and interpretability of these systems, in addition to addressing the challenges of bias, fairness, development time and consistency. By investigating advanced techniques such as reinforcement learning with human feedback, this research aims to address these challenges and develop methods for improving the performance and interpretability of ASAG systems.

1.2 Research Objectives

This paper aims to present a new approach for the automatic generation and assessment of subjective questions by synthesizing the most promising techniques from prior years of research. Interpretability is one of the essential topics we want to focus on in this research, as it enables human users to understand and explain the system's decisions and outputs. It plays a crucial role in increasing the trust, fairness, and overall performance of the system, giving it the ability to adapt to new situations.
The objectives of our research are the following:-

- Construct a suitable dataset for subjective question generation and answer evaluation
- Instruct tune Mistral 7B, Llama-2 7B, Falcon 7B for the task of Subjective Question Generation and Answer Evaluation
- Compare the results and find the best solution

Chapter 2

Related Work

This literature review presents an overview of the current subjective question generation and evaluation research works. This review covers a range of studies conducted in the field, including theoretical and empirical research. Our focus is on using NLP to generate subjective questions and evaluate answers, as this is an under-explored area within the field. The literature review will discuss the different methods and techniques and their strengths and weaknesses. This review provides a foundation for our research, giving the current state of the field and indicating where there is scope for further study.

2.1 Question Generation

Deena et al. [11] suggests a method for automatically constructing evaluation questions to test the cognitive level of e-learners. The authors propose a system that produces subjective questions for evaluating the depth of the learner's comprehension and objective questions for assessing the learner's retention. Their model creates questions using Bloom's Taxonomy and Named Entity Recognizer. They mainly focus on generating only knowledge-level questions of Bloom's Taxonomy for the subjective questions. For the objective questions, they use Named Entity to generate distractors. The approach was evaluated through a study with teachers and students and showed strong results in terms of performance. However, it would be interesting to explore further improvements to the method, such as incorporating additional types of context or using more advanced natural language processing techniques, to see if the existing models can achieve even better performance. It would also be beneficial to explore incorporating other levels of Bloom's Taxonomy in future research to enhance the system's capabilities further.

Du et al. [7] introduces a neural network-based approach for generating questions for reading comprehension. The authors of this paper recommend employing a neural network to map the input text to the output question directly. Their approach generates questions using an RNN Encoder-Decoder architecture with attention mechanisms and LSTM, which enables the consideration of both sentence and paragraph context. They trained three models, one with only sentence-level information, one with paragraph context, and one with both sentences- and paragraph-level data. They discovered that the one with simply the sentence-level information fared the

best. The author utilized the SQuAD dataset for training the model. The authors assessed the model on various question-generating tasks and determined that it attained state-of-the-art performance in terms of BLEU score. As evaluated by human annotators, the model also developed more human-like questions than previous models. Additionally, the authors performed a user survey to assess the effectiveness of the generated questions in strengthening reading comprehension. They found that the generated questions were effective in helping readers better understand the text, as measured by an increase in the number of correct answers given by the readers.

Klein et al. [10] approaches the problem of automatic question generation by combining question generation and question answering. The authors outline a model incorporating two cutting-edge language models, GPT-2 and BERT. The authors propose a method called "Learning to Answer by Learning to Ask." by training on the SQUAD dataset. Along with BLEU and ROGUE, they used BERT QA as a surrogate measure of question generation quality. The authors test the suggested method on many benchmarks for answering questions and find that it beats GPT-2, BERT, and other state-of-the-art systems.

Mohd et al. [5] The paper delves into the attributes of an ideal survey question and introduces an algorithm for generating such questions. The authors underscore the significance of eliciting honest responses, seeking a one-dimensional reply, covering a broad range of possible answers, providing mutually exclusive choices, abstaining from assuming a specific condition, and avoiding emotionally charged or ambiguously worded inquiries. They devised an algorithm that decomposes sentences, extracts key elements, and simplifies them to form questions. The algorithm was tested using a sample set of sentences, and the results demonstrate that it generated real questions with an accuracy exceeding 90%, which can be further optimized.

Agarwal et al. [1] have developed a system for generating questions from a given text. They have analyzed the use of discourse connectives (words or phrases that link clauses or sentences) in question generation by focusing on four subordinating conjunctions and three adverbials. They have determined which connectives make for better content in question generation and found that the second argument was generally a better choice for all the connectives studied. They have also developed a system for identifying the target argument for question generation and used this to apply a series of transformations to the text, such as moving the auxiliary verb to the beginning of the sentence and adding a question word to create the final question. They then evaluated the performance of their system in generating questions, with an overall rating of 6.3 out of 8 on one dataset and 5.8 out of 8 on another.

Lamm et al. [14] talks about the QED corpus, a collection of explanations for extractive question answering (QA) that humans can understand. The corpus is based on the Natural Questions dataset and has annotations for sentence selection, referential equality, and entailment. The passage also mentions that models such as SpanBert and T5 have been used to work with the QED corpus.

Honovich et al. [12] talks about determining if a knowledge-grounded dialogue system is consistent and can give answers based on what it knows about the world.

The technique involves coming up with questions and then finding solutions. It uses a new dataset of annotated human-to-human conversations to ensure the facts are consistent. The passage also mentions a metric called Q2, which compares answer spans using natural language inference and correlates more with human judgments than other metrics.

2.2 Answer Evaluation

Dhokrat et al. [3] presents a method for assessing the quality of answers to subjective questions. Subjective questions are open-ended questions that require the use of critical thinking and analytical skills to provide a solution and are commonly used in education and testing to evaluate the understanding and knowledge of the learner. The authors propose a system that uses natural language processing techniques, including extraction and paraphrasing, and WordNet, to assess the quality of the answers to subjective questions. The system includes a pre-processing module that performs tasks such as tokenization and stemming and a feature extraction module that extracts features from the answers, such as the presence of specific keywords and the length of the answer. The system also includes a classification module that uses machine learning algorithms to classify the answers as high-quality or low-quality based on the extracted features.

Sakaguchi et al. [6] proposes a method combining response-based and reference-based approaches for evaluating short answers to reading comprehension assessments. The response-based method uses details from the student's response, while the reference-based method uses text similarity methods to compare the student's response to reference texts. The research found that integrating both approaches improves performance compared to using either technique alone.

Loukina et al. [8] looks at how we can use a mix of natural language and speech-processing techniques to judge how well people speak English on tests, and they are reviewed in terms of content and delivery (pronunciation and fluency). They evaluate the diversity of responses by repurposing components of an established NLP system developed for textual responses. On the other hand, the way the answers are given is judged using parts of a current speech-scoring system. The performance of a combined model using both types of features is compared to two baseline models using only one feature. The models' scores are compared against human raters' scores to determine how well they performed. The results show that the combined model does better than the text-only and speech-only models separately, but less than was expected. The ASR model is mentioned as being used in the study.

Hou et al. [2] researched to develop a method that uses computer systems to automatically evaluate students' answers in a classroom setting, aiming to enhance student-teacher interactions. This study built a corpus of assessments from a university course and used this corpus to extract relevant information and construct a feature model. The feature model is then applied to a classification problem, with experiments utilizing two- and three-class classification strategies. The experiments with SVM yielded encouraging outcomes, and future improvements are being con-

sidered. The study discovered that by including n-gram concepts in the feature model, the system's performance was heightened, achieving an average precision rate of 71.9%, which is a 5.6% improvement.

Madnani et al. [4] proposed a method for evaluating reading comprehension questions related to living standards was proposed in which a machine learning method is used. The system includes eight features, such as BLEU, ROUGE, and text-copying measurements, as inputs for a logistic regression classifier. The study also suggested a unique summary writing assignment as part of a cognitive method for assessing reading comprehension. In order to create an early automated scoring system, it first extracted NLP features from a holistic rubric used to evaluate student summaries. According to the study's findings, the automatic approach assessed student summaries of two distinct passages with success. Although both kinds of research aimed to assess reading comprehension, the first was based on machine learning, and the second used cognitive methodology.

Alrehily et al. [9] presents an evaluation method that compares student responses to the instructor's reference replies. Pre-processing, Keyword Expansion, Matching, and Grading are the four modules that make up the system. The Pre-processing module receives the reference and student responses, lowercases them, eliminates prepositions, stop words, and punctuation, and then generates cleaned responses. The reference answer is cleaned up and divided into a list of keywords by the Keyword Expansion module, which then creates a dictionary containing synonyms for each keyword. The matching module converts the processed sentences into vectors and evaluates them based on the cosine similarity. The grading module uses similarity to award marks to the students. The system was evaluated through surveys and comparison to traditional assessment methods, with results showing that the system gives correct results, is efficient and easy to use, and consumes fewer resources. Spearman's correlation between electronic and traditional assessment results showed a strong correlation.

