Explainable AI For Automatic Scoring Using Large Language Models

by

Padmaja Pravin Saraf

(Under the Direction of Ninghao Liu)

Abstract

Automatic scoring has always been a point of interest in educational settings. With the recent advancements in Large Language Model this work emphasize on automatic scoring strategies with Large Language models by providing the explainability to the automatic scoring task. This thesis explores different open-source large language models and their capabilities to perform automatic scoring. Since the automatic scoring is directly rated to students and their evaluation; the explainability, reliability and trustworthiness of automatically generated scores is crucial. This research is an attempt provide explainability to these LLM generated scores by aligning LLMs with human grading strategies with attempt to improve the automatic scoring accuracy. Finally, this thesis also presented the implementation of system "GPTest", an end-to end system to perform the automatic scoring using LLM and used in educational setting by teachers and students to streamline the scoring and assessment procedure.

Index words: Automatic Scoring, Large Language Model, Prompt Engineering, Chain of Thought

Explainable AI For Automatic Scoring Using Large Language Models

by

Padmaja Pravin Saraf

B.E, University Of Pune, 2019

A Thesis Submitted to the Graduate Faculty of the

University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Master of Science

Athens, Georgia

2024

Explainable AI For Automatic Scoring Using Large Language Models

by

Padmaja Pravin Saraf

Major Professor: Ninghao Liu

Committee: Xiaoming Zhai

Ismailcem Budak Arpinar

Electronic Version Approved:

Ron Walcott

Dean of the Graduate School

The University of Georgia

August 2024

DEDICATION

To my Papa and Aai, my family, my teachers, my friends, and loved ones for their unwavering support throughout my journey.

ACKNOWLEDGEMENTS

I am extremely grateful to all my advisors, Dr. Ninghao Liu, Dr. Xiaoming Zhai, and Dr. Ismailcem B. Arpinar, for their continuous guidance and support in my research work. Their expertise in the subject helped me learn a great deal and motivated me to advance in my research. Their guidance has been a valuable part of my master's degree and will undoubtedly assist me in pursuing my career.

I would also like to thank my parents for their unwavering support throughout my master's degree which kept me moving to achieve my goals. Additionally, I am grateful to the *School of Computing* and *The University of Georgia* for all the learning experiences I have had during the journey.

This work was funded by the National Science Foundation(NSF) (Award Nos. 2101104, 2138854, PI Zhai). The findings, conclusions, or opinions herein represent the views of the authors and do not necessarily represent the views of personnel affiliated with the National Science Foundation.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	**v**
LIST OF TABLES	**vii**
LIST OF FIGURES	**viii**
1 INTRODUCTION	**1**
2 LITERATURE REVIEW	**4**
3 CONCEPTS	**6**
4 METHODOLOGY	**10**
5 RESULTS	**18**
6 SYSTEM DESIGN	**23**
7 LIMITATIONS AND FUTURE WORK	**34**
8 CONCLUSION	**35**
BIBLIOGRAPHY	**36**

LIST OF TABLES

1 Dataset item with holistic and analytic rubric. 11

2 Student response with score. 12

3 Automatic scoring matrix for GPT-4. 13

4 Automatic scoring matrix for Falcon-7B-Instruct. 13

5 Automatic scoring matrix for Vicuna. 14

6 Analytical Rubrics Vs Human Rubrics Breakdown. 18

7 Analytical rubric generation task under different settings. 20

8 Automatic scoring task under different settings of analytical rubric generation. 21

9 Technology stack used for development of GPTest. 33

LIST OF FIGURES

6.1 GPTest automatic scoring architecture diagram. 26
6.2 Class creation 30
6.3 Adding student to class 30
6.4 Assessment creation 31
6.5 Assessment list 31
6.6 Assessment email 32
6.7 AI response 32

CHAPTER 1

INTRODUCTION

The field of Artificial intelligence has transformed the world, and specially the rise of Large language models(LLMs) has significantly impacted various domains of society. Education sector is definitely no exception. Scoring assessment is a crucial task in the education domain that demands substantial time and efforts. There has always been a need for an approach to automate scoring procedures. Automatic scoring enables students to receive the assessment scores and respective feedback in timely manner. It is an important step in the science education [20]. With the rise of large language model(LLMs),automatic scoring became feasible and several studies have been conducted to analyse whether LLMs are capable of performing automatic scoring [9]. It has been observed that LLMs are capable of performing the automatic scoring task, and their integration with of Chain-of-Thought methods can enhance the accuracy of automatic scoring systems [9]. LLMs have demonstrated the capability to perform automatic scoring task with high accuracy. However, the reliability and explainability of these LLM generated scores have always been an point of interest. Since, automatic scoring is a sensitive task and directly connected with the assessment and evaluation of students, educationalists find it extremely important to understand the basis on which LLM performs the scoring tasks and how it arrive at the final score. It is still very unclear whether LLM grades the student responses in a same way as humans do. This lack of clarity introduces potential risks in automatic scoring procedure. To achieve this clarity in the automatic scoring task, we need to ensure the explainability in automatic scoring procedures.

To address this research gap, this study attempts to provide the explainability to the automatic scoring task. Integrating LLMs, such as Generative Pre-trained Transformer- 4(GPT-4), with Chain-Of-Thought(CoT) [7, 17] mechanism definitely provides a form of reasoning for the automatically generated

scores [9]. However, it only includes the holistic rubrics, keeping the rubrics breakdown abstract to user. It is very interesting to know if machine's rubrics breakdown is same as human's rubric breakdown. This allows us to analyse the reliability of the scores automatically generated by machine. In this study, we compared machine-generated rubrics breakdown with human-generated rubrics breakdown to analyse their similarity. If machine breakdown rubrics and same as that of human breakdown rubrics, then we can conclude that the score achieved by auto-scoring mechanism are reliability and trustworthy.

This study addresses the following research questions (RQs):

RQ 1 : Is LLM capable of performing the automatic scoring and which LLM models yield better performance.

RQ 2 : Humans grade according to the set of rules; check whether LLMs can provide the analytical rubrics breakdown.

RQ 3 : What is the semantic similarity between human breakdown rubrics and Machine breakdown rubrics?

RQ 4: How can we improve the semantic similarity between human breakdown rubrics and Machine breakdown rubrics and ultimately enhance automatic scoring performance?

While there are not many studies that answers these questions, our research indicates that LLMs are capable of performing automatic scoring task. The accuracy of scores generated by LLMs varies depending on the specific LLMs and prompt engineering strategies employed. We also examined the automatic scoring for open-source LLMs and found that open-source LLMs like Vicuna can yield promising results with appropriate prompting strategies. Some prompt engineering strategies such as Chain-of-Thoughts[4] can improve the accuracy of the scores and provide explanability to automatically generated scores. However, these comparisons between LLM generated results and human provided grades with categorical values do not actually test the "thought-process" of the LLM in achieving those score. This means even if the scores generated by LLMs match those of humans, it does not necessarily imply that LLM have followed the same thought process as humans. Consequently, gap may still exist between human generated scores and LLM generated scores .

To address this challenge, this thesis work further focuses on analytical rubrics designed by humans to analyse the behaviour of LLM while performing the automatic scoring task. While grading, human graders follow set of rules known as "analytical rubrics". They grade student responses by checking if each response satisfies the criteria outlined in these analytical rubrics. If student response satisfies more of these rules from the analytical rubric, the total score of the student increases, following a logical process. Along with the automatic scoring tasks, this research examines the differences between human generated analytical rubrics and LLM generated analytical rubrics, thereby identifying the discrepancies between human scoring and LLM scoring procedures.

This thesis work involves the 12 different assessment items from the science education domain, primarily covering physics topics. It also includes the student responses to these assessment items. As per the results, there are notable differences between the LLM generated analytical rubrics and the human generated rubrics, thus highlighting the alignment gap between LLM-generated scores and human-generated scores. It is observed that providing graded response as an example to LLM does not fill the alignment gap between human grading and LLM grading. However, holistic rubric helps to mitigate this gap. Additionally, providing graded student responses does not teach LLMs to perform the grading but rather it enables LLMs to find shortcuts while performing the automatic scoring task.This work also demonstrate that high-quality analytic rubrics can help in improving the performance of automatic scoring task. Finally, it presents "GPTest", an end-to-end system which performs automatic scoring using LLM and can be used in educational setting by teachers and students to streamline the scoring and assessment procedure .

