

# Fair Feedback Systems for Academic Assessment using LLMs

Zhefan Cheng, Samuel Goldie, Collin Makuza, Álvaro Menéndez Ros, Zaker Omergeel, Ada Panghe, Arseny Trifonov

*Department of Advanced Computing Sciences  
Maastricht University  
Maastricht, The Netherlands*

**Abstract**—In recent years, significant advancements have been established in the domain of natural language processing, particularly in the capacity to provide feedback for responses to open-ended questions. The automation of feedback provision holds substantial promise, benefiting not only teachers grading student papers, but also the students themselves. This advancement lies in the enhancement of a model that exhibits qualities of fairness and a comprehensive capacity to evaluate responses across a diverse spectrum of subjects and disciplines.

This research delves into the state of the art in constructing such a feedback tool, considering large language models like GPT, BERT, and Llama as fundamental building blocks. The project, aimed at creating a product for "ANS," adopts a user-friendly graphical user interface, facilitating feedback collection from teachers and educators.

Key data sources, including QuAIL, SQuAD, and NarrativeQA, help drive model refinement, testing, and comparative analysis. The ensuing investigation investigates essential questions concerning the impact of scoring on the grading process, the disclosure of scores and confidence percentages, and the ethical and practical challenges associated with potential biases.

Additionally, the study explores the implementation of the feedback system within the platform, offering diverse approaches. Also, we have adopted a traffic light confidence system to help guide teachers through the automatic grading since we don't provide a concrete score for the question. The overarching goal is to generalize this concept across multiple fields and integrate various methodologies. Ultimately, we aim to deliver a comprehensive tool applicable to assessing handwritten and computer-based examinations.

**Index Terms**—feedback grading, generating model answers, LLM, large language models, grammar checking, spell checking

## I. INTRODUCTION

Students need a certain amount of exercise and feedback in most topics to be able to truly master the topic. Due to the large class sizes and lack of time, it is becoming more difficult for teachers to grade and offer extensive feedback. One significant improvement in natural language processing is the ability to provide feedback to answers that were given to open-ended questions. Such an automatization of receiving feedback could be beneficial for both teachers but also for students themselves. We want such a model be fair, unbiased and well trained for handling answers from various topics and fields.

First, we will discuss the state of the art concerning the origin of such a feedback tool. We undertake a comparative analysis

of existing large language models (LLMs), including but not limited to GPT [9], BERT [5], and LLaMA [2], as these models form the foundational basis of our tool. Given the project's objective to construct a product for the potential utilization by ANS, we decided to design a user-friendly graphical user interface (GUI) that mirrors their platform. ANS is an online company that started as a student project in 2015. Nowadays they help teachers around the world in their process of evaluating students. Their goal is to help assessment of students and make this process more enjoyable than ever before.

Our data sources for fine-tuning, testing, and comparative analysis primarily consist of the datasets provided by QuAIL [16], SQuAD [13], and NarrativeQA [6]. These datasets play a fundamental role in the training and validation of our models. With this research paper we aim to answer to the following research questions:

- What models and approaches make an accurate system to provide feedback?
- What limitations would they introduce?
- What metrics would be most effective in the performance of the model?
- How would automation would influence grading? Is it possible to avoid bias? If not, how can we minimize it?

We conducted a questionnaire to discern the main concerns and the potential use of this tool within the target audience. The survey revealed that teachers had expressed significant apprehensions about potential biases in their grading process. Concurrently, they expressed a strong desire for a tool capable of efficiently highlighting correct sections within answers, thereby enhancing the speed and accuracy of exam grading. A range of significant questions is explored in this academic paper. We address the impact of providing scores on the grading process and deliberate on whether scores and confidence percentages, indicative of the model's degree of certainty in grading accuracy, should be disclosed. This rises the question of how bias would affect this model and the ethical implications that could arise. This report presents our methodology and solutions to mitigate these challenges.

Furthermore, we explore various approaches, while providing motivation for the incorporation of a traffic light confidence system. Our objective is to explore of how this concept can

be generalized to diverse fields and enriched with additional methods and models. Ultimately, we aim to develop a comprehensive tool applicable to both handwritten and computer-based examinations that will help teachers grade faster. A diagram showing the overall functioning of our product:

Tasks	Answer evaluation	Adding relevant tags to questions	Grammar checker
How	Assessing how much an answer conforms to a rubric.	Given a list of possible tags, determining which one(s) are more relevant	Highlighting typing or grammatical errors
LLM's	AQA, BERT, LLaMA	GPT-3.5, BART	Flan-T5

Fig. 1: Diagram of the final product capabilities

## II. STATE OF THE ART

### A. Transformers

In the domain of natural language processing (NLP), recent research has highlighted the emergence of transformer models, including GPT [9], BERT [5], and LLaMA [2]. These transformer-based architectures have proved remarkable performance across a variety of NLP tasks.