2.3 LLMs

LLAMA [21] revolutionized the LLM scene by scaling the model size down and optimizing inference budget while maintaining state-of-the-arts performance. The research produced a series of models ranging from 7B-65B parameters. The authors argued that larger models did not necessarily produce the best performance. For a given budget, a smaller model trained on a large amount of data can produce better results. The authors made several modifications to the transformer architecture like using RMSNorm for prenormalization, using SwiGLU instead of ReLu for activation function, using a rotary embedding style while also optimizing the implementation. As a result, after comparison on various benchmarks, it was seen that the LLAMA-7B model outperformed GPT-3 and the LLAMA-65B model produced similar performance to the state-of-the-arts like Chinchilla and PaLM while being significantly smaller in size than the counterparts.

InstructGPT [15] uses instruct-tuning to make the model prioritize following instruction over generating the next token. The authors use RLHF to finetune the

base GPT-3 so that the model follows instruction better, is more truthful and provides less harmful responses. The authors follow an iterative reinforcement learning with human feedback to achieve the results. They hired labelers to provide desired behavior of the model. This data is then used to finetune GPT-3 using supervised learning. The labels then rank the model outputs from best to worst. This is used to train the reward model. The reward model is then used as a reward function for training model using reinforcement learning. The labelers preferred the InstructGPT output over GPT-3. The authors also used several benchmark to test models performance on truthfulness, toxicity and bias. While the model achieved a better performance on truthfulness and toxicity but not on bias. The approach's scalability and applicability, however, are crucial points of discussion. The procedure is labor-intensive and time-consuming, with a heavy reliance on human evaluators to produce comparison data and provide consistent, high-quality feedback. In addition, while this iterative feedback approach mitigates some of the risks associated with model misspecification, it is not completely immune to these risks, indicating the need for additional research to manage these potential dangers.

Self-Instruct [22] however, tries to solve the problem of human dependence by using generated instructions. A small collection of manually written tasks guides their generation. First, the model generates instructions for new tasks. This phase expands the previous set of instructions to define new tasks. The framework provides input-output instances for the newly generated instructions to supervise instruction tweaking. Finally, heuristics filter low-quality or repetitive instructions before adding valid assignments to the task pool. For repetitive instructions, the authors used ROGUE-L score for identification. This process is run iteratively until enough tasks are generated. This data is then used to finetune the model in a supervised way. The final model was on par with InstructGPT in terms of performance.

In fact, Stamford's Alpaca [20] uses the self-instruct mechanism to train their model. Alpaca is a fine-tuned version of LLAMA 7B. The authors used the same methodology of self-instruct to generate 52K instruction-following demonstrations using the OpenAI's text-davinci-003 model. This data was used to fine-tune the LLAMA 7B model. They conducted a human evaluation and found the model's performance on par with the text-davinci-003 model even though their model was signifantly smaller in size.

The authors of LIMA [23] studied the importance of two steps of LLM training. First, the pretraining step includes unsupervised learning to predict the next token. The second step is the supervised fine-tuning to align the model to specific tasks. The authors hypothized that the bulk of learning occurs in the pretraining step. Therefore, a well-pretrained model can be finetuned using a small amount of well-curated data. To prove their hypothesis, they finetuned LLAMA 65B modelusing a carefully curated dataset of 1000 examples party collected from community forums like stackexchange, wikihow and party manually created. Unlike previously discussed methods, their approach did not include RLHF which reduces the human dependence of the training process. They conducted human evaluation and compared their models with the state of the arts like Alpaca, DaVinci003, Bard, Claude and GPT-4. Their model performed better than Alpaca and DaVinci003 but fell

short in case of Bard, Claude and GPT-4. However, considering the smaller dataset, reduced time to train and smaller size of the model compared to the counterparts, the slight decrease in performance is justified.

LoRA [13] is an approach for finetuning LLMs in a computationally efficient way. As LLMs get bigger and bigger, it is computationally expensive to fully fine-tune these models. Hu et al. proposed LoRA to solve this problem. LoRA freezes the pretrained weights of the LLM and adds trainable parameters for the finetuning making it significantly efficient in terms of computation. The authors demonstrate through extensive experiments that LORA maintains or even exceeds the performance of conventional fine-tuning techniques, despite the substantial reduction in the number of parameters that are updated.

QLORA [17] improves on LORA by introducing quantization. The authors use 4-bit NormalFloat quantization, double quantization and paged optimizer using NVIDIA unified memory to reduce the memory requirement for finetuning drastically. NormalFloat is an improvement over quantile quantization. Quantile quantization operates by approximating the input tensor's quantile which is expensive while the alternatives are error prone. Instead the authors transform all weights to a single fixed distribution by scaling the weights such that the distribution fits properly. Double Quantization refers to the process of quantizing the quantization constants for additional memory savings. The authors propose further memory optimization by making use of NVIDIA unified memory paging. Following their methodology, they report a reduction in memory requirement of a 65B parameter model from 780GB to 48GB. They were able to significantly optimize finetuning without effecting the performance. Their best model achieve 99.3% performance level of ChatGPT.

Finally, A seven billion-parameter language model called "Mistral 7B" is presented in another work [18]. It performs better than current larger models such as 13B or even certain 34B ones. It performs better than them in a number of benchmarks, including arithmetic, reasoning, and code production. It makes use of sliding window attention (SWA) to effectively handle lengthy sequences and grouped-query attention (GQA) for quicker inference. The model shows that substantial results can be obtained with well-crafted language models without increasing model sizes. Additionally optimized for instructional use, Mistral 7B has tools that enforce boundaries in front-facing apps, guaranteeing the creation of moral and safe text. The study emphasizes the possibility of developing more effective, economical, and high-performing language models that can be used in a variety of real-world contexts.

Chapter 3

Data Generation

3.1 Dataset Overview

To fine-tune the LLMs for our tasks, we generated two distinct datasets:

- A dataset with 6,978 datums for the task of fine tuning LLMs for subjective question generation.
- A dataset with 4,523 datums for the task of fine tuning LLMs for answer evaluation.

3.2 Question Generation Dataset

For the question generation task, each datum was created to generate questions based on Bloom's Taxonomy, an educational framework that categorizes cognitive skills into analysis, synthesis, and evaluation. Each task contained an engaging passage no more than 1000 words long, extracted from a pool of several subjects, topic and subtopic combination such as using the subtopic "Immunological tolerance" from the subject "Immunology" of the subject "Biology". The output for each task included an analysis question, a synthesis question, and an evaluation question, each of which was grounded in the content of the provided passage. The data was then separated and expanded such that only one bloom's taxonomy type question is in one datum. So each prompts essentially generates three data. Hence, our final question generation datum structure is:

```
{
    "id": "random_and_unique_id",
    "subject": "A selected subject",
    "topic": "A topic of the subject"
    "subtopic": "A subtopic from the topic"
    "instruction": "Generate one question based on the
       passage: {input passage here}, corresponding to {
       Bloom's Taxonomy Level here} level of Bloom's
       Taxonomy. Ensure the generated question: - Is
       answerable by someone who has read the passage
```

```
        thoroughly. - Necessitate cognitive analysis skill -
        Does not surpass 200 words.",
    "question": "the question that should be generated by
        the model based on the instruction"
}
```

The "id" is a unique identifier for each task and the "subject", "topic" and "subtopic" is randomly chosen from a list of several combinations of subject, topic and subtopics. The "instruction" field provides a directive to generate questions from a given input passage that necessitates the use of analysis, synthesis, or evaluation skills, embodying the higher-order thinking skills outlined in Bloom's taxonomy. It is basically the input field which contains a challenging passage of a relevent subtopic amd a Bloom's Taxonomy level which is either analysis, synthesis, or evaluation. The output is basically the "question" field which is supposed to be generated by our model.

By creating this specialized dataset, we ensured that our LLM had the best chance of effectively learning how to generate quality, subjective questions.

3.3 Answer Evaluation Dataset

For answer evaluation, we have prepared a distinctive dataset consisting of 4,523 unique datums, specifically developed for answer evaluation. The data structure for this is also similar to question generation model. As there were several input output fields we decided to keep things seperate and went with the following structure:

```
{
    "id": "unique identifier",
    "SubjectiveQuestion": "a subjective question",
    "EvaluationCriteria": ["An array of evaluation criterias
        "],
    "studentAnswer": "The student answer",
    "answerEvaluation": "The evaluation of the student's
        answer
    "score": {
      "grammar": "A number out of 6",
      "coherence": "A number out of 2",
      "relevence": "A number out of 2"
    },
    "type": "A type or level of correctness of the answer
        among Perfect, Moderate, Average, BelowAverage,
        Imperfect",
    "subject": {
        "subject": "A selected subject",
        "topic": "A topic of the subject"
        "subtopic": "A subtopic from the topic"
    }
}
```

The "SubjectiveQuestion" field is the question which is supposed to be answered by the student. It is a question from the subject, topic and subtopic combination. Here

the "type" is the type or level of correctness of the student's answer. For simplicity we assumed the students correctness of the answers are of 5 levels. Below are the types from best to worst:

1. Perfect
2. Moderate
3. Average
4. BelowAverage
5. Imperfect

The rest of the fields are self explanatory. So, the input fields are "SubjectiveQuestion", "EvaluationCriteria", "studentAnswer", and the output fields are "answerEvaluation", "score" and "type".