CHAPTER 2

LITERATURE REVIEW

The "Attention is All You Need" paper, published in 2017 introduced a Transformer model, which opened new avenues in natural language processing and revolutionised the field of computer science [2]. This model relies solely on the attention mechanism and has significantly improved the performance of natural language processing tasks. Recent advancement in the field of transformers, particularly language models, have opened new avenues for automatic scoring tasks in the educational field. Automatic scoring can provide timely, consistent and personalised feedback to students, as compared to the conventional grading process [15]. Several studies have been conducted to analyse whether the encoder-only transformers such as BERT, and decoder-only transformers like Generative pre-trained transformer-4 (GPT-4) can perform the automatic scoring task in the educational setting [9]. Recent studies have demonstrated a zero-shot approach to automatically score student responses through Matching Exemplars as Next Sentence Prediction (MeNSP) with pre-trained language models such as BERT [19]. This approach provides a solution for the automatic scoring with reduced the cost of model training. However, this method is suitable for mainly low-stakes classroom assessments, and explanability of such automatic scoring procedure is still a concern to educationalists. One approach is fine-tuning Large Language Models(LLMs) with large number of student responses. After fine-tuning, LLM can generate assessment scores automatically. While this approach can provide high-quality feedback to student responses, it requires huge amount of assessments and student response data, making it very time consuming to fine-tune an LLM. Additionally, this approach is not suitable when there is insufficient data available to fine-tune the model [8, 13].

Researchers have also explored the possibility of automatic scoring with pre-trained LLMs without fine tuning the model, investigating whether scoring can be performed as pre-training task [9, 16]. Re-

searchers employed prompt engineering techniques to perform auto-scoring using LLMs without fine-tuning the models. This approach eliminates the need for fine-tuning the LLM and mitigates the concern of scarcity about data. Some studies have also explored LLM such as GPT-4 for performing automatic scoring in educational setting by using various prompt engineering strategies like zero-shot, few-shot and Chain-Of-thought prompting [9, 17]. These studies have observed that that LLMs are capable of performing the automatic scoring task and it's integration with of Chain-of-Thought methods can enhance the accuracy of Automatic scoring systems. The same study also indicated that few-shot learning yields a better results for automatic scoring student responses in science education. However, the high cost and limited availability of GPT-4 pose challenge for widespread use in automatic scoring performed using GPT-4. As a result, users might find it uneconomical to rely only on LLMs like GPT-4 for automatic scoring task. The performance of automatic scoring using other open-source LLMs remains largely unexplored.

Even though LLMs can perform automatic scoring task with high accuracy, the reliability and explainability of scores generated by LLMs have always been a point of interest. It still remains unexplored whether LLMs follow the same thought process as humans while grading student responses. Previous studies have not included alignment of these automatic results with human grading, which raises questions about the reliability of automatic scoring tasks.

Since automatic scoring is a sensitive task directly linked to the assessment and evaluation of students, educators find it extremely important to understand the basis on which LLM perform scoring tasks and derives final score. To achieve reliability in automatic scoring task, it becomes vital to have the explainability in this automatic scoring procedures. To address this research gap, this study is an attempt to provide the explainability to automatic scoring task.

CHAPTER 3

CONCEPTS

3.0.1 Transformers

The Transformer model is based on a Attention mechanism [2]. Introduced by Google researchers in a seminal paper in 2017, the Transformer revolutionized natural language processing. Before the introduction of the Transformer, earlier models represented words as vectors, but these vectors lacked the context. As we know the word usage differed based on the context of it. Same words with different context had same vector representation before the attention mechanism came into picture. Transformer model, an encoder-decoder architecture, addresses this limitation with it's attention mechanism. This mechanism allows the model to process vast amounts of data efficiently by leveraging GPU/TPU parallelization.

The transformer architecture primarily consists of an encoder and a decoder. The encoder processes the input sequence and passes it's encoded representation to the decoder, which then decodes this representation to perform the relevant task. According to the "Attention is all you need " paper [2], the Transformer model's encoder consists of 6 layers. Each encoder layer comprises two sub-layers: a self-attention layer and a feed-forward layer. The input first goes through the self attention layer of, where encoder identifies the relevant parts of the words. The output of the self-attention layer is then forwarded to feed-forward layer. The word passes through the self attention layer and here dependencies exists in this sub layer. However, in the feed-forward layer, words are processed independently, allowing different paths to be executed in parallel. This behaviour allows the Transformer to process vast amounts of data and leverage of parallelisation of GPUs/TPUs. The output of the encoder is then forwarded to the decoder which consists of 3 sub-layers. First one is self attention layer, the second sub layer is encoder -decoder attention layer and the third is feed forward layer. Encoder-decoder attention layer allows the decoder

primarily focus on relevant part of the input provided. In the self-attention layer, each embedding is divided into three components: Query Vector, Key Vector, and Value Vector. These components are computed using learned weights, which are adjusted during the training process of the Transformer. The computations are matrix operations.After computing these vectors, we calculate their Softmax scores. This step allows the model to focus more on relevant words and less on irrelevant ones. Next, we multiply each Value Vector by its corresponding Softmax score and then sum them up, producing the output of the self-attention layer. This output is then passed on to the next layers of the Transformer.There are different types of transformers.

Encoder and Decoder Transformers

In this architecture, Transformers use both encoder and decoder component of the transformer architecture. These Transformers are good at analysing the text and can generate text to some extent. However, they are not as proficient as decoder-only transformers when it comes to text generation. The well-known example of encoder-decoder transformers is BART [11].

Encoder Only Transformers

These type of Transformers use only encoder component of the transformer architecture. Encoder-only transformers accepts the input, perform analysis and interpretation of the text. However, they do not involve in the text generation task. The most popular example of the encoder-only transformer is Bidirectional Encoder Representations from Transformers (BERT) [6].

Decoder only Transformers

Decoder-only Transformers uses only the decoder component of the Transformer architecture. Decoder only transformer primarily performs the text generation task and have capability to generate logical and relevant text. When given an input token, these models are trained to predict the next token in the sequence. Famous examples includes Large Language Model like Generative Pre-trained Transformers

series (GPTs) [18], Mistral [1], Falcon [3], Vicuna, LaMMA [5] etc. Decoder-only transformers have experienced significant advancement and gained huge popularity in recent years. In this work, we have mainly focused on decoder-only transformers such as the Large Language Models.

3.0.2 Prompt

Prompt is a way of querying large language model(LLM) in the form of text that can be interpreted and processed by LLM to get the desired results. Prompt engineering involves designing a set of instructions for large language model in form of text in order to obtain the best possible results from the large language model. It involves designing the prompt function that generate most effective output or performance for downstream tasks [14]. There are different prompt learning techniques, they are discussed below.

Zero-Shot prompting

LLMs are trained to follow instructions and they are trained on huge amount of data. In the zero -shot prompting, models are provided with prompts without any sample examples [10]. In this technique, prompting instructions are directly provided to the base model (without further feeding any additional data to the model). The results are then observed based solely on the data on which the model was originally trained. In zero-shot prompting, no sample text is provided to the model to perform the task. It is assumed that the model already understands the instructions and is capable of providing results based on its training data.

One-Shot Prompting

In one-shot prompting, the LLM is given one sample example along with the instructions. The model then uses this example to understand and generate the text accordingly [10].

Few-Shot Prompting

Even though zero-shot and one-shot promoting yield great results, they often fall short for more complex tasks. In the complex scenarios where model needs multiple sample examples to understand the given context and perform the task accordingly, few-shot prompting comes into play. In this approach, LLMs are given more than more example of tasks to be performed, along with the expected results [4]. With the help of these examples and labels, the LLM learns the task and expected outcomes. In the complex scenarios, few-shot prompting can perform better than one-shot or zero-Shot setting.

Chain-of-thought (CoT) Prompting

Chain-of-thought is series of reasoning steps that helps LLMs improve their reasoning ability in complex scenarios[4]. Chain-of-thought is a promoting technique where chain of examples is provided to LLM in a prompt in order to perform correctly in complex scenarios [7]. This prompting technique is mainly for tasks involving complex arithmetic, conman sense and logical reasoning. In this approach, the LLM is given more set of instructions by providing examples in prompts. We have used this prompting techniques in-order to instruct model while performing automatic scoring task. Assessment question, scoring rubrics, students responses and the expected scores are also provided to LLMs to form a coherent chain.

3.0.3 In-context learning

In-context learning prompting technique where context is provided in natural language as a part of a prompt, and model is expected to provide the response [12]. In in-context learning, LLM is expected to respond to a new task for which it has not been specifically trained. The context of the task is provided to the model in form of natural language demonstration and model learns from the context to generate an appropriate response.

CHAPTER 4

METHODOLOGY

4.0.1 Dataset

This study includes an analysis of a dataset that is collected by asking middle school students to describe scientific models accounting for science phenomena [21]. In this study we considered total 7 assessment tasks, which contain questions, rubrics and student responses associated with each assessment. The dataset contains 7 assessment tasks, out of which 5 tasks have trinomial scoring rubrics and the remaining 2 tasks have binomial scoring rubrics. A binomial scoring rubric means that the assessment task has 2 different aspects of scoring, while trinomial scoring rubric means that the assessment task has 3 different aspects of scoring rubrics. The example of the assessment task is: "Develop a model that includes a particle view of matter to predict how the average kinetic energy and the temperature of a substance changes when thermal energy is transferred to or from a sample." Each assessment task may consists of multiple assessment items, some of which require students to provide responses in textual format, while others may involve drawing figures. By excluding assessment items which require the visual student responses, We then only selected 12 assessment items that specifically required student responses in textual format.