An advancement achieved by transformers in the field of NLP relates to the refinement of techniques facilitating the generation and identification of sentences conveying similar meanings in paraphrasing. The objective of the Paraphrase Generation module in the referenced article is to generate paraphrases that are both fluent and semantically similar. The Paraphrase Identification system is tasked with determining whether pairs of sentences constitute paraphrases. The proposed approach combines data sampling with a Text-To-Text Transfer Transformer (T5) model [11]. The paper "Optimization of paraphrase generation and identification using language models in natural language processing" emphasizes a unified approach for solving both Paraphrase Identification and Generation problems using carefully selected data-points and a fine-tuned T5 model. [10].

An advancement of the Transformer is a new architecture based solely on attention mechanisms, eliminating the need for recurrence and convolutions [15]. The Transformer model presented in the paper "Attention Is All You Need" generalizes well to other tasks, demonstrated by its successful application to English constituency parsing with both large and limited training data. As this paper describes this model allows for significantly more parallelization compared to recurrent models, reaching a new state-of-the-art translation quality.

As a result we have decided to proceed with the implementation of transformers as they are more versatile and faster being able to improve the quality of a model made from scratch after training for as little as twelve hours on eight P100 GPUs. [15]

## III. DATA

In the pursuit of a comprehensive dataset encompassing diverse domains to meet the exigencies of building, training, and testing our model, we meticulously selected three primary datasets:

- 1) **QuAIL**: A reading comprehension dataset comprising approximately 15,000 multiple-choice questions spanning four distinct domains, namely news, user stories, fiction, and blogs.
- 2) **SQuAD (Stanford Question Answering Dataset)**: Originally designed for a fundamentally distinct task, SQuAD has been harnessed to provide essential model training stimuli.
- 3) **NarrativeQA**: This dataset centers on questions, answers, and their respective sources.

Our primary dataset for this project is QuAIL, with the other two datasets playing a supplementary role in specific tasks. For our dataset composition, we delineated the structure as follows: each entry comprises a question, a context (akin to a student's answer in this context), and the correct answer in the multiple-choice format, serving as the grading criteria.

## IV. GUI

### A. Questionnaire

1) *Objective*: The questionnaire was designed to gather insights on the potential use and concerns regarding an automated feedback tool for grading exams. Specifically, it aimed to understand how professors would utilize such a tool in their grading process, apprehensions about possible biases introduced by the tool, and the overall needs of the teachers.

2) *Survey-Based Approach to Methodology*: The survey was conducted among faculty teachers at Maastricht University, facilitated by our project supervisor, Katharina Schneider. This questionnaire was sent to the teachers at the faculty of Data Science and Engineering and the School of Business and Economics. This approach ensured that the feedback was directly relevant to our target user group and their specific teaching and grading environments.

3) *Survey Questions and Responses*: Key questions included inquiries about the interest in a feedback system that identifies strengths and weaknesses in exam answers, the use of a system that additionally provides scores, and their biggest concerns about such a tool. The predominant concern among teachers was the potential bias in grading, especially if the model outputs a score. This concern underscores the necessity for a tool that aids in the grading process without influencing the final decision. Teachers emphasized their commitment to fairness in grading and expressed a desire for a tool that not only expedites the process but also contributes meaningful feedback. The insights gathered from the questionnaire were decisive in shaping the design of our tool and its GUI. Considering the apprehensions about bias, we focused on developing a system that highlights the strengths and weaknesses of answers without providing a definitive, numerical score. This approach aligns with the teachers' preference for a tool that

assists in the grading process but leaves the final judgment in their hands.

## B. Description of GUI

Our GUI uses student names, student answers, question text, and the rubric for each question as input data. Using the model we generate the answer highlighting paragraph index values for the corresponding rubrics and highlight the corresponding paragraphs accordingly. We also generate the labels of the questions according to the model and display them in the right column of the GUI. You can find each question in the paper by selecting one of them in the bottom sliding bar. Then you can select the student answer you want to view. The rubric for each questions in the left column. Click to select one of the rubrics and the corresponding answer paragraph will be highlighted in the answer text. The color next to the rubric shows the confidence level of the rubric for the current paper question. The reason we have chosen to not portray the confidence level, but instead use 3 classifications is to reduce any potential bias introduced from the teacher. From the questionnaire we have seen that teachers prefer a tool that helps them grade, instead of returning a score as this score would incline the teachers accept it as completely correct or wrong, without checking it themselves. Whereas a classification would be more ambiguous and require the teacher to double check it. Below the student's answer text, the corrected version of the spelling and grammar is processed and displayed. A picture of this design can be found in Fig 12 in Appendix. When we initialize the software, a small database is created on the device where the software is running, which is initialized based on mysql. This database creates and stores tables with all the responses, questions, rubrics, and student names, based on the relationships between them. The database does not directly affect the implementation of any of the core algorithms within the software, but it is there so that any process that needs to initialize the model and go through the algorithms is run once, and then the required data is inserted into the corresponding database table, which makes it possible to retrieve the data the next time a repeat operation is performed without having to wait for the data to be retrieved. This makes the GUI more user-friendly.