3.4 Dataset Generation Process

The procedure for generating both datasets was meticulously orchestrated, adopting a uniform and methodical approach inspired by the strategies employed in creating their dataset for the Alpaca model [20] by utilizing a modified version of Instruct GPT.

We initially listed down several popular subjects such as Mathematics, Physics, Literature, and History, and others to generate content from. Next, we generated several topics for each subjects and then subtopic of each topics, to ensure the models touch a variety of knowledge areas. This also ensures that the model is not biased on a few popular topics. For each task, a subject-topic-subtopic combination was randomly chosen from a diverse range of topics. We then meticulously designed the prompts (discussed in 3.2 and 3.3), ensuring they were both challenging and captivating, necessitating a comprehensive understanding of the subject matter for successful completion. This was done to ensure a high degree of difficulty that would push the capabilities of our language models.

Once the prompts were set, we utilized the GPT-4 APIs to generate the data for us, thus automating a significant portion of the dataset creation process. However, automation was not the sole method employed. Following the automatic generation, each task was subjected to a rigorous manual review to verify the quality and consistency of the dataset, keeping the integrity of the data intact.

This symbiosis of automated data creation and human review facilitated the production of a substantial and diverse dataset without compromising its high-quality standards. The datasets thus assembled have proven exceedingly effective for the fine-tuning of our Large Language Models, empowering them to tackle the intricate tasks of generating and evaluating subjective questions.

3.4.1 Question Generation Prompt

This is the function to get the full prompt which can be used to generate the Question Generation seed tasks:

```
You are an AI, specifically a machine that I employ to generate
    data for me. This data will subsequently assist in fine-tuning
     Large Language Models.

Generate a seed task centered on ${subject.topic}, with a
    particular focus on the topic of ${subject.subTopic}. Follow
    these criteria for the task:

- **Input**: Draft a passage about ${subject.subTopic} ensuring
    it:
  - Is both engaging and informative.
  - Does not exceed 1000 words.

- **Output**: Devise three questions based on the passage, each
    corresponding to a specific level of Bloom's Taxonomy:
  - One 'analysis' level question.
  - One 'synthesis' level question.
  - One 'evaluation' level question.

Ensure each question:
  - Does not surpass 200 words.
  - Is answerable by someone who has read the passage thoroughly.

Return the seed task using the following JSON format:


{
  "input": "PASSAGE_HERE",
  "analysis": "ANALYSIS_QUESTION_HERE",
  "synthesis": "SYNTHESIS_QUESTION_HERE",
  "evaluation": "EVALUATION_QUESTION_HERE"
}
```

3.4.2 Answer Evaluation Prompt

For each student answer types listed above (1), we needed to use different prompts and hence different runs. This eliminates relying on GPT 4 APIs for randomness.

This is the full prompt used to generate "Perfect" type student answer evaluation data:

```
You are a machine that I use to generate data for me that I later
     use to fine-tune Large language models. Now, Generate a
    datum from the topic of ${subject.subTopic} from ${subject.
    topic} that includes a thought-provoking subjective question,
    an explicit breakdown of the evaluation criteria for an ideal
    answer, and a sample student response. The student response
```

```
    should be a perfect and ideal answer to the subjective
    question and the answer is relevant, coherent and contains no
    grammatical mistake.

**Here is the datum outline:**

**Generated Subjective Question**
A question that requires a deep understanding of the concepts of
    ${subject.subTopic}.

**Evaluation Criteria**
A comprehensive breakdown of what a perfect answer should
    encompass, focusing on relevance to the question, coherence of
     the argument, and correctness in grammar and spelling.

**Sample Answer**
A response that is highly relevant, coherent, and grammatically
    correct, aligning perfectly with the evaluation criteria.

**Evaluation:** Explicate why this response is the perfect answer
     according to the evaluation criteria. Emphasize the high
    degree of relevance, the coherence of the argument, and the
    correct use of grammar and spelling in the response. In the
    evaluation, emphasize the connection between the student's
    response and the established evaluation criteria. This will
    help illustrate the grading process clearly and provide a
    useful reference for understanding how subjective questions
    can be evaluated in an objective manner.

I want the response to be in the following JSON format:
{
"SubjectiveQuestion":String,
"EvaluationCriteria":String[],
"SampleAnswer":{
"studentAnswer":string,
"Answer Evaluation":string
}
}
```

Next is the prompt for type “Moderate”:

```
You are a machine that I use to generate data for me that I later
     use to fine-tune Large language models. Now, Generate a
    datum from the topic of ${subject.subTopic} from ${subject.
    topic} that includes a thought-provoking subjective question,
    an explicit breakdown of the evaluation criteria for an ideal
    answer, and a sample student response. The student response
    should be relevant to the question, but has a decent amount of
     incoherence in argument, and a moderate amount of grammar and
     spelling mistakes.

**Here is the datum outline:**
```

```
**Generated Subjective Question**
A question that requires a deep understanding of the concepts of
    ${subject.subTopic}.

**Evaluation Criteria**
A comprehensive breakdown of what a perfect answer should
    encompass, focusing on relevance to the question, coherence of
     the argument, and correctness in grammar and spelling.

**Sample Answer**
A response that is relevant to the question, moderately correct
    with few incoherent arguments and a moderate amount of grammar
     and spelling mistakes.

**Evaluation:** Critically assess the relevance, coherence, and
    grammar/spelling of the response. Highlight the relevance and
    correctness of the response to the question, but emphasize the
     moderate lack of coherence in the argument and grammar and
    spelling mistakes slightly hinder the response. Although their
     score would decrease in spelling, grammar and coherence, they
     still would get maximum in relevance

In the evaluation, emphasize the connection between the student's
     response and the established evaluation criteria. This will
    help illustrate the grading process clearly and provide a
    useful reference for understanding how subjective questions
    can be evaluated in an objective manner.

I want the response to be in the following JSON format:
{
"SubjectiveQuestion":String,
"EvaluationCriteria":String[],
"SampleAnswers":{
"studentAnswer":string,
"answerEvaluation":string
}
}
```

The following is prompt for “Average”:

```
You are a machine that I use to generate data for me that I later
     use to fine tune Large language models. Now, Generate a
    datum from the topic of ${subject.subTopic} from ${subject.
    topic} that includes a thought-provoking subjective question,
    an explicit breakdown of the evaluation criteria for an ideal
    answer, and a sample student response. The student response
    should be relevant to the question, but filled with
    incoherence in argument, and riddled with grammar and spelling
     mistakes.

**Here is the datum outline:**

**Generated Subjective Question**
```

```
A question that requires a deep understanding of the concepts of
    ${subject.subTopic}.

**Evaluation Criteria**
A comprehensive breakdown of what a perfect answer should
    encompass, focusing on relevance to the question, coherence of
     the argument, and correctness in grammar and spelling.

**Sample Answer**
A response that, while relevant to the question, is riddled with
    incoherence in the argument and exhibits grammar and spelling
    mistakes.

**Evaluation:** Critically assess the relevance, coherence, and
    grammar/spelling of the response. Highlight the relevance of
    the response to the question, but emphasize that the lack of
    coherence in the argument and numerous grammar and spelling
    mistakes greatly hinder the response. Despite the student's
    understanding of the topic at hand, their inability to
    effectively communicate their understanding significantly
    reduces their score.

In the evaluation, emphasize the connection between the student's
     response and the established evaluation criteria. This will
    help illustrate the grading process clearly and provide a
    useful reference for understanding how subjective questions
    can be evaluated in an objective manner.

I want the response to be in the following JSON format:
{
"SubjectiveQuestion":String,
"EvaluationCriteria":String[],
"SampleAnswers":{
"studentAnswer":string,
"answerEvaluation":string
}
}
```

