GRANTEE SUBMISSION REQUIRED FIELDS

Title of article, paper, or other content

Automated Assessment of Comprehension Strategies from Self-Explanations using LLMs

All author name(s) and affiliations on PDF. If more than 6 names, ERIC will complete the list from the submitted PDF.

Last Name, First Name	Academic/Organizational Affiliation	ORCID ID
Nicula, Bogdan	National University of Science and Technology	
Dascalu, Mihai	National University of Science and Technology	
Arner, Tracy	Arizona State University	
Balyan, Renu	SUNY Old Westbury	
McNamara, Danielle S.	Arizona State University	

Publication/Completion Date—(if *In Press,* enter year accepted or completed) 14 October 2023

Check type of content being submitted and complete one of the following in the box below:

☑ If article: Name of journal, volume, and issue number if available
□ If paper: Name of conference, date of conference, and place of conference
□ If book chapter: Title of book, page range, publisher name and location
□ If book: Publisher name and location
□ If dissertation: Name of institution, type of degree, and department granting degree

Information, 14(10)

DOI or URL to published work (if available) https://doi.org/10.3390/info14100567

Acknowledgement of Funding— Grantees should check with their grant officer for the preferred wording to acknowledge funding. If the grant officer does not have a preference, grantees can use this suggested wording (adjust wording if multiple grants are to be acknowledged). Fill in Department of Education funding office, grant number, and name of grant recipient institution or organization.

"This work was supported by U.S. Department of Education **[Office name]** National center for Educati through **[Grant number]** Arizona Stat to **Institution]** R305A130124. The opinions expressed are those of the authors and do not represent views of the **[Office name]** National Center for Educa or the U.S. Department of Education.

Article

Automated Assessment of Comprehension Strategies from Self-explanations using LLMs

Bogdan Nicula [1], Mihai Dascalu [1,2,*], Tracy Arner [3], Renu Balyan [4], and Danielle S. McNamara [3]

[1] University Politehnica of Bucharest, 313 Splaiul Independentei, 060042, Bucharest, Romania; bogdan.nicula@upb.ro (B.N.), mihai.dascalu@upb.ro (M.D.)
[2] Academy of Romanian Scientists, Str. Ilfov, Nr. 3, 050044, Bucharest, Romania;
[3] Arizona State University, Department of Psychology, PO Box 871104, Tempe, AZ 85287; tarner@asu.edu (T.A.), dsmcnama@asu.edu (D.M.)
[4] SUNY at Old Westbury, Math/CIS Department, Old Westbury, NY - 11568; balyanr@oldwestbury.edu (R.B.)
* Correspondence: mihai.dascalu@upb.ro

Abstract: Text comprehension is an essential skill in today's information-rich world, and self-explanation practice helps students improve their understanding of complex texts. This study is centered on leveraging open-source Large Language Models (LLMs), more specifically FLAN-T5, to automatically assess the comprehension strategies employed by readers while understanding STEM texts. The experiments rely on a corpus of 3 datasets (N = 11,833) with self-explanations annotated on 4 dimensions: 3 comprehension strategies (i.e., bridging, elaboration, and paraphrasing) and overall quality. Results improved with fine-tuning, using a larger LLM model, and providing examples via the prompt. Besides FLAN-T5, we have also considered GPT3.5-turbo to establish a stronger baseline. Our best model considers a pre-trained FLAN-T5 XXL model and obtained a weighted F1-score of 0.721, surpassing the 0.699 F1-score previously obtained using smaller models (i.e., RoBERTa).

Keywords: Language Models, Large Language Models, Self-explanation, Self-explanation Strategies

Citation: Nicula, B.; Dascalu, M.; Arner T.; Balyan R.; McNamara, D. Title. *Information* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

Reading and learning from text is a critical skill for learners to acquire new knowledge which is essential for educational and career success. To comprehend text, the reader constructs a mental model of the text while they read. This mental model can be represented at three levels: 1) surface-level knowledge of the exact words in the text, 2) textbase-level semantic representation of ideas, and 3) the situation model that combines the textbase with the reader's prior knowledge. The ability to leverage strategies that support comprehension is a critical skill that readers need in the absence of the essential prior knowledge necessary to develop a coherent situation model. Proficient readers are more likely to spontaneously employ strategies while reading to help them comprehend difficult texts than students who are less skilled readers [1]. Fortunately, students can learn when and how to implement these reading comprehension strategies through direct instruction and deliberate practice. One such strategy, with considerable evidence supporting its use by students with limited prior knowledge or lower reading skills, is self-explanation.