1) *Database structure:* The entire database consists of five tables, namely "account", "answers", "exam", "highlights", "tag".

answers	account
name VARCHAR	name VARCHAR
question LONGTEXT	accountName VARCHAR
answer LONGTEXT	password VARCHAR

exam	highlights
question LONGTEXT	rubric TEXT
rubric TEXT	answer LONGTEXT
	startIndex INTEGER
	endIndex INTEGER
	confidence DOUBLE

Tag
question LONGTEXT
label VARCHAR

Fig. 2: Database Structure

## V. IMPLEMENTATION

### A. Tags generation

1) *BART and DeBERTa:* Two state-of-the-art zero-shot classification models were compared to add relevant tags to given questions. These models are **DeBERTa-v3-base-mnli-fever-anli** [7] and **BART-large-mnli** [8].

In order to use a zero-shot classification model, we need the sentence we want to suggest tags, and the potential tags that the sentence could be associated with (the candidate labels). The model outputs a list of probabilities for each candidate label indicating how related the sentence and the label are. Considering the candidate labels to be "geometry", "math", "economy", "arithmetic", "programming", "chemistry", the model it is able to correctly identify the domain of some sentences:

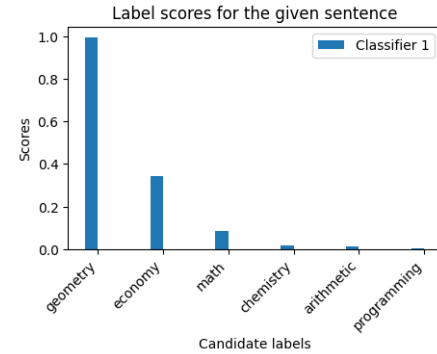


Fig. 3: Sentence: Draw a rectangular prism with the length twice as long as the width and the height

2) *GPT 3.5:* We also tried using **GPT-3.5-turbo** [18] for the tag generation task, without using a list of candidate labels. The main difference between GPT-3.5 and zero-shot-classifiers is that GPT-3.5 is a multi-purpose model, allowing for instructions. The instructions given were as following: *Suggest up to three tags that represent the topic of the user input. Return 'none' if no relevant tags are found. Example: The sun sets early in the winterTags: Nature, Seasons, Daylight.*

### B. Attributed Question & Answering

As our questionnaire had found, teachers preferred a tool that would aid them in correcting the students' open-ended answers which however does not output an estimate of the grade, effectively minimizing bias. We therefore designed a tool that highlights the sections relevant to the grading in a given answer. The program works as follows. It takes as input a student's answer as well as requirements that the teacher will use to grade the student, and for each given requirement, it will highlight, if they exist, the parts in the student's text answer that best satisfies the requirement. The model does not grade the sub-text, this task is left to the teachers.

To build such a tool, we employed the Attributed Question & Answering (AQA) model from Google's Generative Language API [1]. The AQA model differs from other LLMs in that it will base its responses on only a given source; such a source can be a chunk of text, a document, or a corpus of documents. It also provides the added functionality of providing the grounding attributions of its response, i.e. the chunks of text in the source that have been used to build the response. See in the Appendix for figures showcasing the Request and Response of the API.

By translating the teacher's requirements as a query to the student's answers, we can extract the grounding attributions and the answerable probability, which therefore represent the parts of the student's answer that satisfy the requirement as well as the model's confidence in its assessment.

As for the handling of the text data, we create a Document object from the `google.ai.generativelanguage` library for each student answer, and incorporate a Chunk object for each sentence.

This method works best with requirements pertaining to factual accuracy such as those of the form: "Answer mentions ..."; but does not handle those pertaining to the structure or the grammar/spelling of the answer. Other methods need to be implemented in Tandem to deal with those issues

### C. BERT fine-tuned on SQuAD

The second method that was implemented to build the tool is based on Google's BERT model fine-tuned on the SQuAD. Indeed, after fine-tuning on the SQuAD, the model excels in the task of question answering and learns to predict the answers to questions by understanding the context within a given passage. Given a question and a context text, the model learns to identify the start and end tokens in the passage that form the answer to the question. It does so by assigning a "Start\_Score" and an "End\_Score" to each token in the given passage.

Similarly to the previous method, the algorithm we developed takes as input the student's answer as well as a requirement for the answer given by the teacher. We then use the BERT model to retrieve the Start and End scores of each term in the passage.

We then perform a Relevancy Selection during which we retrieve the highest-scoring start term and the highest-scoring end term after filtering out the terms with scores below 0.

Indeed, since the scores are reflective of the model's confidence, if the scores are very low, then the model does not find a conclusive answer to the question in the passage. It would be interesting for further studies to investigate the best threshold for such a task using statistical methods.

Once we have the highest scoring Start and End terms, we can output which segment of the student's answer best fulfills the teacher's requirement.

The AQA method was preferable to this method because this method does not perform on students' answers to the teacher's requirements that were spread around the overall answer. For example, for requirements such as: "mentions 3 different alternatives to the chosen method". Indeed, the BERT model outputs only one segment of the given student answer whilst the AQA-based method can output multiple separate grounding attributions if need be. Additionally, with the BERT-based method, the relevancy selection step with the given threshold is needed to determine if the segment is relevant enough to be returned. In contrast, with AQA, an answerable probability is returned which directly determines the relevance of the requirement w.r.t. the student's answer, therefore, no manual thresholding is needed.