Next, the prompt of type "BelowAverage":

```
You are an AI, specifically a machine that I employ to generate
    data for me. This data will subsequently assist in fine-tuning
     Large Language Models.

Your goal is to generate a datum from the broad and fascinating
    topic of ${subject.subTopic} in ${subject.topic}. This datum
    should encompass a thought-provoking subjective question, an
    extensive and thorough breakdown of the evaluation criteria
    for an ideal answer, and a sample student response. However,
    this student response will be different. It will reflect a
    below-average understanding of the topic, producing an answer
    that, while holding some relevance to the question, is marred
    by incoherence in argumentation, and a plethora of grammatical
```

```
    and spelling errors.

**Here is the expanded datum outline:**

**Generated Subjective Question**
Formulate a subjective question that stimulates intellectual
    curiosity and challenges a student's depth of understanding of
     ${subject.subTopic} in ${subject.topic}. This question should
     not merely test memorization, but rather the student's
    ability to analyze, synthesize, and evaluate the learned
    information.

**Evaluation Criteria**
Present an exhaustive breakdown of what an ideal answer should
    encompass for the generated subjective question. Basically it
    should contain the criteria in detail for what the perfect
    answer to the question  should look like.

**Sample Answer**
Craft a student's response that, while showing some degree of
    relevance to the question, is marred by a lack of coherence in
     argumentation and a high frequency of grammatical and
    spelling errors. This answer should be representative of a
    below-average understanding or expression of the subject
    matter. If the answer was to be scored, the student would get
    4/10.

**Evaluation:** Elaborate on the relevance, coherence, and
    grammar/spelling of the student's response. Note the relevance
     of the response to the question, but emphasize that the
    disjointed argument and numerous grammatical and spelling
    mistakes significantly detract from the answer's quality.
    Despite the student's rudimentary understanding of the topic,
    their inability to effectively express this understanding in a
     clear, coherent manner results in a low score.

In your evaluation, draw a clear connection between the student's
     response and the established evaluation criteria. This should
     illuminate the grading process and provide a useful point of
    reference for understanding how subjective questions are
    evaluated in an objective manner.

The response should be in the following JSON format:
{
"SubjectiveQuestion":String,
"EvaluationCriteria":String[],
"SampleAnswer":{
"studentAnswer":string,
"answerEvaluation":string
}
}
```

And finally, the prompt for "Imperfect" type student answer evaluation data:

```
You are an AI, specifically a machine that I employ to generate
    data for me. This data will subsequently assist in fine-tuning
     Large Language Models.

Your goal is to generate a datum from the broad and fascinating
    topic of ${subject.subTopic} in ${subject.topic}. This datum
    should encompass a thought-provoking subjective question, an
    extensive and thorough breakdown of the evaluation criteria
    for an ideal answer, and a sample student response. This
    student response, however, will differ from conventional
    samples. Instead of a flawless response, it should reflect a
    lack of understanding of the question's core concepts. The
    response will not be relevant to the asked question, it will
    lack coherence, and will be riddled with grammatical
    inaccuracies.

**Here is the expanded datum outline:**

**Generated Subjective Question**
Formulate a subjective question that stimulates intellectual
    curiosity and challenges a student's depth of understanding of
     ${subject.subTopic} in ${subject.topic}. This question should
     not merely test memorization, but rather the student's
    ability to analyze, synthesize, and evaluate the learned
    information.

**Evaluation Criteria**
Present an exhaustive breakdown of what an ideal answer should
    encompass for the generated subjective question. Basically it
    should contain the criteria in detail for what the perfect
    answer to the question  should look like.

**Sample Answer**
The answer should be wrong judging by the evaluation criteria and
     it be should be irrelevant to the question posed,
    demonstrating a fundamental misunderstanding or
    misinterpretation of the query. The answer should also be
    marred by a lack of coherence in argumentation and a high
    frequency of grammatical and spelling errors.

**Answer Evaluation:**
Meticulously dissect the student's response in relation to the
    evaluation criteria. Highlight each area where the response
    fails: its irrelevance, lack of coherence, and grammatical
    inaccuracies. The emphasis should be on providing a clear,
    objective, and thorough critique of the response,
    demonstrating how and why it falls short in meeting the
    evaluation criteria. This analysis will not only showcase the
    grading process but also provide a reference point for
    understanding how subjective questions are assessed and the
```

```
    importance of crafting well-thought-out, relevant, and
    grammatically correct responses.

The response should be in the following JSON format:
{
"SubjectiveQuestion":String,
"EvaluationCriteria":String[],
"SampleAnswer":{
"studentAnswer":string,
"answerEvaluation":string
}
}
```

Chapter 4

Methodology

This research thesis aims to delve into the fine-tuning process of a Mistral 7B, LLaMA 2 7B, Falcon 7B, which is a tool for subjective question creation and answer evaluation. To obtain the weights for the model, Hugging Face was utilized, whilst lit-gpt [16] was responsible for carrying out the instruct tuning procedure[15]. The lit-gpt library enables various LLMs with LoRA[13] and qLoRA [17] techniques.

4.1 Instruct tuning Model

Instruct tuning is a valuable approach for refining transformer models to better adhere to directions, particularly in the case of tasks that involve producing responses based on explicit instructions, such as subjective question creation and answer evaluation. By taking advantage of 52K self-instruct demonstrations, the Stanford team [20] utilized instruct tuning to fashion their Alpaca model by effectively fine-tuning LLaMA into an instruction-following model. This technique was instrumental in heightening the performance of Alpaca, producing high-quality responses which are akin to those produced by fully fine-tuned 7B parameters.

4.2 Importance of Instruct Tuning Model

- It helps the model better understand and follow instructions, leading to more accurate and relevant responses.
- It can improve the model's performance on subjective question generation and answer evaluation tasks.

4.3 LoRA (Low-Rank Adaptation)

LoRA, a Low-Rank Adaptation technique, is a cost-effective and efficient way to fine-tune Large Language Models [13]. This method can be especially beneficial for tuning an LLM due to the following reasons:

- **Reduced number of trainable parameters**: LoRA is a cutting-edge technique that involves incorporating trainable rank decomposition matrices into

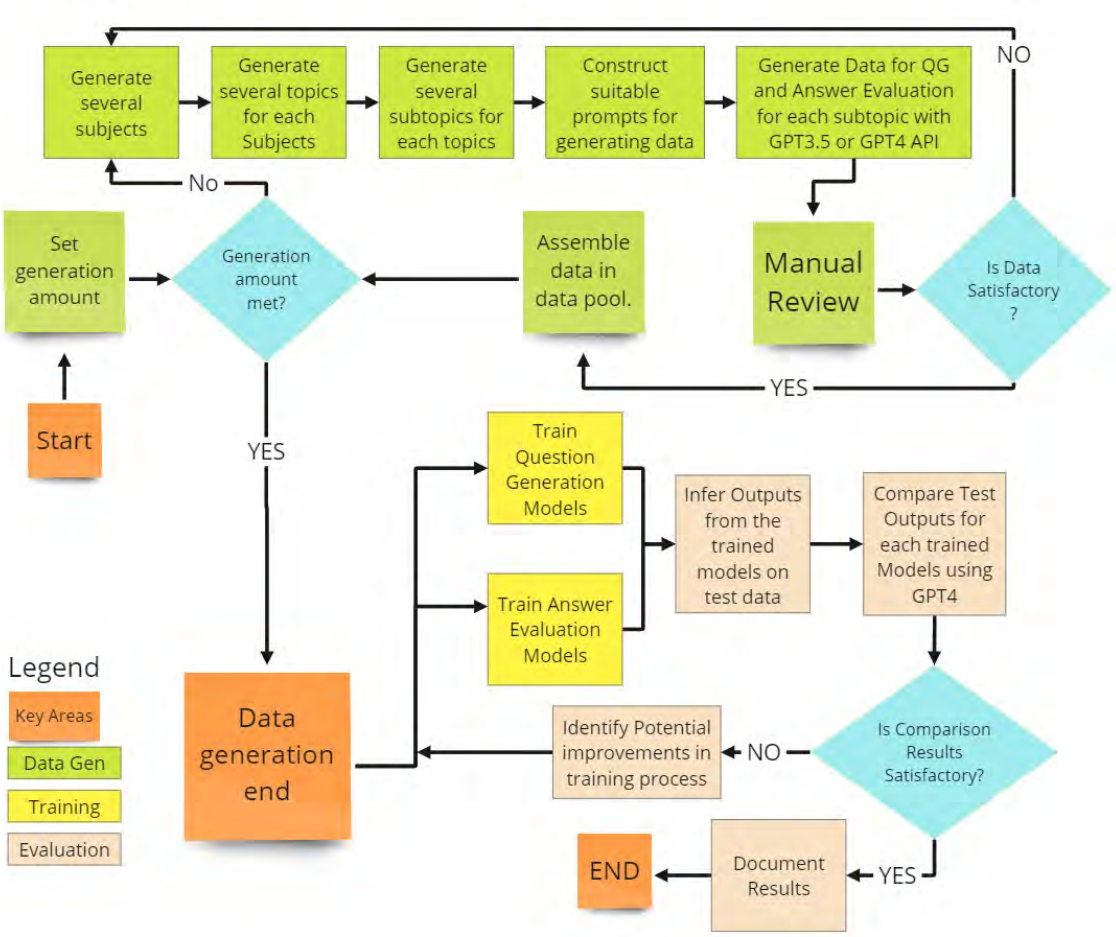


Figure 4.1: Full Workflow

each layer of the Transformer architecture. As opposed to fine-tuning the entire model, this approach focuses on reducing the number of trainable parameters. By doing so, LoRA accelerates fine-tuning and minimizes overfitting since it greatly simplifies training

- **Memory efficiency**: Low-rank approximation techniques utilized by LoRA result in a reduction of GPU memory requirements, ultimately enabling the fine-tuning of extensive models on consumer hardware despite limited available memory.
- **Task switching**: A notable benefit of utilizing LoRA is its ability to simplify task switching by replacing solely the LoRA weights rather than all parameters. This essentially produces a more inexpensive and faster mode for switching between tasks.
- **Comparable performance**: Although LoRA uses fewer trainable parameters, it performs as well or better than full fine-tuning in terms of model quality. Furthermore, unlike Adapters, it does not add any inference latency and thus stands out as a parameter-efficient fine-tuning technique. In other words, LoRA proves effective without compromising performance measures while avoiding any additional delays during the inference phase.