Self-explanation (SE) is the practice of explaining the meaning of portions of a text to oneself while reading. Engaging in self-explanation encourages students to generate inferences, in which they connect sentences or idea units between sections of the text or between texts. Similarly, students may generate elaborative self-explanations in which they connect their own prior knowledge to new information they read in the text. Generating bridging and elaborative self-explanations supports readers' inference making, which, in turn, supports the development of their mental representation of the text.

Developed by McNamara [2], Self-explanation Reading Training (SERT) teaches readers the strategies they can use to enhance text comprehension. The training guides students through each strategy in increasing order of difficulty, starting with comprehension monitoring. The purpose of comprehension monitoring is to help students understand when they need to implement the remaining strategies to support their comprehension. This work focuses on the three remaining strategies: paraphrasing, bridging inference, and elaboration. Paraphrasing refers to reformulating a sequence of text in one's own words. SERT can help develop readers' text comprehension skills by forcing them to access their vocabulary to translate the ideas into a more familiar language. Bridging involves linking multiple ideas across a text or across multiple texts (e.g., two different articles about the same topic). Generating bridging inferences requires the reader to find connections between the ideas and to structure them in a coherent way; this serves to support the development of their mental representation of the text. Elaboration involves linking information in the text and the reader's knowledge base; this helps the reader integrate new information with existing knowledge.

Considerable evidence indicates that these strategies support readers' comprehension of complex texts. However, additional benefit can be realized when the reader receives feedback about the accuracy or quality of their self-explanation [3]. One way readers can receive feedback is from instructors who review and score self-explanations based on a rubric [4]. This method is time-consuming and does not provide readers with the feedback they need in real time. To alleviate this challenge, students can practice their reading and self-explaining using an intelligent tutoring system where they both have the opportunity to engage in deliberate practice of reading and self-explaining, but they also receive essential guiding feedback [5]. Thus, refining and improving software applications that can detect the presence of these strategies in the readers' productions can be helpful for both evaluation and training. Natural Language Processing (NLP) [6] techniques and Machine Learning can be used to develop such models, given a large enough dataset containing labeled examples of the presence and absence of these strategies in readers' self-explanations. Previous work [7] has shown that such automated models can be built to reliably assess self-explanation reading strategies. The recent release of more sophisticated and readily accessible large language models further supports the expansion of this previous work.

1.1. Large Language Models

Chatbots have become increasingly prevalent in various domains, including customer service, social media, and entertainment. Their popularity increased even further with the launch of ChatGPT last year, followed shortly by other competitors such as Google Bard. The key innovation that accelerated the adoption of chatbots in multiple fields is the development of large language models (LLMs). These models are trained on massive amounts of heterogeneous text data (including news articles, web pages, social media posts, and scanned books) and datasets tailored to specific tasks.

These models capture statistical patterns of natural language, such as syntax, semantics, and pragmatics. Their knowledge of these patterns enables the generation of new complex texts relevant to the input they have been prompted with. LLMs are highly adaptable to different NLP tasks and domains and can be fine-tuned on specific data sets or prompts to perform a wide variety of natural language generation tasks, including summarizing, translation, text completion, and question answering. They also manifest "emergent capabilities" [8], skills they were not trained explicitly on but are easy to solve based on the memorized statistical patterns.

However, LLMs' impressive capability to generate various relevant, cohesive, and coherent texts comes with caveats. These models can sample from the most statistically relevant sequences and complete a given prompt flawlessly. Still, they do not offer guarantees regarding the correctness of the generated information [9]. Furthermore, they are still susceptible to a variety of attacks, such as injecting a request with a small sequence of words that can deviate the flow of the interaction in a different direction from what was intended initially [10].

LLMs, the backbone of such systems, are a fairly recent type of neural architecture that has grown in size and performance in the past few years. They are part of a class of deep learning architectures called Transformers, stemming from the original model introduced by Google in 2017 [11]. Depending on their structure, modern Transformer-based models can be classified into 3 categories:

- **Encoder only.** Models which understand the text and are used in classification/regression tasks. An example of an encoder-only model is the BERT [12] model, followed by its improved version RoBERTa [13].
- **Decoder only.** Models that excel at text generation. The GPT (Generative Pretrained Transformer) family with various versions (e.g., 3 [14] or 4 [15]) are good examples of the Decoder-only architecture.
- **Encoder-decoder.**: Models capable of both understanding and generating text. They are useful for translation, abstractive summarization, question answering, and many other tasks. The Text-to-Text Transformer (T5) [16], followed by its improved version FLAN-T5 [17] pre-trained on a large collection of datasets, are examples of such an architecture.