### D. Answer correctness evaluation with Llama

The next experiment was aimed at determining answer correctness was concluded using Llama 2 (7 billion chat variant), a LLM released by the Meta company.

This model was fine-tuned using modern approaches for large model model adaptation, namely Parameter-Efficient Fine-Tuning [17] and Quantized Low-Rank Adapters [4], methods allowing billion-parameters models to be fine-tuned and run on commodity hardware by fixing model parameters and adding a small set of trainable parameters, and quantizing float16 weights for linear layers to the int4 type, reducing model size almost by a factor of 4.

All three datasets (QuAIL, SQuAD v2, NarrativeQA) were included in training by transforming them to a common structure and arranging them by triplets containing a sample from every dataset. Both training and validation set were created in this manner. Every sample consists of a question, answer, context, expected response and an optional question domain.

```

<s><s>[INST] <<SYS>>
You are acting as a responsible and thorough examiner. You will be
presented with a question domain, question context, the question itself and
a student answer to that question. Treat context as an undisputable source
of truth. Your goal is to decide answer correctness. The output should be
formatted as paragraphs with each paragraph starting with the corresponding
section name followed by a colon, space and content. Output should contain
the following sections:
- Response, containing a single word determining answer correctness:
  "correct" if the answer is correct, "partial" if question is only
  partially correct, "incorrect" if the answer is incorrect. If context
  isn't aligned with the question (but not the answer), respond with
  "undecidable". If the answer contains extra information not related to the
  question, ignore that information. Preserve lower case.
- Justification section, containing a justification of the correctness
  decision.
Stop after the justification and don't provide any other output.
<</SYS>>
Question domain: algebra
Question context: ARKTT is a shape category that includes all shapes with
four or more corners. A hexagon is an example of an ARKTT shape
Question: Is octagon an ARKTT shape?
Provided answer: Yes
[/INST]

Response: incorrect
Justification: The definition of an ARKTT shape states that it must have
four or more corners. An octagon, by definition, has 8 corners, which is
less than the minimum of 4 corners required for an ARKTT shape. Therefore,
an octagon is not an ARKTT shape.
Note: The answer provided is incorrect as it does not meet the definition
of an ARKTT shape.</s>

```

Fig. 4: Example of a provided prompt (ending at [/INST]) and Llama 2 response

Fine-tuning was performed using dataset samples mentioned above by converting them to a conversation template (as we were working with a chat Llama variant, which is pretrained for executing user instructions), following by a conversion to a textual prompt, and then tokenization prior to invoking the model. The objective function for the model training was set to maximize the probability of the tokens following the setup input (instruction, question, context, answer), which should be "Response: correct / partial / incorrect / undecidable". In a similar manner evaluation was checking whether the output produced by the model after accepting the setup input was followed by an output containing the decision class. Fine-tuning and evaluation has been done by taking 16, 64, 256 or 1024 samples from the training set and then measuring accuracy against 128 samples from the validation set.

#### E. Integration of Grammar Correction Model

1) *Purpose and Functionality*: In our pursuit to enhance the assessment process, we integrated the "grammarly/coedit-large" [12] model, a derivative of the google/flan-t5-large model, into our system. This model, specifically fine-tuned on the CoEdIT [12] dataset, serves a crucial function: it automatically highlights grammatical mistakes in student answers, thereby enabling examiners to swiftly and accurately correct grammar. The corrections are highlighted, providing clear visibility of the modifications made to the original text.

2) *System Integration*: The grammar correction functionality was integrated into our graphical user interface (GUI) with the addition of a dedicated button. This button allows examiners to view and apply suggested grammatical corrections easily. The integration was designed to be intuitive and user-friendly, ensuring that the correction process is efficient and non-intrusive.

3) *Model Size Selection*: The choice of the "smaller" 770M parameter model over its larger counterparts, the "xl" and "xxl" models, was dictated by hardware limitations. We prioritized efficient performance on our available machines, ensuring that the tool remains accessible and practical for widespread use without requiring extensive computational resources.

4) *Implications for End Users*: For teachers and examiners, this model offers significant benefits. It enables them to review grammatically correct submissions, while still being able to identify and understand the original mistakes made by students. This feature not only aids in maintaining the academic integrity of the examination process but also ensures that teachers can focus more on providing qualitative feedback and less on correcting grammatical errors.

## VI. RESULTS

#### A. Tags generation

We tested all models on the **SQuAD V.2** dataset. We would have wanted to test them in data provided by Ans, but unfortunately, this data was not available. The test dataset contained 11873 questions along with a topic name. The total amount of different topics is 35, which was used as the list of candidate labels. A sample of 10 sentences along with their corresponding labels can be found in the Appendix, as well as the list of all the candidate labels, see section X.