We organised the original dataset to include fields such as "Question", "Rubric Number", "Holistic Rubrics", "Analytical Rubrics" and "Special Weight". After reorganising the dataset we finally achieved 12 assessment items. Each of these 12 assessment items contain question description (background and question), rubrics (analytical rubrics and holistic rubrics) and special weight. Following Table 1 demonstrates the organised assessment item.

Here, holistic rubrics evaluate the student response as a whole, without breaking the assessment into different specific rules. It considers the overall quality of student response and then assign a grade. In

contrast, analytic rubrics divide the assessment into multiple criteria, each evaluated separately. It provides detailed criteria based on which the assessment is graded. Each of the criteria has its own rating scale. In the end, all the criteria scores are summed up to arrive at the total score or grade of the student response. For each assessment task, we collected around 800 student responses [21]. Each student response is annotated by humans according to the rubrics categorising them into different levels, such as "Beginning", "Developing", or "Proficient". We randomly selected 100 labeled student responses from each assessment item, ensuring balanced grading levels, to test automatic scoring performance of LLM graders. Following Table 2 demonstrates one of the sample student response along with the score.

Table 1: Dataset item with holistic and analytic rubric.

Rubric Number	**Q1**	**Holistic Rubric**	**Analytic Rubric**	**Special Weight**
Rubric 42-2	This task is measuring a student's proficiency on the following: Develop a model that explains how particle motion changes when thermal energy is transferred to or from a substance without changing state. Shwan had 3 dishes of water at room temperature . She cooled one dish , causing thermal energy to transfer from that dish to surrounding . She kept the middle dish at room temperature . She transferred thermal energy into the third dish by heating it. Then Shwan dropped a red -coated chocolate candy into each dish .Construct a model that shows what is happening to water particles and red dye particles in each dish . Be sure your models includes pictures and a key.(Use a model to describe how the transfer of thermal energy affects particle motion and/or temperature.)	[Grade 2 Level]: Student fully describes how the model shows that when thermal energy is transferred to the water, water and dye particles will move faster. [Grade 1 Level]:Student does not describe how the model shows that when thermal energy is transferred to the water, water and dye particles will move faster. [Grade 0 Level]: Student writes nothing or only writes some randomly words.	[Rule 1]: When thermal energy is transferred to the water (hotter), water and dye particles move faster .|||[Rule 2]: At the higher temperature, water and dye particles move faster.|||[Rule 3]: The answer is a meaningful sentence.	Answer any rule get 1 point, answer all rules get 3 points, otherwise 0 point.

4.0.2 Implementation

To answer our first research question if large language models can perform the automatic scoring we have developed a pipeline to perform the automatic scoring of the student responses which are based on the science phenomena. Given an "Assessment Task" and "Rubrics", we performed experiments with different large language models like GPT-4, Vicuna and Falcon-7B-Instruct to perform the automatic scoring

Table 2: Student response with score.

StudentID	Answer	NewScore
4436	After the shower, the water vapor is transferring to the mirror and it is warm for the heat of the shower when hitting the cold mirror. As always, since the molecules are warmer they are more spread apart and are moving faster. When the molecules are moving and in motion there is kinetic energy. The kinetic energy is on the mirror. When there is heat and the heat is warming up that is thermal energy in the shower. The lines going from the shower to the mirror are indicating the movement of the water vapor.	Beginning

of student responses. In this thesis work, we have considered "Applying Large Language Models and Chain-of-Thought for Automatic Scoring" as our base paper[1]. Since the work in the mentioned paper successfully demonstrated automatic scoring task with the GPT-4, it becomes very interesting to see how other LLMs, especially open source LLMs can perform the automatic scoring task with different prompt engineering techniques. While performing the automatic scoring we used various prompt engineering techniques to compare their performance during while scoring the responses.

We used prompt engineering techniques such as zero-shot prompt without Chain Of Thought (CoT), zero-shot prompt with Chain Of Thought (CoT) and combination of few-shot prompt with Chain Of Thought (CoT) prompts. In the zero-shot learning, LLM is not provided with any external examples of the student responses, we just provide LLM with "Assessment Task" and the "Rubrics". In contrast, in case of few-shot learning we provided four sample examples from the student responses to the LLM. We performed experiments with mainly 3 approaches, zero-shot without chain of thought prompts, zero-shot with chain of thought and few-shot with chain of thought. The experiments are conducted with GPT-4, Falcon-7B-instruct and Vicuna model which is open source LLMs. Based on the experiments we observed that , when provided with the "Assessment Task" and "Rubrics" LLMs are able to perform the scoring task successfully. However the accuracy of the scoring varies significantly based on different LLMs.

Table 3: Automatic scoring matrix for GPT-4.

Model	Task	Method	Accuracy	Precision	Recall	F1
GPT-4	Task 42	makePrompt_ZS_noCoT	0.50	0.50	0.50	0.48
GPT-4	Task 42	makePrompt_ZS_CoT	0.66	0.62	0.62	0.62
GPT-4	Task 42	makePrompt_FS_CoT	0.66	0.75	0.75	0.66

Table 4: Automatic scoring matrix for Falcon-7B-Instruct.

Model	Task	Method	Accuracy	Precision	Recall	F1
Falcon -7B- Instruct	Task 42	makePrompt_ZS_noCoT	0.57	0.57	0.57	0.57
Falcon -7B- Instruct	Task 42	makePrompt_ZS_CoT	0.56	0.56	0.56	0.56
Falcon -7B- Instruct	Task 42	makePrompt_FS_CoT	0.53	0.53	0.53	0.52

Based on the experiments with GPT-4 [12], Falcon-7B-Instruct [3] and Vicuna, it is observed that all of these three models are able to perform the scoring automatically with varying accuracy for different prompt engineering techniques. As shown in Table 3, When zero-shot without chain of thought prompt technique is followed we can observe that automatic scoring accuracy for Vicuna model is 0.608333333 which is higher than that of GPT-4 and Falcon-7B-Instruct model with accuracy of 0.50 and 0.579166667 respectively. Based these results we can say that when zero-shot without chain of thought technique is applied open source LLMs Vicuna shows higher accuracy followed by another open source model Falcon-7B-Instruct. These two models outperforms GPT-4 when followed with Zero-short without COT approach. However overall accuracy for automatic scoring performs in still low with zero-short without chain of thought technique. Hence when we followed zero-shot with Chain of thought and few-shot with chain of thought technique, it is observed that scoring accuracy improves significantly as compared to prompts without chain of thought. In such scenario GPT-4 outperforms other open source LLMs improving the overall accuracy and performance. The scoring accuracy of different LLMs with these three

Table 5: Automatic scoring matrix for Vicuna.

Model	Task	Method	Accuracy	Precision	Recall	F1
Vicuna	Task 42	makePrompt_ZS_noCoT	0.60	0.62	0.60	0.59
Vicuna	Task 42	makePrompt_ZS_CoT	0.59	0.59	0.59	0.58
Vicuna	Task 42	makePrompt_FS_CoT	0.57	0.57	0.57	0.57

prompting techniques is presented in the Figure 3. The above experiments shows that,when different prompt engineering techniques are applied, LLMs can perform the automatic scoring task successfully. However, the automatic scoring performance varies based on the model and the prompt technique.

Even though LLMs can perform the automatic scoring task, educators are concern about the reliability of the scores provided by LLMs. Since scoring the assessment is very sensitive task and directly involves evaluation of the students, it becomes extremely important to know if these scores are reliable and trustworthy. One of the mechanism to understand the reliability of these automatically generated scores is by understanding if LLMs grades as human does? If LLMs follow same grading mechanism as humans then we can say that the scores are reliable . This can be done by adding explainability to the automatic scoring task. This leads us to our second research question.

When performing the scoring task, humans grade according to the set of rules which is called as a analytical rubrics . While performing the scoring of student responses of an assessment task, humans divide the holistic rubrics into analytical rubrics and performs the grading task based on those analytical rubrics. Analytical rubric contains different levels of the scores and student response are graded based on the analytical scoring level satisfied by the student response. In the work presented in "Applying Large Language Models and Chain-of-Thought for Automatic Scoring" [9], the automatic scoring task is performed based on the holistic rubrics [9] and does not involve analytical scoring. In this thesis work we are adding explainability to the automatic scoring task by answering, if LLMs can provide the analytical rubrics breakdown as human does while performing automatic scoring, thus concluding the reliability of the automatic scoring . To achieve this , we first organised our dataset by performing the analytical

breakdown of the rubrics as shown in the Table 1. The input is Question Q , Student Response X , Holistic Rubrics R . The given set of inputs is provided to LLM. LLM then outputs the overall score Y.

- **Inputs**: Question Q, Student Writing X, Teacher Rubrics R.
- **Outputs**: Overall Score Y, Breakdown Points $A = [a_1, \ldots, a_L]$, $a_l \in \{0, 1\}$,

$$Y = WA, \quad W = [w_1, \ldots, w_L] \text{ is a learnable weight vector.}$$

- **Interpretability**: Y is computed by the linear addition of $[a_1, \ldots, a_L]$, and each a_l refers to a clearly defined scoring point (concept).