4.3.1 Quantized Low-Rank Adaptation (QLoRA)

QLoRA is a technique that employs a high-precision method to quantize a pre-trained model to 4 bits, followed by the integration of a compact set of learnable Low-Rank Adapter weights that are fine-tuned with the backpropagation step through the quantized weights. This approach strikes a balance between precision and computation, making it a viable method for fine-tuning large language models.

4.3.2 4-bit NormalFloat Quantization

4-bit NormalFloat (NF4) quantization is a novel encoding optimized for the distribution of neural network weights. It compresses the full pre-trained language model to reduce memory requirements. The quantized base model is then supplemented with low-rank adapters added densely throughout the layers. The adapters use full 16-bit precision and are fine-tuned while the base model remains fixed.

4.3.3 Bfloat16 Floating Point Precision

Bfloat16 is a 16-bit floating point format that reduces precision from 24 bits to 8 bits while preserving the number range of the 32-bit IEEE 754 single-precision floating-point format (binary32). This format is frequently utilized in mixed-precision arithmetic since it allows bfloat16 numbers to be expanded to larger data types and acted upon. It is particularly useful in machine learning and deep learning applications due to its balance between range and precision.

4.4 Model Training

We trained the Llama 7b, Falcon 7b, and Mistral 7b models on a single NVIDIA RTX A6000 GPU. The dataset comprised 4500 samples with an 80/15/5 training/-validation/test stratified split. The training aimed to evaluate subjective answers against a reference answer and to generate subjective questions from a given passage.

4.4.1 Trained Models

LLaMA 7B

- **Total parameters:** 7 billion
- **Trainable parameters with qLora:** 21 million
- **Total heads:** Not specified
- **Training tokens:** 1 trillion
- **Vocabulary size:** 32,000
- **Hidden size:** 4096
- **Hidden layers:** 32
- **License:** Non-commercial bespoke license
- **Model architecture:** Auto-regressive language model
- **Training period:** January 2023 to July 2023

Mistral 7B

- **Total parameters:** 7.3 billion
- **Trainable parameters with qLora:** 21 million
- **Total heads:** Not specified
- **Attention window size:** 4096
- **Language:** English
- **License:** Apache-2.0
- **Model architecture:** Uses Grouped-query Attention and Sliding Window Attention

Falcon 7B

- **Total parameters:** 7 billion

- **Trainable parameters with qLora:** 21 million
- **Total heads:** 71
- **Training tokens:** 1,500 billion
- **Context length:** 2048
- **Embedding length:** 4544
- **Feed-forward length:** 18176
- **Block count:** 32
- **License:** Apache-2.0
- **Model architecture:** Causal decoder-only model

4.4.2 GPU Specifications

- **GPU Memory:** 48 GB GDDR6
- **Memory Interface:** 384-bit
- **Memory Bandwidth:** 768 GB/s
- **RT Core Performance:** 75.6 TFLOPS
- **Tensor Performance:** 309.7 TFLOPS
- **NVLink Bandwidth:** 112.5 GB/s (bidirectional)
- **GPU Architecture:** Ampere
- **CUDA Cores:** 10,752
- **Tensor Cores:** 336
- **Process Size:** 8 nm

4.4.3 Hyperparameters

The training was configured with the following hyperparameters:

- **Learning rate:** 2×10^{-4}
- **Batch size:** 128
- **Micro batch size:** 1
- **Gradient accumulation iterations:** batch_size
- **Maximum sequence length:** 2500
- **Maximum iterations:** 10,000 (train dataset size)

- **Weight decay:** 0.01
- **Quantization:** bnb.nf4
- **Precision:** bfloat16 floating-point format
- **LoRA parameters:**
  - r**:** 8
  - α**:** 16
  - **Dropout:** 0.1
  - **Query, Key, Value, Projection, MLP, Head:** Enabled
- **Warmup steps:** 100

4.4.4 Training Settings

The training was configured with the following settings:

- **Eval interval:** 10 steps
- **Eval iterations:** 100
- **Eval max new tokens:** 350

4.4.5 Training Prompts

We used the following prompt structure to fine-tune the subjective answer evaluation models. A similar structure is followed to fine-tune the subjective question generation models, albeit with different instructions and inputs.

```
Below is an instruction that describes a task, paired with an input that
provides further context.  Write a response that appropriately completes
the request.

### Instruction:
Evaluate the answer to the following question.  Give a score in terms of
relevance, coherence, grammar and explanation of the evaluation.
Please structure your response as follows.  Begin with the 'Answer Evaluation'
section, offering an in-depth review and analysis of the student's answer
with respect to the given evaluation criteria.

Follow this with a whole number numerical score for relevance (out of 6),
coherence (out of 2), and grammar & spelling (out of 2) for the student's
answer.  Each score should be listed on a new line, preceded by its respective
category.

### Input:
Question:
What are the key principles and challenges in conducting clinical trials
```

```
for new drugs in the field of clinical pharmacology?

Evaluation Criteria:
Identification and discussion of key principles in clinical trials for
new drugs
Analysis of challenges faced in conducting clinical trials
Demonstration of understanding of the field of clinical pharmacology
Coherence and clarity of argumentation
Accuracy of information
Grammar and spelling

Answer:
Clinical pharmacology is when you study drugs and how they work in people.
There are challenges like finding enough people for the trials and making
sure the drug is safe.  The principles are like making sure the drug works
and is better than what's already out there.  It's important to test on
different kinds of people and record the results.  Sometimes, people get
side effects.
```

4.4.6 Training Process

1. The code begins by setting up hyperparameters such as learning rate, batch size, and weight decay, among others. These parameters are crucial for controlling the learning process of the model.

2. We then set up the training environment, including the data directory, checkpoint directory, and output directory. It also sets up the precision and quantization for the training process.

3. The model is loaded from a checkpoint, and only the LoRA layers are marked as trainable. This is done to fine-tune the model on a specific task.

4. The optimizer used is AdamW, which is a variant of the Adam optimizer that includes weight decay. This optimizer is known for its efficiency and effectiveness in training deep learning models.

5. A learning rate scheduler is used, specifically the CosineAnnealingLR scheduler. This scheduler adjusts the learning rate according to a cosine function, which can help achieve a better model by exploring different learning rates during training.

6. The training process involves iterating over the training data, computing the loss, and updating the model parameters using the optimizer. The loss function used is cross-entropy, a common choice for classification tasks.

7. The model is evaluated at regular intervals during training. This involves computing the loss on a validation set and generating some output from the model to check its performance.

8. Finally, the trained model is saved for future use. The saving process involves filtering out the LoRA weights, which are the only parts of the model that have been updated during training.

4.4.7 Training and Validation Loss (Answer Evaluation)

The training and validation loss for the subjective answer evaluation models were visualized as shown in Figure 4.2 and 4.3.

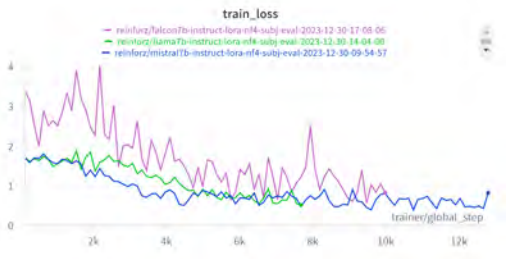


Figure 4.2: Train Loss Graph

Figure 4.3: Validation Loss Graph

4.4.8 Training and Validation Loss (Question Generation)

The training and validation loss for the subjective question generation models were visualized as shown in Figure 4.4 and 4.5.