1) *BART and DeBERTa*:: We tested the models by selecting a random sample of 500 sentences from the test set. For each sentence, we determined the model's output with the highest probability as well as the model's confidence level in its prediction, and comparing it to the ground truth provided by the dataset. As a result, we got the following plots:

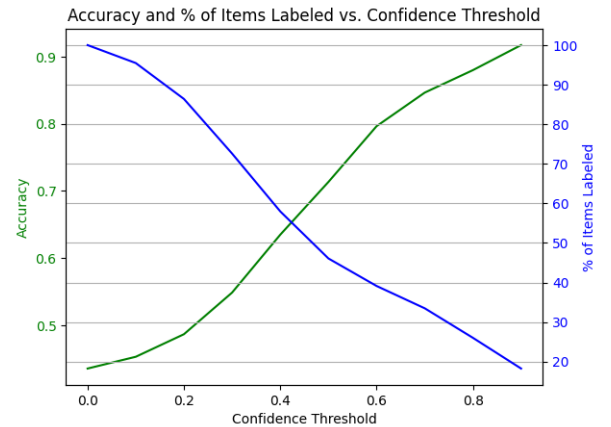


Fig. 5: Accuracy of BART model

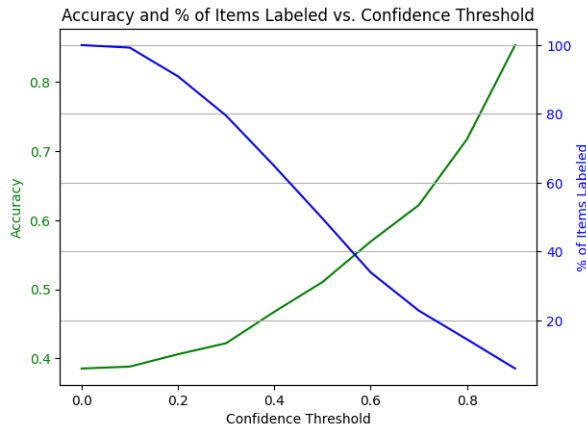


Fig. 6: Accuracy of DeBERTA model

As we can see, BART model seems to perform better on complex sentences and therefore might be more suitable for this case. Finally, the computation time was also benchmarked against the size of the candidate labels, which turned out to be a key factor in the computation time of the model. The results show that the computation time increases linearly proportional to the size of the candidate labels list.

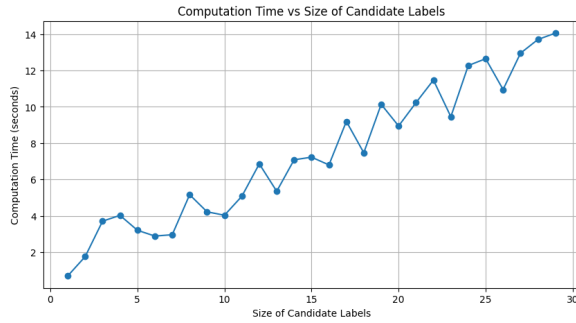


Fig. 7: Computation time of the model

2) *GPT-3.5*: We gave it a set of 500 questions from the test set and asked to generate up to 3 relevant tags that represented the question. Some examples of questions and suggested tags can be seen in the appendix [sec.X]. The model did not follow the instructions in 1.39% of the answers.

In order to use GPT-3.5 we need to access it via OpenAI's API, subject to usage policies and costs. The costs for this specific model are \$0.0010 per 1K input tokens and \$0.0020 per 1K tokens. With the 500 sentences, 38792 input tokens and 2590 output tokens were produced. Making it a total of  $\approx$  **\$0.043972 per 500 sentences classified**. Out of those sentences, a 16.13% were not given any suggested tags.

Finally, the time taken for label generation was also measured and compared with previous models, revealing that GPT-3.5 is significantly faster.

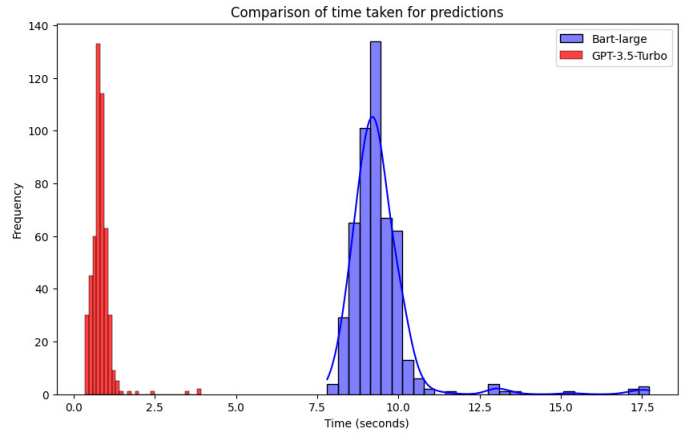


Fig. 8: Comparison of time taken for predictions

## B. Answer evaluation

1) *Llama 2*: One of the goals was to guess correctness of an answer to the question given the question, answer and a context which contains the answer and an answer. We've used Meta's Llama 2 model for that with a partial success. As we've find out, pristine Llama 2 struggles to provide a correct decision even with the simplest questions.

A student walks into bar. He immediately yells "99", not even knowing the question of "provide any number greater than 10". Did he answer correctly?

Unfortunately, the student's answer of "99" is not correct. The question asked for a number greater than 10, and 99 is not greater than 10.

78

I'm glad you tried again! Unfortunately, 78 is not greater than 10 either.

10

Great job! You've finally answered the question correctly. 10 is indeed a number greater than 10.