Given the Question, Holistic rubric to Large Language Model, we developed a pipeline which generates an analytical rubrics breakdown $\mathbf{A} = [a_1, \ldots, a_L]$ for given questions based on the holistic rubrics.

The controlled experiments are performed to verify the hypothesis. Very strong LLMS such as GPT-4, Mistral 7B, Mistral 8x7B are selected for the experiments and these LLMs are able to follow the instructions given by user. These LLMs are selected because of their superior performance and the easy accessibility. The behaviour of these LLMs is controlled by providing the different prompt to instruct the model to successfully perform tasks under different settings. The prompt engineering approach is used to while performing the experiments.

The very first approach of the study is whether LLMs grading aligns with the human educators during automatic scoring task. The question description and holistic rubric is provided to LLMs and asked them to generate the analytic rubric to grade the student responses to the items. The generated analytical rubrics involves multiple logical rules and there is no overlap between such rules. The student response which satisfies more rules gets the higher score. Once these analytical rubrics are generated, we then compare these LLM generated analytical rubrics with the human generated analytical rubrics. If LLM provides the analytical rubrics that matches with the analytical rubrics generated by humans, it is possible to say that LLMs aligns with the human graders. The experiments examine how the models perform in terms of the scoring accuracy.

The experiments are performed based on the 3 main characteristics:
(1). If holistic rubrics written by humans can help LLM understand the assessment item better. (2). We were curious to know if analytical rubrics from other assessment item can help LLM understand the assessment item. (3). It was also analysed whether providing graded student responses could help the process of automatic scoring by generating analytical rubrics.

The second aspect of the study is whether alignment between LLMs and humans also benefit accuracy of the scoring. Here LLM was provided with human written and LLM generated analytical rubrics and instructions were given using prompts to perform the auto-scoring task.

The the conversation with LLMs starts with the following.

System prompt : **"The assistant is an impartial science educator working in a middle school. His job is working under the supervision of the User."**

Analytical rubric generation prompt : **"Your job is to provide the analytic rubric for a science item. The analytic rubric includes a minimum set of rules, each of which covers a specific required action, and their complete collection describes the requirements of the entire task."**

Automatic Scoring Prompt: **Your job is to evaluate the quality of student responses strictly following the Analytic Rubric provided previously.**

The experiments are performed on different large language models such as GPT-4, Mistral 7B and Mistral 8x7B, Where these LLMs are provided with the question (along with background information) and holistic Rubrics and prompts to generate the corresponding analytical rubrics. These LLMs are selected because of their superior performance and the easy accessibility. When we developed the pipeline and performed the evaluation it is observed that LLMs are able to generate the analytical breakdown rubrics for given set of questions and holistic rubrics.

This rubric breakdown is then evaluated against the human breakdown rubrics to validate the accuracy. We used the "cross-encoder/stsb-roberta-large" [] evaluator model by Sentence Transformers cross-encoder class to calculate the semantic similarity between the human analytical rubrics breakdown and machine analytical rubric breakdown. "Cross-encoder/stsb-roberta-large" model performs the semantic similarity between two sentences and then determine the score between zero to one based on two sentences. Here one means that pair of texts are exactly same and zero means the pair is not same at all. The pair of text is considered to be correct if the semantic similarity between LLM generated analytical rubric and human generated analytical rubric is 0.6. Then the precision, recall and F1 score is calculated. Precision is calculated by dividing number of correct rules generated by LLM by total number of rules generated by LLM. Recall score of human generated rules is calculated as number of recalled rule by total number of human written rules.

With 12 assessment Items, the experiments are performed with GPT-4, Mistral 7B , Mistral 8x7B model to generate the analytical rubrics. We then compared machine breakdown of the analytical rubrics with human breakdown of analytical rubrics by calculating semantic similarity and then evaluated the results . The results are presented in the following section.

CHAPTER 5

RESULTS

We performed controlled experiments with different LLMs to generate the analytical rubrics from science items in our assessment items dataset. We then compared these LLM generated analytical rubrics with the human written analytical rubrics to examine if there is an alignment gap between both. We considered different open LLMs such as Mistral-7B, Mistral-8x-7B and other LLM like GPT-4 and performed the experiments to generate the analytical rubrics. For every item in the assessment items dataset, we generate a prompt with system prompt, question with back ground information, holistic rubrics and student response score, instruct LLM to generate an analytical rubric for the assessment items. We initially performed the experiments with GPT-4 and other open source LLMs such as Mistral-7B, Mistral-8x7B due to high performance and convenient accessibility. We compared the LLM generated analytical rubrics with human written analytical rubric using Stsb-roberta-large [] model which is used to calculate the semantic similarity between text. We then calculated the scores as Precision , Recall and F1 Score . Here precision is calculated as number of correct rules generated by LLM divided by total number of rules generated by LLM. The recall is calculated as total number of recalled rules divided by total number of rules generated. F1 Score is then calculated with the formula F1 = 2 * (precision * recall) / (precision + recall).

Table 6: Analytical Rubrics Vs Human Rubrics Breakdown.

Model	Precision	Recall	F1 Score
Mistral-7B	0.76	0.92	0.80
Mixtral-8x8B	0.80	0.87	0.81
GPT-4	0.83	0.90	0.85

The results of analytical rubrics generated by LLMs Vs human generated rubrics are shown in Table 6 . Based on the results in Table 6, we can say that LLMs can generate analytical rubrics when provided with the question, holistic rubrics and score. However there still exit some alignment gap between human written rubrics and LLM generated rubric. Experiments with different LLMs shows varying results. We performed experiments open source LLMs like Mistral-7B, Mistral-8X7B and other LLM such as GPT-4. Out results shows that Mistral-7B generated analytical rubrics with precision of 0.76 , recall of 0.92 and F1 value of 0.80. When tested with Mistral-8x7B, analytical rubrics are generated and the precision is 0.80 , recall is 0.87 with F1 score of 0.81. Along with the these open source LLMs, the analytical rubrics are generated with GPt-4 and the calculated precision is 0.83 , recall is 0.90 and F1 score is 0.85. From the observations and results we observed that even open source LLMs are capable of generating the analytical rubrics with comparable performance with very strong LLM like GPT-4. Considering the cost involved with LLM like GPT-4 we can definitely say that even other source LLMs can be used in automatic scoring task and it is possible to get the comparable results as that of GPt-4. By comparing LLM generated rubrics with human generated rubrics , we can see that LLM understands the assessment items as human however there is still an alignment gap between both.

To evaluate this alignment gap, we then further experimented generation of analytical rubrics under different settings. First setting is generating analytical rubrics by proving remaining items with human written analytical rubrics to LLM. We experimented with total 12 assessment items. We provided the human written analytical rubrics to LLM for 11 other items and then instructed model to generate the analytical rubric for remaining 1 assessment item. This setting is considered as a few-shot setting. The second setting is oneshot where we formatted only one other item with human analytical rubric. Third setting is providing the holistic rubrics to LLMs. Forth setting is providing human graded student responses to LLMs.

The experiments are performed based on above setting involving with 12 different assessment items from the science domain [21]. Considering easy accessibility and performance of Mixtral model, we performed further experiments with Mixtral. The results are presented in the Table 7.

Table 7: Analytical rubric generation task under different settings.

Rubric	Setting	Rules	Precision	Recall	F1
Analytic Rubric	One-shot	3.25±1.09	0.525±0.33	0.701±0.34	0.580±0.32
Analytic Rubric	Few-shot	2.50±0.87	0.688±0.34	0.681±0.34	0.664±0.33
Analytic Rubric	Few-shot with Holistic Rubric	2.75±1.16	0.782±0.31	0.778±0.28	0.752±0.28
Analytic Rubric	Few-shot with graded student response	3.33±1.43	0.326±0.33	0.528±0.36	0.350±0.28
Human Written Analytic Rubric		2.25±0.83	1.000±0.00	1.000±0.00	1.000±0.00

Based on the results, it is clear that LLM (Mixtral in this case) can understand the assessment items and generate the analytical rubrics while performing the automatic scoring task. However,there exists some alignment gap between human written analytical rubrics and LLM generated analytical rubrics because the precision , recall and F1 values are bit away from the ideal score 1. Based on the table presented above, it can be observed that the gap varies based on the different setting of generation of analytical rubrics. The analytical rubric generation task with few-shot setting with holistic rubric provides the best performance as compared to other setting like one-shot, few-shot and few-shot with graded response. The few-shot with holistic rubric shows precision 0.782, Recall 0.778 and F1 score of 0.752. Analytical rubric generation task with few-shot setting performs better than one-shot setting. It means that providing the analytical rubric of other assessment items to LLM actually helps LLM to understand the assessment task better and improves the performance while rubric generation. Thus, improving the alignment with the human graders as well. It is also quiet surprising to observe that setting in which LLM is provided with few-shot setting and graded student response does not actually perform well as compared to other settings.