Figure 4.4: Train Loss Graph

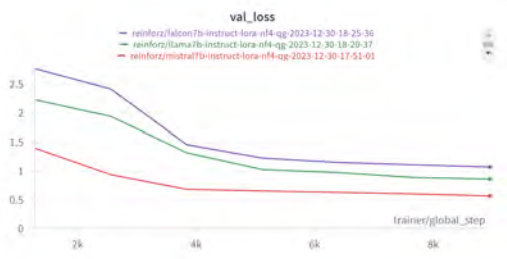


Figure 4.5: Validation Loss Graph

Chapter 5

Result Analysis

5.1 Model Evaluation Metric

For model evaluation, we used GPT-4 as a tool to rank the model outputs for each test case. Since both Subjective Question Generation and Answer evaluation are complex task requiring high level understanding of Natural Language, traditional metrics such as BLEU, ROGUE, etc. which are often based on matching the expected output is not a suitable method here. Due to the subjective nature of these tasks, there is no one predetermined correct answer.

Human evaluation could be explored as an option. However, it presents limitations in terms of cost and time. Furthermore, variations in the perspectives of to different evaluators would lead to lack of consistency in the evaluation of the models. According to [19] GPT-4's evaluation aligns with Human Evaluation. Moreover, GPT-4 will maintain a consistency and avoid human biases. As a results, we thought it suitable to use GPT-4 as an evaluation tool for our specific needs. This method not only ensured a consistent and unbiased evaluation across all test cases but also allowed us to efficiently process a large volume of data, maintaining high standards of accuracy and reliability in our findings.

Using GPT-4, we ranked the outputs of the models we trained (Mistral 7B, Llama 7B, Falcon 7B) and GPT-3.5 from 1 to 4, 1 being the best and 4 being the worst. This evaluation method was used in both Question Generation and Answer evaluation part. To ensure avoiding any bias towards GPT-3.5, we anonymize the model names. The following sections reports a detailed analysis of the results obtained for both the tasks.

5.2 Question Generation

The table 5.1 shows the aggregated results of GPT-4 rankings for Question Generation.

Mistral 7B proves to be highly effective in terms of question generation by ranking top in the most number of test examples (65.78%). It ranks in the top half almost 90% of the time. This indicates that instruct-tuned Mistral 7B can produce high

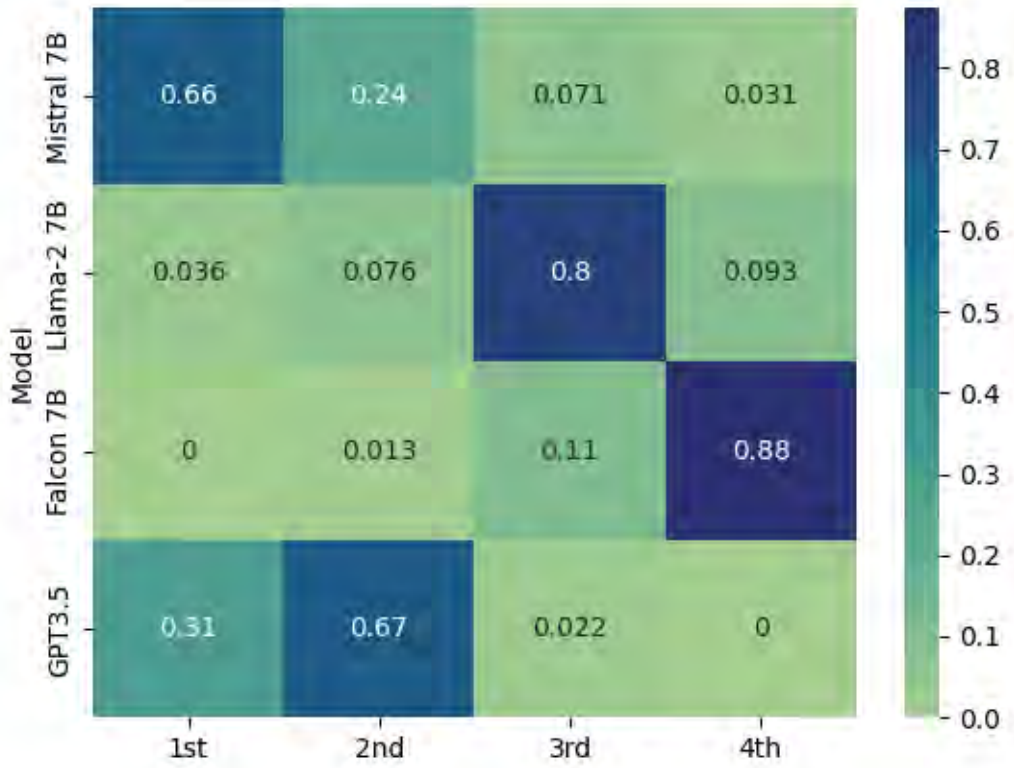


Figure 5.1: Question Generation Model Ranking Heatmap. Rows denote models and the columns denotes the ranks. Each square shows the percentage of times the model of its row placed in the rank of its column.

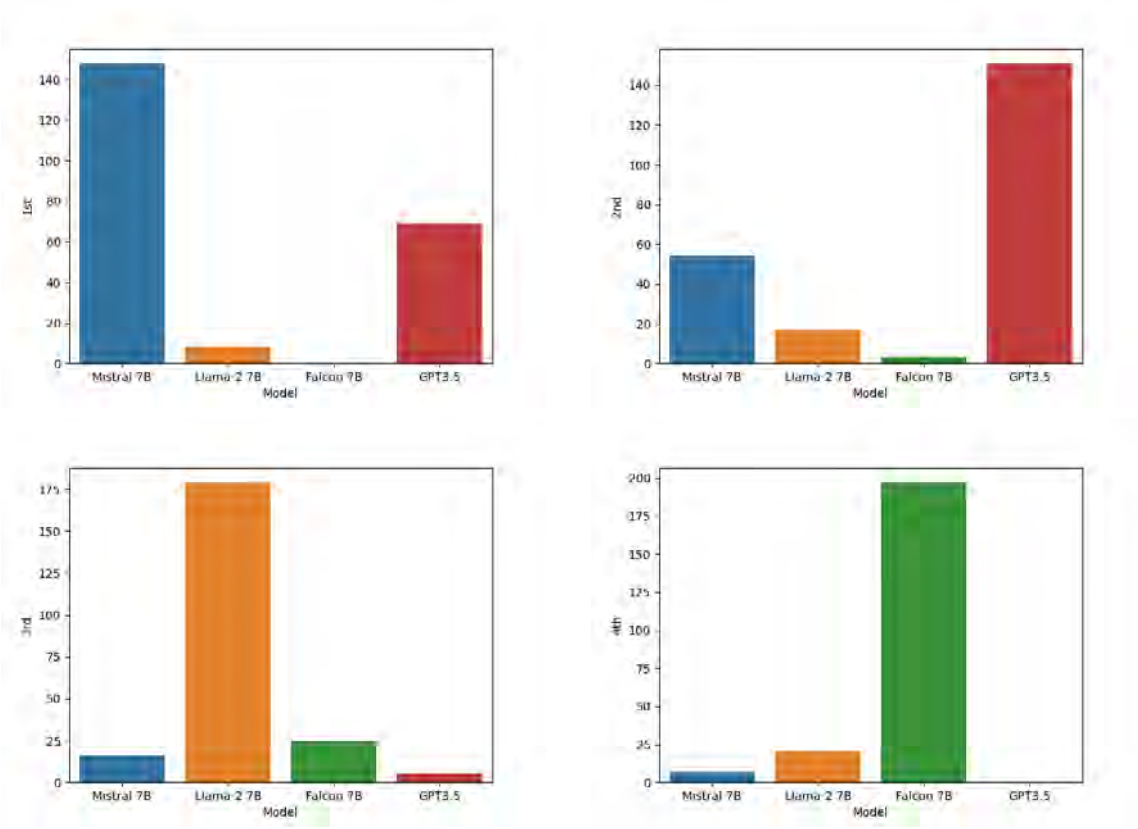


Figure 5.2: Distribution of Question Generation Model Placement for Each Rank

Figure 5.3: Distribution of Ranking for Each Question Generation Models

Table 5.1: Aggregated GPT-4 Rankings for Question Generation

Model/Rank	1st	2nd	3rd	4th
Mistral 7B	148	54	16	7
Llama-2 7B	8	17	179	21
Falcon 7B	0	3	25	197
GPT3.5	69	151	5	0

Model/Rank	1st	2nd	3rd	4th
Mistral 7B	91	108	15	11
Llama-2 7B	3	27	173	22
Falcon 7B	0	4	29	192
GPT3.5	131	86	8	0

Table 5.2: Aggregated GPT-4 Rankings for Answer Evaluation

quality subjective questions.