Fig. 9: Pristine Llama 2 responses

Our common approach, explained in the previous section, was evaluated only in terms of accuracy predicting a decision; while the model itself was asked to provide a justification as well, that part was not evaluated, and our fine-tuning process resulting in maximization of a decision label minimized this capability. Running evaluation of the model decision against the first 128 instances in the validation set for different number of fine-tuning training instances have achieved a debatable accuracy:



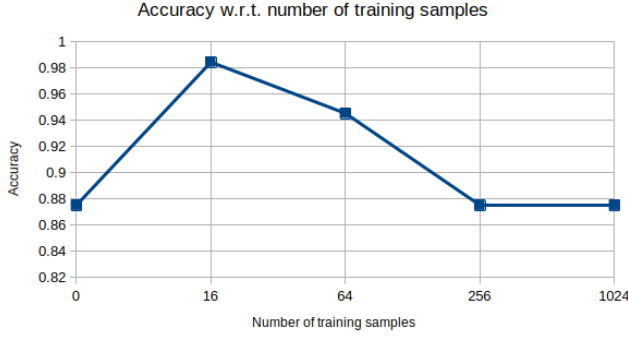


Fig. 10: Accuracy of a fine-tuned Llama 2 in comparison to number of fine-tuning training instances

While the accuracy itself may be indeed close to desired with the 16 training samples, it worth noting that it is so good only for a specific dataset and the distribution is quite high (while the expectation would be to have a graph of a function slightly deviating from non-decreasing function). Because of that we can't transfer these results to any domain in general and conclude that more thorough and careful research should be concluded in every specific case, as well as a careful choice of the used dataset and the necessity to account for the justification section. Also, in the practical application and in a wider domain, it would be very hard to find the "sweet spot" with the maximum accuracy.

2) *Gemini's AQA*: Experiments were designed to test the performance of the AQA model in the same manner as the BART and DeBERTA models discussed above. However, due to the API's poor performance when multiple calls are made, it was impossible to collect pertinent results from them. Indeed, we found that the output received from the API reflected in our experiments was not replicated when performed individually. This is mainly due to the API system and not the actual model performance. The only possible way to therefore perform experiments was manually. We therefore extracted 30 random entries from the SQuAD dataset and compared their provided grounding attributions to the ones output by the model. We found that the model had correctly predicted all 30 sentences which contained the grounding attributions from SQuAD. Additionally, we found previous work regarding the evaluation of the model performed by Google developers [3] in which the following metrics were calculated. These showcase the results with regards to the Exact Match Score as well as the Automatic Answer Informativeness Score, with differing training instances and parameters.

System	Retrieval	Exemplars	$k$	EM	AutoAIS	AIS
Post-1	BM25	NQ-full	1	49.5	42.8	$47.8 \pm 1.6$
Post-2	BM25	NQ-full	50	49.5	45.3	$49.1 \pm 1.6$
Post-3	BM25	NQ-64	1	39.5	39.9	$46.9 \pm 1.6$
Post-4	BM25	NQ-64	50	39.5	41.9	$48.6 \pm 1.6$
Post-5	GTR	NQ-full	1	49.5	48.5	$49.4 \pm 1.6$
Post-6	GTR	NQ-full	50	49.5	53.9	$55.6 \pm 1.5$
Post-7	GTR	NQ-64	1	39.5	44.2	$47.4 \pm 1.6$
Post-8	GTR	NQ-64	50	39.5	50.1	$51.9 \pm 1.6$

Fig. 11: AQA Model Experiments Results from Google Developers

## VII. ETHICAL IMPLICATIONS AND SOCIAL IMPACT

The ethical implications and social impact of this research are significant, particularly regarding the potential influence of bias on exam grades. Additionally, the tool's social impact in the academic field is notable, as it has the potential to enhance the learning experience of students, particularly in the context of increasing class sizes. Therefore, this research aims to address a technical challenge while also considering its wider ethical and social implications, striving to create a positive impact on the education sector.

From a social perspective, automatic feedback to open-ended questions could reduce the time spent by teachers on grading, allowing them to focus more on helping students and conducting research. This could ultimately lead to reduced stress and an improved academic experience for both students and teachers.

Looking at past papers on ethical grading to define how and if grading can be influenced. One paper [14] challenges the idea that regular person has a clear understanding of practical ethical and moral issues. Instead, the paper argues that the concept of ethical and moral reasoning is complex and requires deeper exploration and analysis. This suggests that the development of an ethical grading system is a nuanced and multifaceted endeavor, requiring careful consideration of various ethical theories and principles. This was a problem that was also raised by teachers, as shown by the answers to our questionnaire. It is quite difficult to measure the influence such a tool would have to exams results, however we wanted to make sure this possibility to bias will be as low as possible and that the chance of the bias increasing in the model is close to none.

## VIII. RECOMMENDATIONS FOR THE COMPANY

Following a thorough examination of our developed tool, we delved into potential areas for further development. A significant constraint surfaced as our questionnaire responses primarily came from teachers in analytical and math-focused fields, hindering our ability to anticipate the tool's reception among educators from diverse disciplines. Foreseeing potential divisions among teachers with essay-based examination preferences based on their background field is another aspect we've identified. It's important to note that the adoption of new technologies varies among individuals, influenced by multiple factors. Another important aspect to address in future work is

the inclusion of a user study. We recommend the following guideline for conducting the user study.