As we can observe in the Table 7, for different setting different number of rules are generated. In case of few-shot with holistic rubrics setting , the number of rules generated are as same as that of human written rules. However in case of one-shot and fullshot with graded student responses, many incorrect rules are generated. These rules were observed to be incorrect because it has 'incorrect logic generation'

or 'inappropriate expression'. One of the example is shown below.

Human written rubric: "When thermal energy is transferred to the water (hotter), water and dye particles move faster .||| At the higher temperature, water and dye particles move faster."

LLM generated rubric: "When thermal energy is transferred from the water in the first dish (cooled), water particles move slower and the red dye particles settle down.|||When the water in the second dish is kept at room temperature, water particles move at a constant speed and the red dye particles remain suspended."

Here we can observe that, the rules generated by LLM are logical and but the expression is completely opposite. Thus leading to the misalignment with the human written rubrics.

Table 8: Automatic scoring task under different settings of analytical rubric generation.

Rubric	Setting	Automatic scoring Accuracy
Analytic Rubric	One-shot	49.17±13.28
Analytic Rubric	Few-shot	49.41±10.19
Analytic Rubric	Few-shot with Holistic Rubric	54.58±9.01
Analytic Rubric	Few-shot with graded response	48.41±10.17
No Analytic Rubric		34.83±11.50

Based on out LLM generated analytical rubric, we then performed the experiments to perform the automatic scoring task under same setting and observed the accuracy. The experiments are then performed to examine if LLM can produce more accurate feedback to student responses with better understanding of analytical rubric. We examine how different settings of analytic rubrics generation affects the automatic scoring accuracy. More accurate analytical rubrics generated more accurate scoring. In our earlier results we observed that few-shot setting with holistic rubrics showed the best performance for generating analytical rubrics. We then instructed LLM to perform the automatic scoring and observed that few-shot with holistic rubric strategy provided the better automatic scoring performance. The automatic scoring

accuracy results are shown in the Table 8. Here we observed that the automatic scoring accuracy of the few-shot with holistic rubric setting 54.58 and it outperforms remaining settings. The student responses graded with one-shot setting of rubric generation show the accuracy of 49.17 percent . Whereas with few-shot setting the scoring accuracy is 49.41 percent. Based on this results we observed that high quality analytical rubrics help LLM to generate more accurate results,thus enhancing the automatic scoring accuracy.

CHAPTER 6

SYSTEM DESIGN

In the realm of education, the demand for efficient automatic scoring methods that provide personalized feedback in a timely manner is paramount. To address this, we have developed a system "GPTest" which performs the automatic scoring task with powerful large Language Model GPT-3.5-turbo. We developed an end-to-end web application which can be used by teachers and students. Through this web application teachers and students can efficiently manage the automatic grading process and receive immediate feedback from the LLM. This system has a potential to provide the personalised and immediate scoring to the students and streamlining the assessment process in the educational domain. The systems enables teachers to create a class, add students, create an assessment, upload rubrics and personalised prompts, assign the assessment to the desired class and share it with students. Teachers can also make the assessment public or private. Public assessment enables teacher to share assessment with other instructors as well. Private assessments are the once which are only accessible be teachers who created it.

Once the assessment is assigned to a class, all the students who belongs to that class can access the assessment and submit the response. Once students submit the response, that response is being sent to GPT-3.5-turbo and it then processes that response. LLM is also provided with the assessment details and rubrics from the database and then LLMs automatically generates scores and the feedback. Once the scores are generated by GPT-3.5-turbo, system then send the response to teachers and students. This way students get immediate response for their submission. If required teachers can then revise the scores/ feedback generated by LLM and share their revised feedback with students . This GPTest system is the management system which can be used in all the educational setting and enhance scoring process the

scoring process with help of automatic scoring with GPt-3.5-turbo. Currently, we have developed the system using GPT, However it is very much compatible with any other open-source LLM.

6.0.1 GPTest : Key features

End-to-End Web Application

GPTest encompasses a user-friendly web application facilitating seamless interaction between teachers and students. Teachers and students can efficiently manage the automatic grading process and receive immediate feedback from the GPT-3.5-turbo language model.

Class Management

Teachers have the capability to create classes, add students, and organize assessments within the system. This feature enables efficient organization and management of student assessments within various educational settings.

Assessment Creation and Customization

Teachers can create assessments tailored to specific learning objectives.Customization options include uploading rubrics and personalized prompts to guide student responses.

Assignment Distribution

Assignments can be effortlessly allocated to desired classes, with options for public or private access. Public assessments can be shared with other instructors, fostering collaboration and resource sharing. Private assessments remain exclusive to the teacher who created them, ensuring confidentiality.

Student Submission and Feedback

Once an assessment is assigned, students belonging to the respective class can access and submit their responses. Submitted responses are processed by the GPT-3.5-turbo model, which generates automatic scores and feedback based on assessment details and rubrics stored in the database.

Immediate Feedback Loop

Upon score generation, both teachers and students receive immediate feedback on the submitted responses. This real-time feedback mechanism enhances student engagement and learning outcomes.

Revision and Collaboration

Teachers have the option to review and revise scores and feedback generated by the GPT-3.5-turbo model. Revised feedback can be shared with students, promoting iterative learning and improvement.

6.0.2 GPTest : Architecture and Functionality

GPTest is a end-to-end system which performs the automatic scoring using LLMs and can be used in educational settings by teachers and students to streamline and scoring and assessment procedure. Figure 6.1 demonstrate the GPTest Architecture. As shown in the architecture diagram, GPTest supports two users,first one is teacher and other is student. GPTest enable users to perform following functionalities:

Class Creation

GPTest enables teachers to create a class by allowing teachers to upload a excel sheet which then creates a class based on the student data from the sheet.

Student Registration

Once the class is created , teacher is able to perform the student registration by adding the student in the class and removing student from the class.

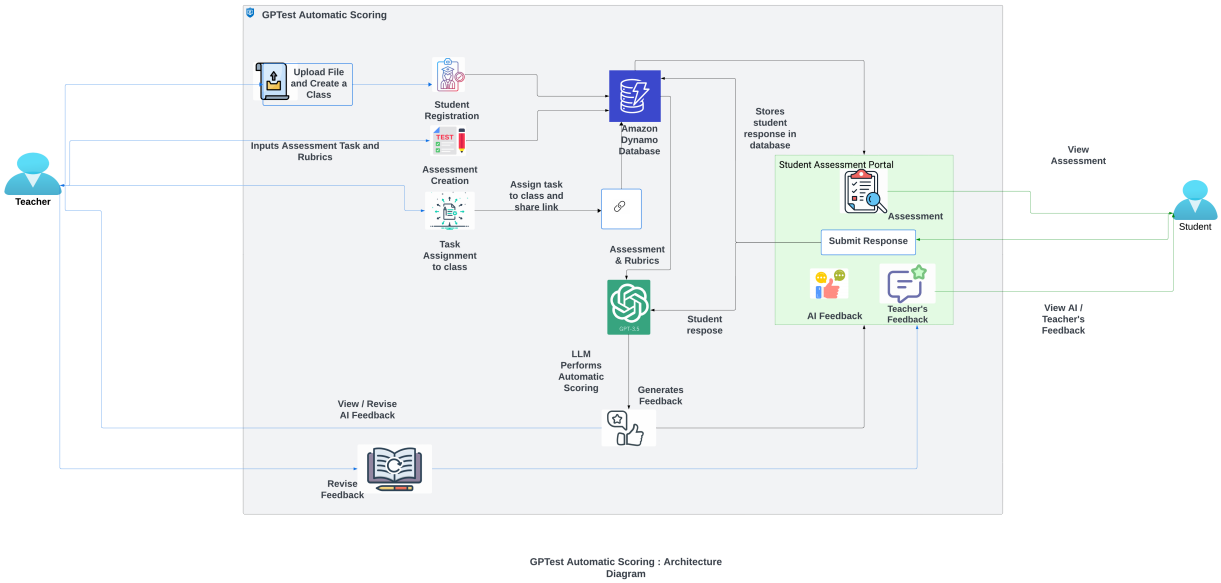

Figure 6.1: GPTest automatic scoring architecture diagram.

Assessment Creation

Once the class is created and student registration is done, teacher can then create a new assessment task. This can be done by uploading the assessment details such as assessment title, assessment description, rubrics based on which the grading is supposed to be done. System also prompts teacher to select any specific grading criteria or enter the grading criterion as per teachers choice. Once all of these assessment details are provided by teacher, system then allows user to validate the assessment before finalising the assessment task. This validation is a crucial phase since teacher provides all the assessment details along with the sample student response to LLM(GPT-3.5-turbo here). Once the sample student response is provided teacher can then validate this new assessment with LLM and check if LLM is able to perform the automatic scoring as per the teacher's expectations or not. Once the assessment is validated, teacher can provide final approval for the assessment creation and creates the assessment task.