1.1.1. FLAN T5

The T5 model is an encoder-decoder Transformer trained on a combination of supervised and unsupervised tasks, all having a text-to-text format (i.e., receiving text input and outputting text). The supervised training is done on tasks from the GLUE [18] and SuperGLUE [19] benchmarks converted to fit the text-to-text paradigm. The unsupervised or self-supervised tasks involve reconstructing the original text when receiving corrupted input (e.g., by randomly removing 15% of tokens and replacing them with sentinel tokens). The T5 models that have been made public cover a wide range of sizes, from the 60 million parameters t5-small model to the 11 billion t5-11b model.

The FLAN-T5 model [17] represents an enhanced version of T5 fine-tuned on a larger number of tasks while emphasizing chain-of-thought scenarios. Using the FLAN approach, the authors trained both T5 and PaLM [20] models and achieved state-of-the-art performance on several benchmarks with the 540 billion FLAN-PaLM model.

1.1.2. GPT3.5-turbo

The GPT-3 model was perceived as a considerable step when released in 2020 in terms of performance and size (175 billion parameters). The GPT3.5-turbo model was released in November 2022, with OpenAI providing scarce information regarding its training. Its size is estimated to be comparable to that of GPT3.5. The model was trained using Reinforcement Learning from Human Feedback (RLHF) [21] and is designed to perform better in conversational settings and iterative task-solving. It gained popularity as it represented the backbone of the popular free version of the ChatGPT conversational agent.

Besides GPT3.5-turbo, OpenAI provides several other text-to-text endpoints such as 'text-davinci-003' and GPT4 [15]. The former was tested on a subset of the tasks presented in this article but did not perform better than the GPT3.5-turbo endpoint. The latter, GPT4, is the backbone of the paid ChatGPT Plus service and is expected to provide better replies, but it also is a more closed system with little detail being provided about its architecture, the training dataset, and or the training setup. We opted against evaluating this alternative as at the time the experiments were done, it was 20-30x more expensive than GPT3.5-turbo, and our aim was to create an open-source model.

Table 1 displays the size of the models that were taken into consideration for this study. The FLAN small and base models were useful for fast initial experimentation, but they are not featured in the results section as their small size doesn't provide the models enough expressiveness to perform well on these tasks.

Table 1. FLAN and GPT3.5 model sizes

Name	Type	Size
FLAN	small	60M
FLAN	base	250M
FLAN	large	780M
FLAN	XL	3B
FLAN	XXL	11B
GPT3.5-turbo	-	150B-175B

Current Study Objective

The overarching objective of this study is to develop an automated model for evaluating the comprehension strategies (paraphrasing, elaboration, and bridging) employed by readers and the overall quality of the produced self-explanations.

This study is focused on evaluating the extent to which open-source Large Language Models (LLMs) can be leveraged to build such an automated system. The results are compared to the performance of previous methods, which relied on smaller and less resource-intensive machine learning models [7]. We also analyzed how the performance of these LLM models scales with model and prompt size. We have a side-by-side comparison between open-sourced models and the OpenAI API used as the backbone of the popular ChatGPT. We release our best model on HuggingFace and the corresponding code on GitHub: https://github.com/readerbench/self-explanations.

2. Method

2.1. Corpus

The corpus used in is study consists of three datasets containing 11,833 annotated self-explanations [22]. The datasets were collected from high school and undergraduate students who were asked to read one or two science texts and generate self-explanations for nine to 16 target sentences. An entry consists of the target sentence, a self-explanation, and categorical scores for paraphrase presence, bridging, elaboration, and overall self-explanation quality.

The corpus was split into train/dev/test using a ratio of 54.5%/27.5%/18%. The categorical scores for the four tasks ranged from 0 to 2 or 3. The problem of predicting these scores was modeled as a classification task, with each score representing a class. The values are codified consistently across tasks so that class 0 always represents low-quality self-explanations or the absence of a particular strategy. In contrast, higher values represent self-explanations of higher quality. In the case of bridge presence and elaboration presence, the final 2 classes containing higher-quality examples were merged to reduce the class imbalance. After these changes, the elaboration presence task had 2 classes, paraphrase, and bridge presence had 3 classes, whereas the overall quality task had 4 classes.