However, in terms of consistency, GPT-3.5 is ahead of Mistral 7B. Even though it ranks at the top less than Mistral, its generated responses are ranked 1st or 2nd almost 97% of the times. Moreover, its responses were never ranked to be last for any test prompts. This highlights the consistent ability of GPT-3.5 to generate high quality questions.

Llama-2 7B after being finetuned ranked predominantly at 3rd(79.56%). This is consistent with the observation that Llama-2 7B adds extra irrelevant details after its generated questions. It only ranks in the top half 10% of the times.

Falcon 7B demonstrated the weakest performance, predominantly ranking last (87.56%). This is aligns with the observation that it often generates gibberish or the worst responses. None of its responses ranked first and its presence in the 2nd and 3rd position is minimal at about 12%.

In conclusion, after fine-tuning, Mistral 7B demonstrates a performance comparable to GPT-3.5 in terms of Question Generation quality. However, Llama-2 7B and Falcon 7B shows poor performance. The poor performance of these model could be because of architectural differences in these models or due to the fact that they weren't optimized for these tasks. More research is needed to shed light onto this.

5.3 Answer Evaluation

The results of Answer Evaluation rankings are shown in the table 5.2.

The results show GPT-3.5 consistently outperforming other models. It comes out on top the most with 58.22%. Its consistency in its generation of answer evaluation is made more evident by the fact that it ranks in 1st or 2nd over 96% of the times and does not rank last in any test cases.

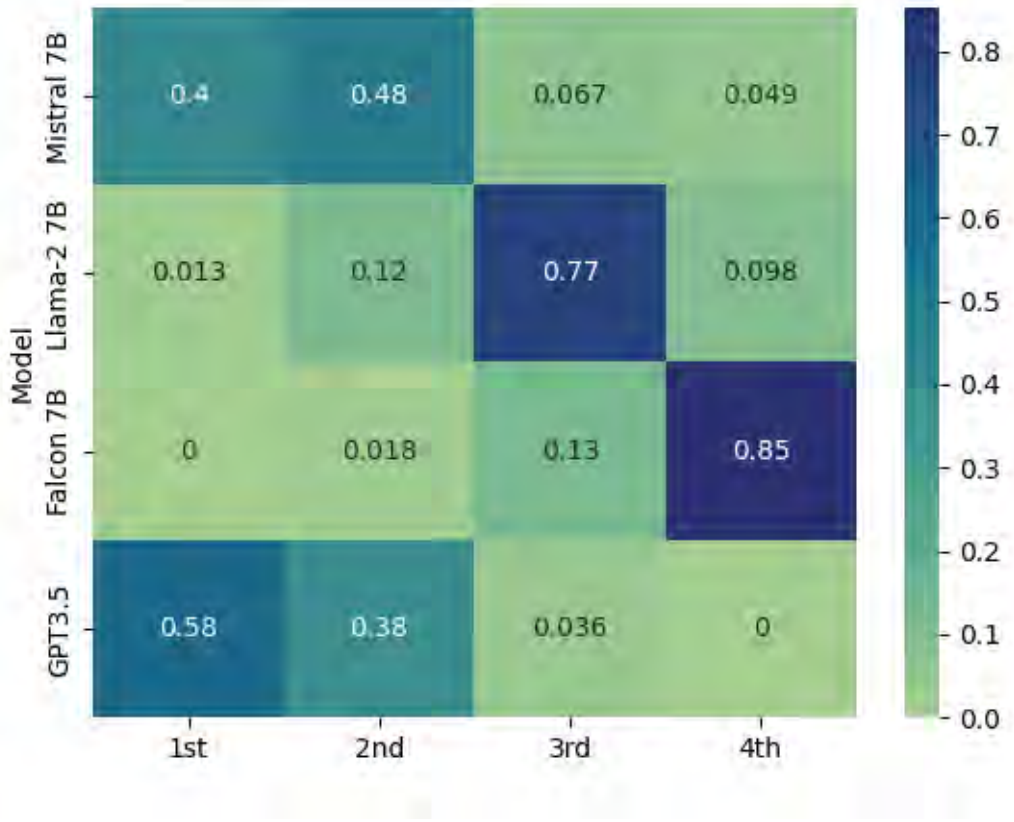


Figure 5.4: Answer Evaluation Model Ranking Heatmap. Rows denote models and the columns denotes the ranks. Each square shows the percentage of times the model of its row placed in the rank of its column.