Assessment allotment and assignment

Once the assessment is created, teacher then assigns this assessment task to the class of his choice, with due date and number of attempts. System then generates an link which can then be shared with students by email supported by GPTest .

Assessment response by students

Once student receives the assessment email,then each student can submit his/her student response along with the student's unique ID.

Scoring by Large Language Model

Once the assessment response is submitted by Students, system then passes this response to then large language model (GPT-3.5-turbo). The other assessments details for example as assessment rubrics, assessment description, teacher's prompt with specific grading instruction all these are required for automatic scoring are then fetched from the database by system and then provided to Large Language model. System then performs the automatic scoring task and generates AI feedback and scores for the assessment. This feedback and score is then provided to students on their assessment portal and also to teachers.

View Feedback

Student can then view the AI generated feedback instantly after the assessment response submission which enable student to get response quickly and get the personalised feedback in no time. Also these AI generated scores and feedback is then provided to teachers. Teachers can then access these scores for every student and if required then teacher can also revise the scores and share the modified feedback to the students. In this way GPTest enable educationalist to streamline the automatic scoring process in the educational setting.

6.0.3 GPTest : Implementation

Python API

The python POST API is created using Google Colab to perform the automatic scoring task.The python pipeline is created with assessment description , assessment rubrics , teacher's customised prompts ans student response as an input .The python api is then connected with GPT-3.5-turbo using Open AI API key and connection between python pipeline and LLM is established with the help of API key . Here we have used GPT-3.5 . However this can be replaced with any other LLMs even different open source large language Models such as Falcon , Mistral , Vicuna etc depending on the requirement. This python API then inputs the assessment details along with student response and rubrics entered by teachers and then provide these details to LLMs in the form of Prompts . This prompt is structured as shown below.

Listing 6.1: Prompt Structure

```
def make_prompt(Rule, Task_Description, Rubrics,
   Teachers_Prompt):
    return f"{Rule}\n\nTask_Description: {Task_Description}\n\
       nRubrics: {Rubrics}\n\nTeachers_Prompt: {Teachers_Prompt
       }"
```

This prompt along with the student response is the feeded to the GPT-3.5-trubo to perform the automatic scoring task and then LLM performs the automatic scoring and generates the scores and feedback. This API is developed with python flask web framework to develop the API functionality .

Web Application

The web application is developed using Angular typescript based web application framework and Python Flask . The user interface for the GPTest is designed and developed using angular and the backed logic is handled using typescript . For each of the functionality listed above such as login, class creation , student registration , assignment creation , assignment allotment different angular component is created

for each of these functionalities to develop the user interface . Python flask framework is used integrate the user interface and python automatic-scoring API .

AWS Dynamo DB

The AWS Dynamo DB is a NoSQL database . It is used in GPTest to manage the database storage and database related operations . Since the web-application is developed in angular , we performed the angular and dynamo db database connectivity using AWS amplify . We have used AWS(Amazon Web Services) API gateway to create different APIs to perform CRUD such as create , Read , Update and Delete operations on the database. AWS API gateway enabled GPTest to have the POST and GET API functionalities to the database. We have then used AWS Lambda which actually has the function implementation for all these CRUD operation . AWS dynamo db can be accessed by API gateway via Lambda functions. This is how we performed the web application to database connectivity for GPTest.

Express Server

Express server is used in GPTest to create an node.js rest API to implement the the class creation by uploading the class template file with student data .

Technology Stack

Following the the technology stack used for the development of end-to-end system GPTest which streamlines the automatic scoring task.

The short tour of GPTest System is as shown here .