2.2. LLM Prompting

The format of the prompt (i.e., input text for the LLM) can influence the quality of the provided answer [23]. Therefore, we tried to structure the input similarly to how input was structured for the tasks on which the initial FLAN-T5 model was trained. Additionally, we experimented with adding a "System role" entry at the beginning of the prompt for the requests made to GPT3.5-turbo, as suggested by the OpenAI GPT3.5-turbo API documentation [**?**]. The "Context" section provides additional descriptive information regarding the task to be solved.

Both the FLAN-T5 models and the GPT3.5-turbo API were queried in 0-shot, 1-shot, and multi-shot settings to evaluate how examples can assist the model in providing better answers. In the multi-shot setting, the model was provided one example per class, selected from the training set. This was feasible because the tasks had a maximum of four classes.

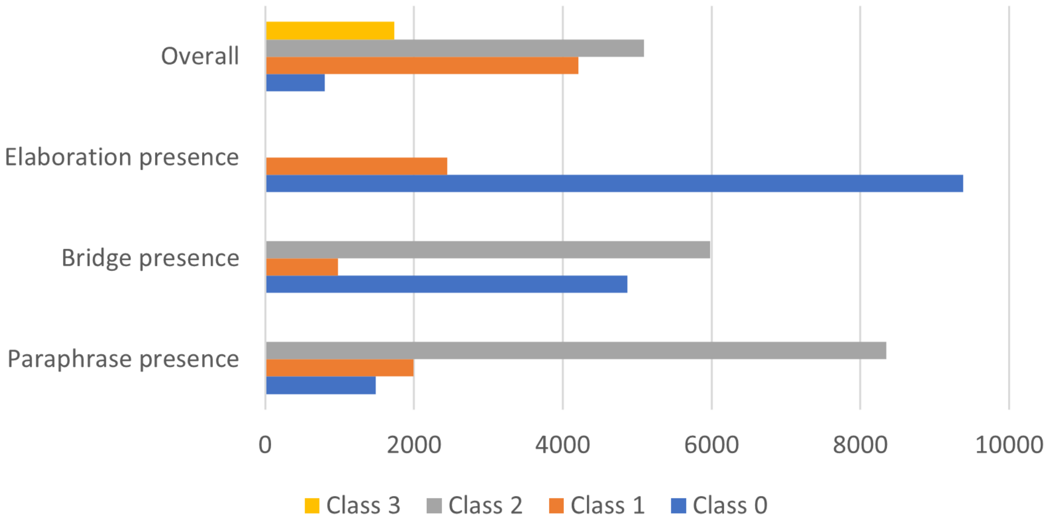


Figure 1. Class distribution per task.

The "Target question" section contains the question that the model must answer. Since answering the question involves reading the generated self-explanation and the source sentence, we added them to this section and labeled them as "S1" and "S2". Preliminary experiments indicated that the models performed better when using these naming conventions rather than "Generated Sentence" and "Original Sentence" or other combinations.