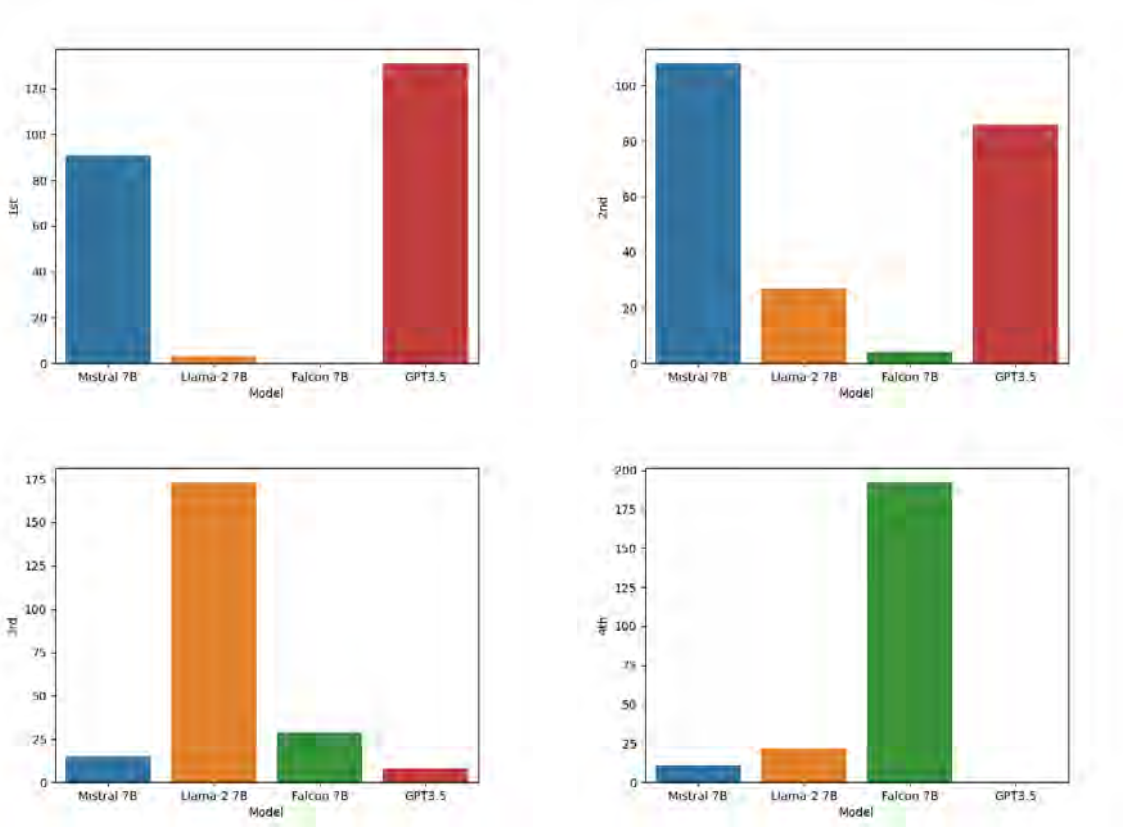


Figure 5.5: Distribution of Answer Evaluation Model Placement for Each Rank

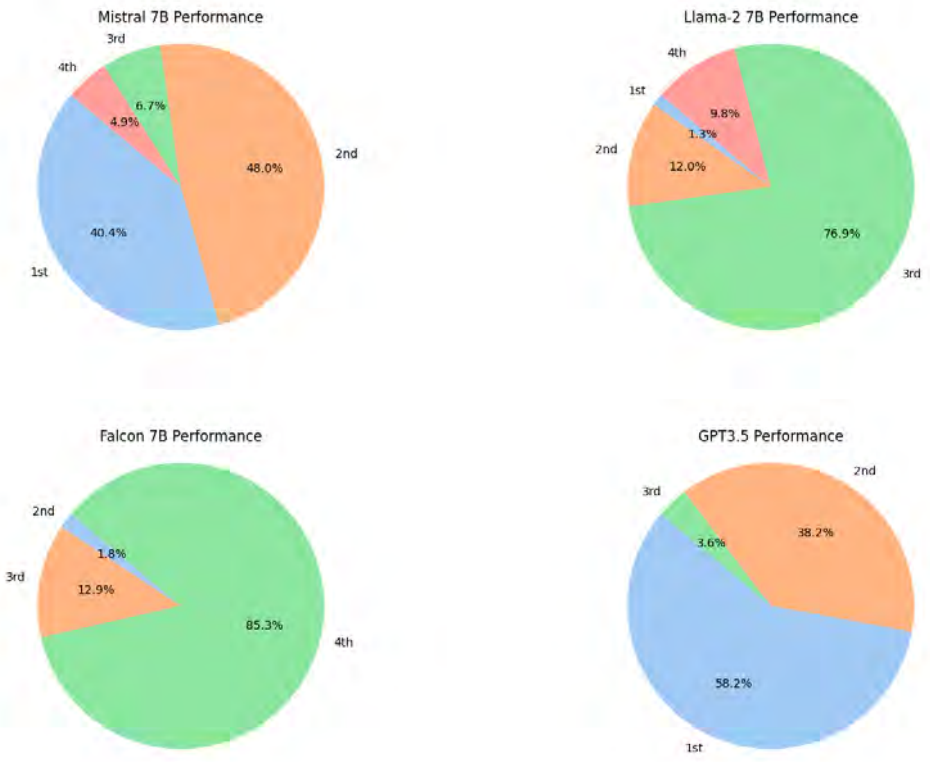


Figure 5.6: Distribution of Ranking for Each Answer Evaluation Models

Mistral 7B is closely follows GPT-3.5 in the task of answer evaluation. it ranks 1st around 40% of the times and second about 48% of the times. Overall, it is in the top half of the ranking for more than 88% of the test examples. Even though GPT-3.5 outperforms finetuned Mistral 7B, it noteworthy that Mistral 7B performs so well even though it is a far smaller model than GPT-3.5.

Similar to Question Generation, Llama-2 7B and Falcon 7B fails in Answer Evaluation tasks as well ranking predominantly in the bottom half. Llama-2 7B leads all other models by a mile in 3rd position. Its low percentage of ranking in 1st and 2nd place at around 13% signifies its inability to generate consistent and high quality evaluation for the prompted answers. Falcon 7B again comes out at last for the most part and has no top rankings. Its ranking in 2nd and 3rd position is also negligible.

In conclusion, GPT-3.5 performs the best in Answer Evaluation outperforming the finetuned models. Although Mistral is outperformed by GPT-3.5, its large amount of high ranking is promising. Since Answer Evaluation is a much complex task than Question Generation, GPT-3.5 performs better due to its large size compared to our finetuned models. In the future, larger models could be finetuned for the task of Subjective Question Answer Evaluation to test if they can perform on par with GPT-3.5.

Chapter 6

Conclusion

In conclusion, this thesis has explored the use of Natural Language Processing in education and its potential to enhance how we learn, teach, and evaluate. We have highlighted the applications of LLMs in creating personalized learning experiences, generating and assessing questions, and evaluating students' writing and speech. We constructed two suitable datasets, one for subjective question generation and another for answer evaluation (3.1). Next, we instruct tuned three LLMs for each of the tasks (4.1). Finally, we compared the results (5.1) and found out that the "Mistral 7B" perform almost at the same level as the much larger model, GPT-3.5-Turbo.

The research presented in this thesis aimed to investigate different methods for generating and evaluating subjective questions using various NLP techniques. Through the literature review, we identified a gap in the research on subjective question generation. The study demonstrates that NLP techniques can effectively generate and evaluate subjective questions, which can determine students' abilities to analyze, interpret, compare, and evaluate information. These abilities are crucial for success in various careers and ongoing learning.

The proposed methods and framework can be helpful for teachers, educators, and educational institutions in improving the assessment process and help individuals deepen their understanding of a book or article through self-assessment.

The limitations of the research are also highlighted below, and the importance of further research in this field is emphasized. Future research can look into applying other NLP techniques or exploring otherto other areas of study and improving current methodologies.

6.1 Future Work

- Exploring other newer NLP techniques instead of just relying on LLMs for subjective question generation and answer evaluation
- Using more human made data instead of synthetic ones.
- Exploring multiple other new open source LLMs and perform comparative analysis on their generated output.

- Instruct-tuning larger or better versions of current or future models for this task.
- Using multitude of evaluation metrics in order to assess the quality of the generated text. Involving humans in the loop to ensure the evaluation is as accurate as it can possibly be.
- Explore the changes in accuracy of using various other finetuning or future instruct-tuning approaches.

Overall, this research contributed insights in the field of NLP by investigating subjective question generation, which is an under-explored area, and proposing methodologies for the same. It also aims to provide a framework for educators, teachers, and educational institutes to improve the assessments and the learners' self-assessment process.

Bibliography

[1] M. Agarwal, R. Shah, and P. Mannem, "Automatic question generation using discourse cues," Jun. 2011, pp. 1–9.

[2] W.-J. Hou and J.-H. Tsao, "Automatic assessment of students' free-text answers with different levels.," *International Journal on Artificial Intelligence Tools*, vol. 20, pp. 327–347, Apr. 2011. DOI: 10.1142/S0218213011000188.

[3] A. Dhokrat, G. Hanumant, and C. Mahender, "Assessment of answers: Online subjective examination," Dec. 2012, pp. 47–56.

[4] N. Madnani, J. Burstein, J. Sabatini, and T. O'Reilly, "Automated scoring of a summary-writing task designed to measure reading comprehension," in *BEA@NAACL-HLT*, 2013.

[5] H. Mohd, M. S. Husain, and M. A. Shaun, "Automatic question generation from text," *International Journal of Innovative Research in Science Engineering and Technology*, vol. 10, pp. 10 080–10 087, Apr. 2015.

[6] K. Sakaguchi, M. Heilman, and N. Madnani, "Effective feature integration for automated short answer scoring," Jan. 2015, pp. 1049–1054. DOI: 10.3115/v1/N15-1111.

[7] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," Jan. 2017, pp. 1342–1352. DOI: 10.18653/v1/P17-1123.

[8] A. Loukina, N. Madnani, and A. Cahill, "Speech- and text-driven features for automated scoring of english speaking tasks," Jan. 2017, pp. 67–77. DOI: 10.18653/v1/W17-4609.

[9] A. Alrehily, M. Siddiqui, and S. Buhari, "Intelligent electronic assessment for subjective exams," May 2018, pp. 47–63. DOI: 10.5121/csit.2018.80804.

[10] T. Klein and M. Nabi, "Learning to answer by learning to ask: Getting the best of GPT-2 and BERT worlds," *CoRR*, vol. abs/1911.02365, 2019. arXiv: 1911.02365. [Online]. Available: http://arxiv.org/abs/1911.02365.

[11] D. G., R. Kothandaraman, N. Banu P K, and D. K. Kaliyan, "Developing the assessment questions automatically to determine the cognitive level of the e-learner using nlp techniques," *International Journal of Service Science, Management, Engineering, and Technology*, vol. 11, pp. 95–110, Apr. 2020. DOI: 10.4018/IJSSMET.2020040106.

[12] O. Honovich, L. Choshen, R. Aharoni, E. Neeman, I. Szpektor, and O. Abend, "Q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering," 2021.

[13] E. J. Hu, Y. Shen, P. Wallis, *et al.*, *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL].

[14] M. Lamm, J. Palomaki, C. Alberti, *et al.*, "Qed: A linguistically principled framework for explainable question answering," *TACL*, 2021. [Online]. Available: https://aclanthology.org/2021.tacl-1.48/.

[15] L. Ouyang, J. Wu, X. Jiang, *et al.*, *Training language models to follow instructions with human feedback*, 2022. arXiv: 2203.02155 [cs.CL].

[16] L. AI, *Lit-gpt*, https://github.com/Lightning-AI/lit-gpt, 2023.

[17] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, *Qlora: Efficient finetuning of quantized llms*, 2023. arXiv: 2305.14314 [cs.LG].

[18] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, *Mistral 7b*, 2023. arXiv: 2310.06825 [cs.CL].

[19] A. Sottana, B. Liang, K. Zou, and Z. Yuan, *Evaluation metrics in the era of gpt-4: Reliably evaluating large language models on sequence to sequence tasks*, 2023. arXiv: 2310.13800 [cs.CL].

[20] R. Taori, I. Gulrajani, T. Zhang, *et al.*, *Stanford alpaca: An instruction-following llama model*, https://github.com/tatsu-lab/stanford_alpaca, 2023.

[21] H. Touvron, T. Lavril, G. Izacard, *et al.*, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL].

[22] Y. Wang, Y. Kordi, S. Mishra, *et al.*, *Self-instruct: Aligning language models with self-generated instructions*, 2023. arXiv: 2212.10560 [cs.CL].

[23] C. Zhou, P. Liu, P. Xu, *et al.*, *Lima: Less is more for alignment*, 2023. arXiv: 2305.11206 [cs.CL].