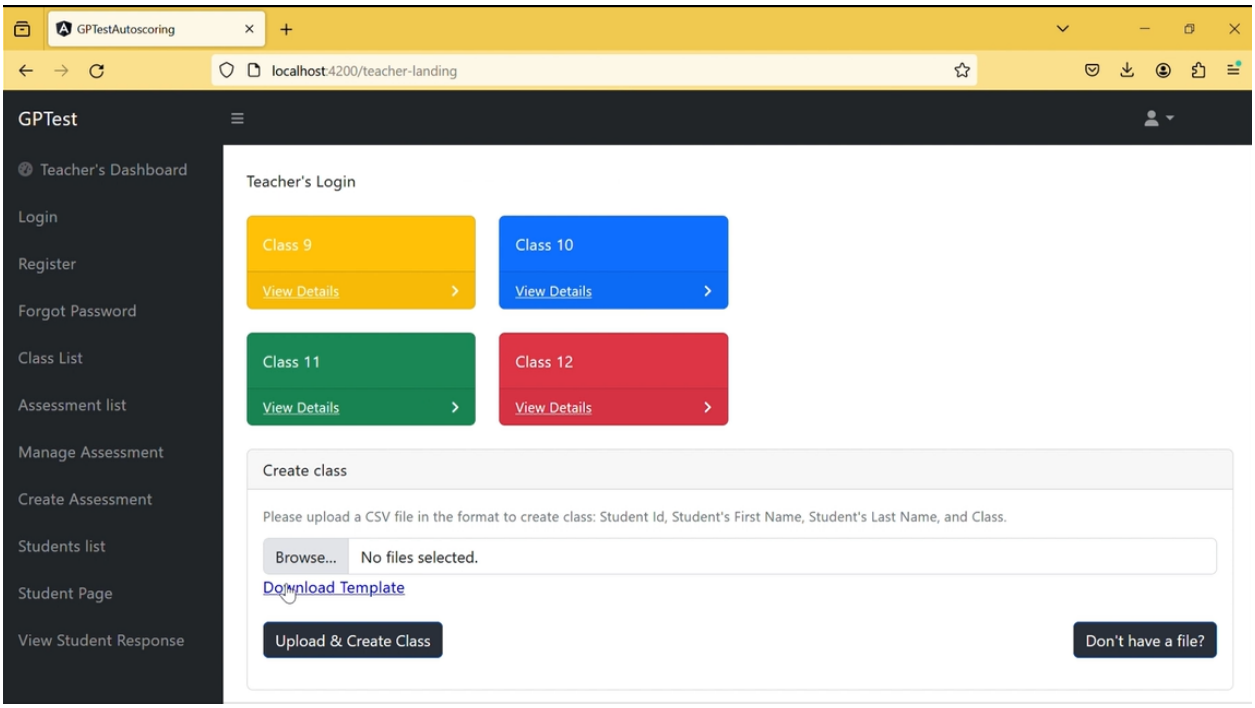

Figure 6.2: Class creation

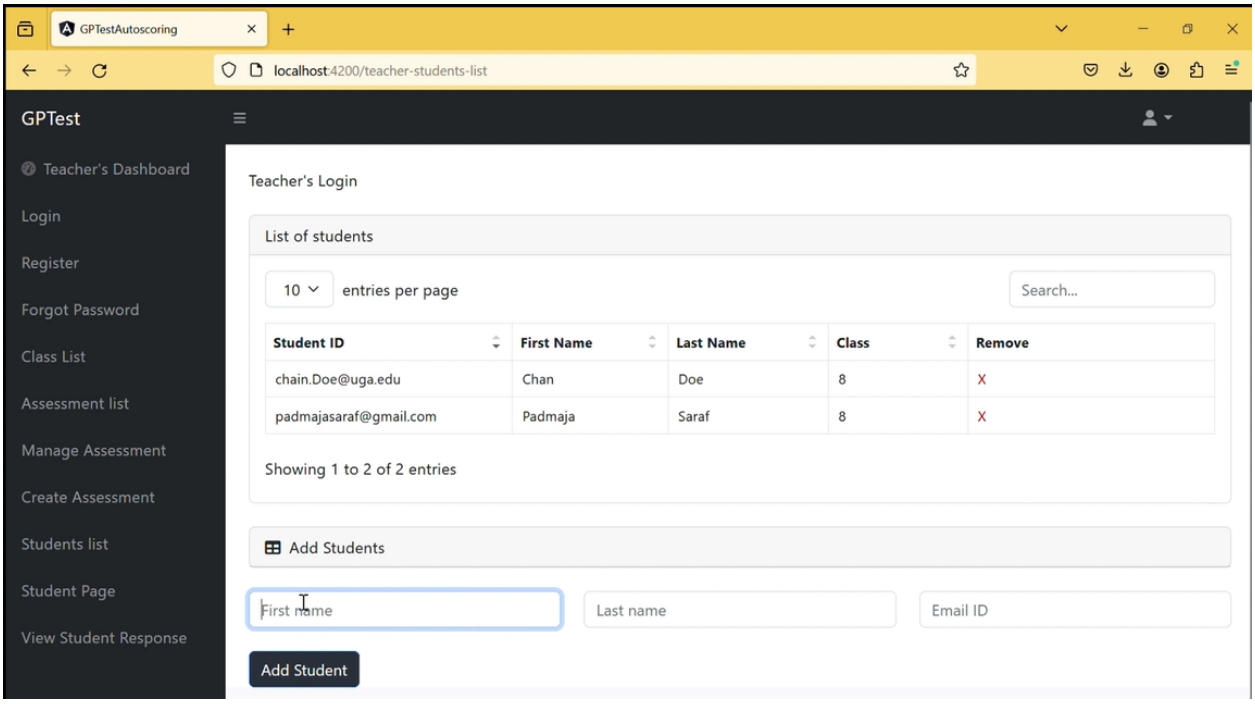

Figure 6.3: Adding student to class

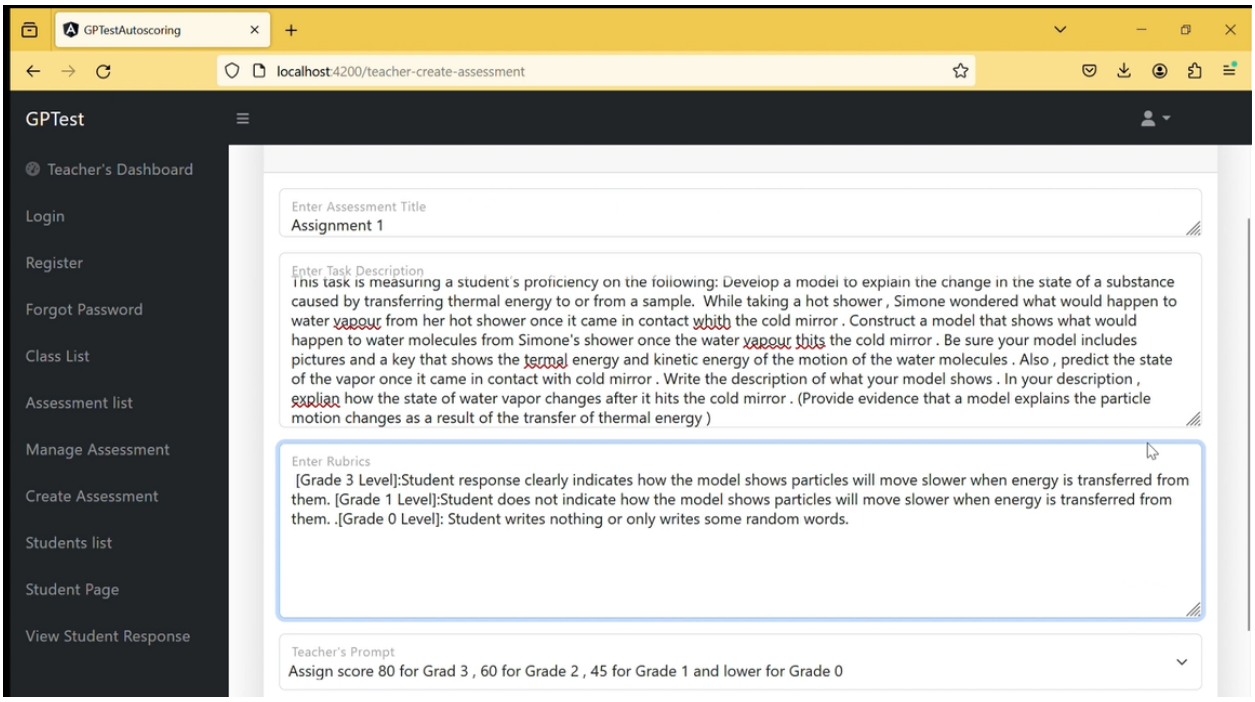

Figure 6.4: Assessment creation

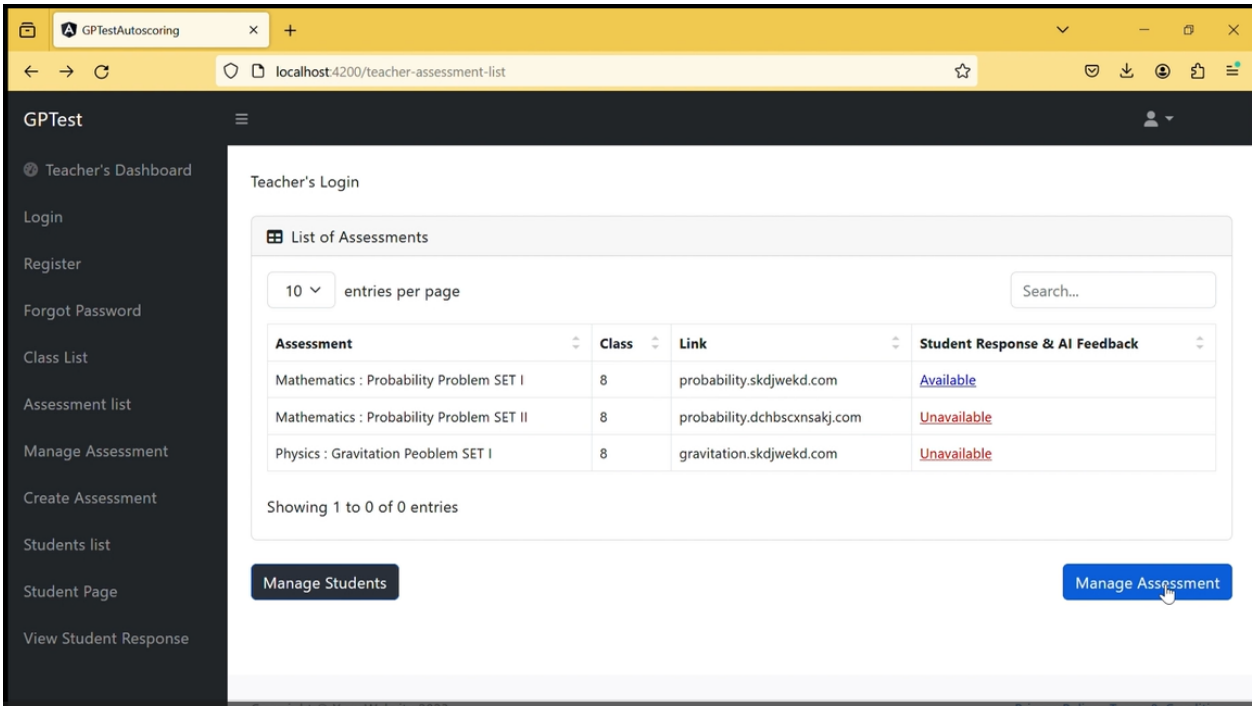

Figure 6.5: Assessment list

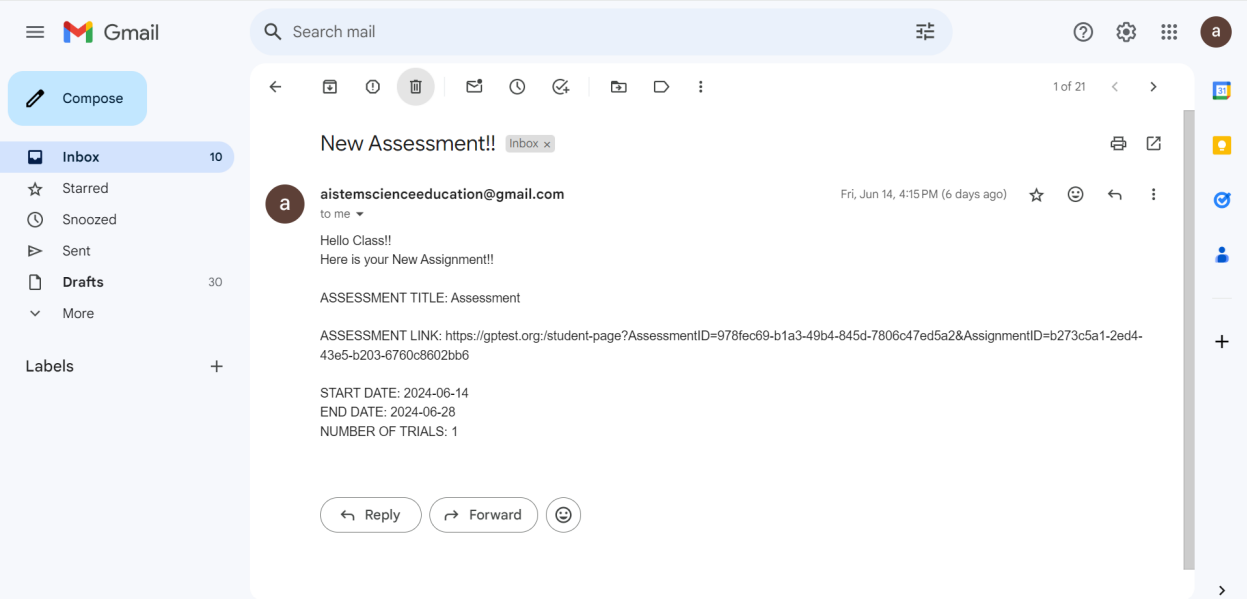

Figure 6.6: Assessment email

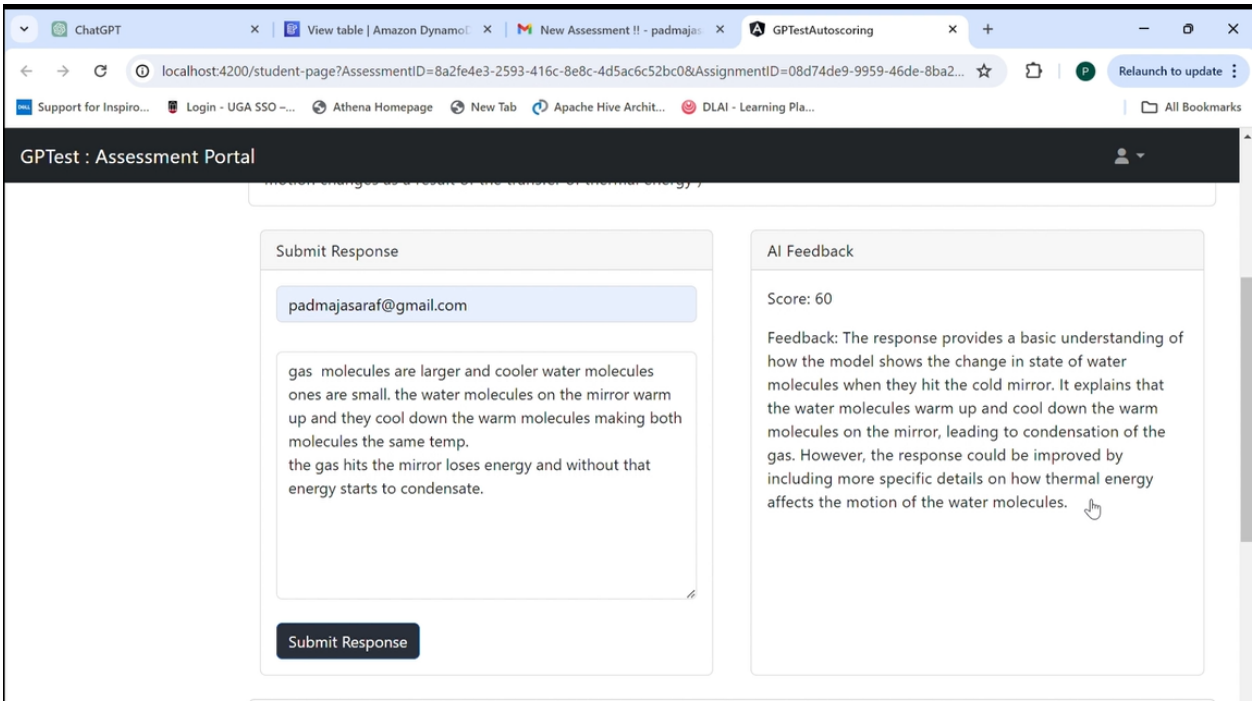

Figure 6.7: AI response

Table 9: Technology stack used for development of GPTest.

Technology	Version
Python	3.11.1
Angular	15.2.10
Typescript	4.9.5
DynamoDB	2019.11.21
Pycharm	2023.1
NodeJS	20.8.1
Express Server	4.19.2
Google Colab	21.2.2024
AWS Amplify,Lambda, API Gateway	6.2.0

CHAPTER 7

LIMITATIONS AND FUTURE WORK

This thesis focuses on the task of automatic scoring, a critical area as it directly impacts the assessment and evaluation of students. The results indicate that large language models (LLMs) can perform automatic scoring with varying degrees of accuracy depending on the specific model and prompt engineering techniques employed. Among the models tested, GPT-4 achieved the highest accuracy at 66 percent, followed by the Vicuna model at 60 percent, and the Falcon-7B-instruct model at 57 percent.

Despite these promising results, the current accuracy levels suggest that automatic scoring systems are not yet reliable enough for real-world educational settings. There is a clear need for further improvement in accuracy to ensure the reliability and fairness of these systems.

Future research could focus on enhancing the accuracy of automatic scoring by using prompt engineering techniques while generating analytical scoring rubrics. Thus, ultimately improving the automatic scoring accuracy with high quality analytical rubrics. Additionally, integrating recent advancements in multi-agent systems could further improve the accuracy of automatic scoring tasks with help of multiple LLMs in system. Such improvements could enable AI-powered automatic scoring systems to be more effectively utilized in educational settings.

CHAPTER 8

CONCLUSION

In this thesis, we presented an explanability for the automatic scoring task using LLM such as GPT-4 and other open source LLMs like Mistral, Falcon and Vicuna with varying configurations and different prompt engineering strategies. We successfully compared the automatic scoring accuracy for different open source LLMs and performed the comparative analysis. In this study we investigated the alignment gap between LLMs and human graders for automatic scoring tasks. We observed that there exits an alignment gap between human graders and LLMs. However, our further experiments proposed that minimising the alignment gap between the analytical rubrics generated by LLMs and humans can significantly improve the overall scoring accuracy of the LLMs. Further, this work successfully presented an end-to-end system "GPTest" which performs the automatic scoring using GPT which can be used in educational setting by teachers and students to streamline the scoring and assessment procedure.

BIBLIOGRAPHY

[1] Alexandre Sablayrolles Albert Q. Jiang et al. "Mistral 7B". In: *arXiv:2310.06825* (2023). URL: https://arxiv.org/abs/2310.06825.