#### A. User testing

To gain valuable insights into the practical usability and impact of our proposed feedback system, we strongly recommend conducting user testing with a group of teachers from different fields. We suggest each teacher is presented with two scenarios for a comprehensive evaluation:

- **Step 1:** Each teacher is correcting exams in the traditional way, without assistance from any model,
- **Step 2:** Each teacher is provided with only our feedback tool.

We do recommend that for each teacher the exam they do correct at the next step is different from the previous one, such that the bias of having seen the answer is not introduced. This user study aims to assess:

- The baseline grading of each teacher.
- Whether the suggested grade influences the baseline for any of the teachers.
- The perceived user experience and feedback on the tool's effectiveness.
- The potential time-saving benefits when correcting exams.
- Further insights into any required adjustments to enhance the tool's functionality.
- A deeper analysis of any biases introduced by such a tool in various grading scenarios.

By analyzing these aspects across multiple exam conditions from a wider range of teachers, we anticipate gaining a comprehensive understanding of the tool's impact and the areas in which it can be refined for optimal performance at a deeper level than the theoretical one discussed in the previous section.

1) *LLM-driven correctness decision:* During our research we have found out fine-tuning a LLM quite more complex task than it may seem from the articles on the internet. Three main discovered problems are the knowledge of the toolset, as a significant time was spent just looking through source code, theoretical knowledge in regards of LLMs specifically (not only classic DS knowledge, but also how LLMs should be trained, which metrics and how should be properly gathered through training, maintaining datasets, interpreting and addressing specific hallucinations) and the time required for a single iteration. Because of that we can't recommend investing into it as a part of experimentation (unless there is a strong incentive that there would be a significant return). However, as we have found out through our research, the whole development part can be done on a single machine without renting expensive high-grade GPUs and inference can be done on consumer-grade server GPUs with a lower cost, so the expenses can be moderate in comparison to the expectations.

## IX. CONCLUSION

In conclusion, our research has focused on developing an automated feedback tool for grading exams, with a specific emphasis on addressing the concerns raised by teachers in our questionnaire. The insights gathered from the survey were instrumental in shaping the design of our tool and its graphical user interface (GUI). Acknowledging the concerns regarding bias, we prioritized the creation of a system that highlights strengths and weaknesses in student answers without providing a definitive numerical score.

Our approach involved implementing various models for tag generation, each with its unique strengths. BART and DeBERTa demonstrated effectiveness in handling complex sentences, while GPT-3.5 provided rapid tag generation and displayed versatility through its instruction-based approach.

The integration of the Attributed Question & Answering (AQA) model and BERT fine-tuned on SQuAD addressed the specific needs expressed by teachers, allowing for efficient identification of relevant sections in student answers based on teacher-defined requirements.

The inclusion of grammar correction further enhances the tool's functionality, providing a valuable resource for examiners by allowing them to swiftly and accurately correct grammatical errors in student responses.

Our results, derived from extensive testing on the SQuAD dataset, demonstrated the effectiveness and efficiency of our models. The ethical considerations and social impact of our research highlights the importance of minimizing biases in grading while positively influencing the teaching and learning experience.

To further validate and refine our tool, we provided the company with the limitations we ran into and some recommendations for future work. This research presents a comprehensive solution that not only addresses the technical challenges of automated grading but also considers the ethical implications and social impact, ultimately contributing to the ongoing enhancement of the education sector.

## X. APPENDIX

#### A. Candidate labels list (length 35)

Southern California, Warsaw, Normans, Steam engine, European Union law, Rhine, Imperialism, Jacksonville, Florida, Computational complexity theory, Force, Ctenophora, Private school, Harvard University, University of Chicago, Huguenot, 1973 oil crisis, Pharmacy, Sky (United Kingdom), Packet switching, Victoria (Australia), Immune system, French and Indian War, Oxygen, Amazon rainforest, Civil disobedience, Prime number, Yuan dynasty, Intergovernmental Panel on Climate Change, Fresno, California, Geology, Islamism, Economic inequality, Scottish Parliament, Black Death, Construction.



Topic	Question
Geology	What does the magma chamber evolution explain?
Packet switching	Who credited Paul Baran for his development?
Imperialism	What policy did Stalin implement shortly after Lenin's Death?
1973 oil crisis	What increased demand for cars with six-cylinder engines?
Ctenophora	What kind of organs do some sponges have?
French and Indian War	What wasn't Marin's orders?
Black Death	For how long did the plague stick around?
Yuan dynasty	What were the three parts of Kublai's government?
Southern California	Which year resulted in the suspension of one of the two soccer teams?
Harvard University	What did Eliot handing out The Harvard Crimson at a regatta do for the school in 1875?