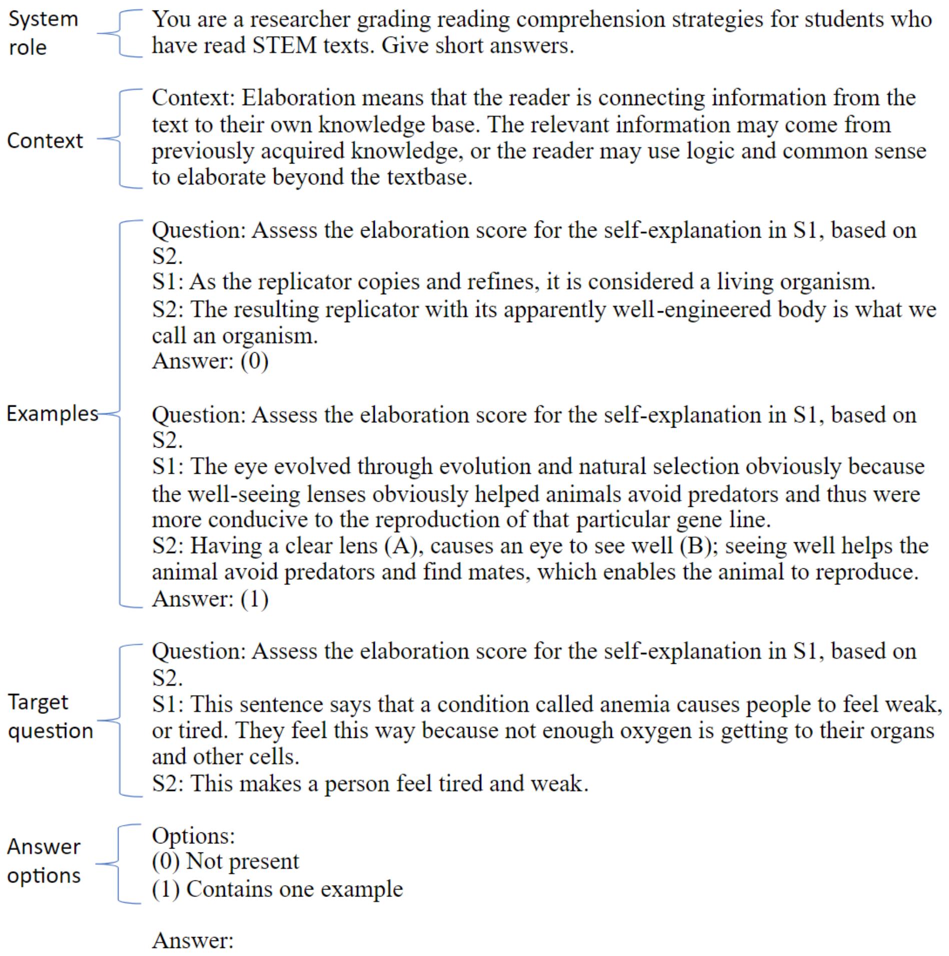
Lastly, the "Answer options" section lists the possible answers and a short description. Experiments were also performed with more detailed descriptions of the classes, but this only improved performance in the case of the GPT3.5-turbo experiments.

2.3. LLM Fine-tuning with LoRA

There are multiple methods of adapting pretrained LLMs to help them perform better on certain tasks. One such option is using the last set of hidden features that the model produces and training a small deep-learning model to predict the expected output based on the set of hidden features while freezing the updates for the LLM parameters. This is efficient in terms of resources but can add latency on inference because the depth of the model is increased. A second option consists of selective fine-tuning, in which only a subset of the LLM's layers are trained while the rest are kept unchanged. This approach can also be efficient, but it involves manually selecting which layers to train, an operation not necessarily intuitive. The third option consists of fine-tuning the entire model. Out of the three approaches, this should yield the best results, but it requires the most GPU memory and training resources.

Apart from the classical methods listed above, PEFT (Parameter Efficient Fine-Tuning) methods rely on training only a subset of the LLM parameters without manually selecting what parameters to train. One of the most popular PEFT techniques considered in our work is LoRA: Low-Rank Adaptation of Large Language Models [24]. LoRA efficiently fine-tunes LLMs by freezing the pretrained model and injecting trainable rank decomposition matrices into each layer. The authors claim that LoRA can reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times for a 175 billion parameter GPT-3 training. Furthermore, the method adds no extra inference latency.

The innovation that LoRA brings is the use of low-rank parametrized update matrices. In a classical fine-tuning setting for a weight matrix $W_0 \in \mathbb{R}^{d \times k}$, we would have an update after backpropagation equivalent with $W = W_0 + \Delta W$ with ΔW having the same dimensions as the pretrained matrix. LoRA considers the following decomposition: $\Delta W = BA$ with $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, with rank $r \ll min(d, k)$. The two low-ranked matrices, A and B, will be trainable throughout the run while W_0 is frozen and initialized so that the


System role
Context
Examples
Target question
Answer options

You are a researcher grading reading comprehension strategies for students who have read STEM texts. Give short answers.

Context: Elaboration means that the reader is connecting information from the text to their own knowledge base. The relevant information may come from previously acquired knowledge, or the reader may use logic and common sense to elaborate beyond the textbase.