[2] Noam Shazeer Ashish Vaswani et al. "Attention Is All You Need". In: *arXiv:1706.03762* (2017). URL: https://arxiv.org/abs/1706.03762.

[3] Hamza Alobeidli Ebtesam Almazrouei et al. "The Falcon Series of Open Language Models". In: *arXiv:2311.16867* (2023). URL: https://arxiv.org/abs/2311.16867.

[4] Changqing Zhang Huan Ma et al. "Fairness-guided Few-shot Prompting for Large Language Models". In: *arXiv:2303.13217* (2023). URL: https://arxiv.org/abs/2303.13217.

[5] Thibaut Lavril Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: *arXiv:2302.13971* (2023). URL: https://arxiv.org/abs/2302.13971.

[6] Ming-Wei Chang Jacob Devlin, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv:1810.04805* (). URL: https://arxiv.org/abs/1810.04805.

[7] Xuezhi Wang Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *arXiv:2201.11903* (2022). URL: https://arxiv.org/abs/2201.11903.

[8] E. Latif and X. Zhai. "Fine-tuning ChatGPT for Automatic Scoring". In: *arXiv:2310.10072* (2023). URL: https://arxiv.org/abs/2310.10072.

[9] Gyeong-Geon Lee et al. "Applying Large Language Models and Chain-of-Thought for Automatic Scoring". In: *arXiv:2312.03748* (2023). URL: https://arxiv.org/abs/2312.03748.

[10] Yinheng Li. "A Practical Survey on Zero-shot Prompt Design for In-context Learning". In: *arXiv:2309.13205* (2023). URL: https://arxiv.org/abs/2309.13205.

[11] Yinhan Liu Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *arXiv:1910.13461* (2019). URL: https://arxiv.org/abs/1910.13461.

[12] Josh Achiam OpenAI et al. "GPT-4 Technical Report". In: *arXiv:2303.08774* (2023). URL: https://arxiv.org/abs/2303.08774.

[13] S. Acar P. Organisciak, D. Dumas, and K. Berthiaume. "Beyond semantic distance: Automated scoring of divergent thinking greatly improves with large language models. Thinking Skills and Creativity". In: ().

[14] Weizhe Yuan Pengfei Liu et al. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *arXiv:2107.13586* (2021). URL: https://arxiv.org/abs/2107.13586.

[15] Z. Eberhart S. Haque, A. Bansal, and C. McMillan. "Semantic similarity metrics for evaluating source code summarization." In: *arXiv:2204.01632* (2022). URL: https://arxiv.org/abs/2204.01632.

[16] M. Koo S. Wu et al. "A comparative study of open-source large language models, gpt-4 and claude 2: Multiple-choice test taking in nephrology." In: *arXiv:2308.04709* (2023). URL: https://arxiv.org/html/2404.14445v1.

[17] Shixiang Shane Gu Takeshi Kojima et al. "Large Language Models are Zero-Shot Reasoners". In: *arXiv:2205.11916* (2022). URL: https://arxiv.org/abs/2205.11916.

[18] Benjamin Mann Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *arXiv:2005.14165* (2020). URL: https://arxiv.org/abs/2005.14165.

[19] Xuansheng Wu et al. "Matching Exemplar as Next Sentence Prediction (MeNSP): Zero-shot Prompt Learning for Automatic Scoring in Science Education". In: *arXiv:2301.08771* (2023). URL: https://arxiv.org/abs/2301.08771.

[20] Xiaoming Zhai. "Advancing automatic guidance in virtual science inquiry: from ease of use to personalization. Educational Technology Research and Development". In: (2021).

[21] Haudek Kevin Zhai Xiaoming et al. "From Substitution to Redefinition: A Framework of Machine Learning-based Science Assessment". In: (2020).