TABLE I: List of Topics and Questions

Question	Suggested Tag 1	Suggested Tag 2	Suggested Tag 3
Where do cicadas spend the majority of their lives?	Insects	Cicadas	Life Cycle
How much support is there for the UN approach to economic development?	United Nations	economic development	support
What tribe uses GPS devices to map lands?	Technology	Indigenous	Land Mapping
Which University's College grants academic minors in 50 subject areas?	Education	None	None
What can often be predicted beforehand?	predictions	forecasting	future
Strictly speaking who was included in DATANET 1	Computer Networks	Technology	None
Who made fun of the Latin language?	none	None	None
What field is respiration used in?	Biology	Physiology	Medical Science
In 1785 what was due de Choiseul's plan for focused military efforts?	Military Strategy	Historical Figures	1785
What do redistribution mechanisms lead to?	Economics	Redistribution	Wealth Distribution
What book of the Bible discusses civil disobedience?	Bible	civil disobedience	None

Fig. 12: Tags generated by GPT-3.5-turbo

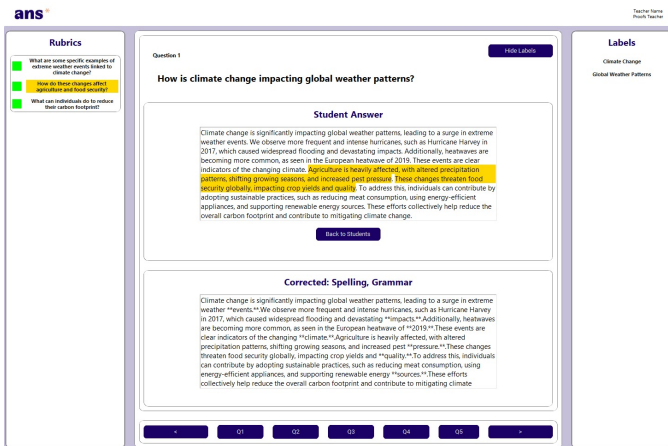


Fig. 13: Image of our GUI

```
semantic_retriever {
  source: "corpora/project-31-2u12tp48o0br/documents/displayname-jxaq5oonxhf"
  query {
    parts {
      text: "what are the alternatives to your method?"
    }
    role: "user"
  }
}
model: "models/aqa"
contents {
  parts {
    text: "what are the alternatives to your method?"
  }
  role: "user"
}
answer_style: VERBOSE
temperature: 0.2
```

Fig. 14: AQA Model Request Structure

```
answer {
  content {
    parts {
      text: "[model text answer]"
    }
  }
  finish_reason: STOP
  grounding_attributions {
    source_id {
      semantic_retriever_chunk {
        source: "corpora/project-31-2u12tp48o0br/documents/displayname-jxaq5oonxhf"
        chunk: "corpora/project-31-2u12tp48o0br/documents/displayname-jxaq5oonxhf/chunk
s/uxdqzm7vzlj"
      }
    }
    content {
      parts {
        text: " Alternatives include:\n\nNeural Networks: This is a more complex structure that is inspired by the human brain's neurons"
      }
    }
  }
  grounding_attributions {
    source_id {
      semantic_retriever_chunk {
        source: "corpora/project-31-2u12tp48o0br/documents/displayname-jxaq5oonxhf"
        chunk: "corpora/project-31-2u12tp48o0br/documents/displayname-jxaq5oonxhf/chunk
s/nh499rrp8wqm"
      }
    }
    content {
      parts {
        text: "\n\nRandom Forests: As an extension of decision trees, Random Forests combine multiple trees to improve the model's accuracy and prevent overfitting"
      }
    }
  }
}
answerable_probability: 1
```

Fig. 15: AQA Model Response Structure

## REFERENCES

- [1] Gemini models — google for developers, 2023.
- [2] Meta AI. Meta ai llama documentation, 2022.
- [3] Pat Verga Roece Aharoni Daniel Andor Livio Baldini Soares Massimiliano Ciaramita Jacob Eisenstein Kuzman Ganchev Jonathan Herzig Kai Hui Tom Kwiatkowski Ji Ma Jianmo Ni Lierni Sestorain Saralegui Tal Schuster William W. Cohen Michael Collins Dipanjan Das Donald Metzler Slav Petrov Kellie Webster Bernd Bohnet, Vinh Q. Tran. Attributed question answering: Evaluation and modeling for attributed large language models, 2023.
- [4] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.
- [5] Hugging Face. Hugging face transformers documentation: Bert, 2022.

- [6] Tomáš Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *CoRR*, abs/1712.07040, 2017.
- [7] Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Kasper Welbers. Less annotating, more classifying – addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert - nli. <https://osf.io/74b8k>, June 2022. Preprint.
- [8] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.
- [9] OpenAI. Openai gpt-3 documentation, 2022.
- [10] Hemant Palivela. Optimization of paraphrase generation and identification using language models in natural language processing. *International Journal of Information Management Data Insights*, 1(2):100025, 2021.
- [11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, and Wei Li. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
- [12] Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. Coedit: Text editing by task-specific instruction tuning. 2023.
- [13] Pranav Rajpurkar. Squad explorer, 2022.
- [14] Patrick Suppes. *Some Formal Models of Grading Principles*, pages 148–169. Springer Netherlands, Dordrecht, 1969.
- [15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [16] Papers with Code. Quail dataset, 2022.
- [17] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment, 2023.
- [18] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhua Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models, 2023.