Question: Assess the elaboration score for the self-explanation in S1, based on S2.
S1: As the replicator copies and refines, it is considered a living organism.
S2: The resulting replicator with its apparently well-engineered body is what we call an organism.
Answer: (0)

Question: Assess the elaboration score for the self-explanation in S1, based on S2.
S1: The eye evolved through evolution and natural selection obviously because the well-seeing lenses obviously helped animals avoid predators and thus were more conducive to the reproduction of that particular gene line.
S2: Having a clear lens (A), causes an eye to see well (B); seeing well helps the animal avoid predators and find mates, which enables the animal to reproduce.
Answer: (1)

Question: Assess the elaboration score for the self-explanation in S1, based on S2.
S1: This sentence says that a condition called anemia causes people to feel weak, or tired. They feel this way because not enough oxygen is getting to their organs and other cells.
S2: This makes a person feel tired and weak.

Options:
(0) Not present
(1) Contains one example

Answer:

Figure 2. Example of prompt for the elaboration task in a multi-shot setting.

initial update matrix is 0. As such, LoRA was the best alternative when fine-tuning the FLAN-T5 models.

3. Results

In this section, we explore the extent to which the performances presented in previous studies [7] can be surpassed by employing out-of-the-box or fine-tuned LLMs. The input received by the models consisted of prompts like the ones described in section 2.2, which contained the student's self-explanation and the target sentence. Because of computational constraints, we skipped the scenarios in which the target sentence was omitted or was extended by including the previous sentences. The results obtained in the previous study by Nicula *et al.* [7] for overall quality also indicated that these two changes do not positively influence the results.

We used F1-score as the evaluation metric for the results. Because LLMs can generate incorrectly formatted answers, we will consider all badly formatted answers as belonging to class 0, which have been coded to contain low-quality examples. The percentage of correctly formatted answers will also be reported in order to understand how well the models have adapted to the task format.

We analyzed the percentage of correctly formed answers on the Overall task to observe how well the models adapt to the task format. The FLAN-T5 large and GPT3.5-turbo models conform to the expected format of answers (see Figure 3). Replies generated by FLAN-T5 XL and XXL versions improve (i.e., follow the correct format) when they are presented with more examples in the prompt. When looking at the output of the models, we observed that the FLAN-T5 XL and XXL models tend to provide more verbose replies, not necessarily incorrect, but do not match the expected format.

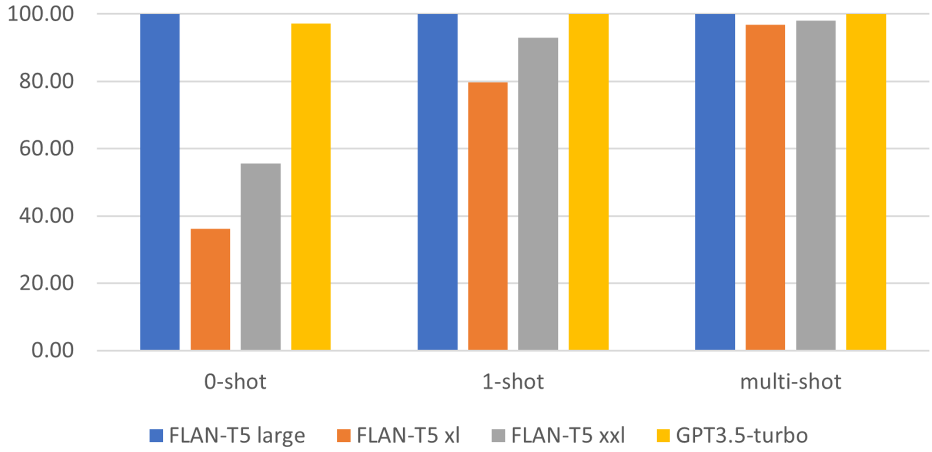


Figure 3. Percentage of correctly formed answers for the 'out-of-the-box' evaluation

The results are grouped into two sub-sections: the first subsection focuses on the 'out-of-the-box' performance of FLAN-T5 and GPT3.5-turbo, while the second subsection presents the performance of FLAN-T5 after fine-tuning using the LoRA method.

3.1. Out-of-the-box performance

In this section, the 'out-of-the-box' performance is assessed without fine-tuning. The models covered in this section are FLAN-T5 large, XL, and XXL models, along with the GPT3.5-turbo API. The same prompt structure was used for all FLAN-T5 models. This structure was chosen after a series of experiments evaluating how small changes in the prompt can affect the model's performance. The final version of the prompt for FLAN-T5 did not include a context section. Also, the prompt had shortened versions for the options in the Answer options section, labeled alphabetically instead of numerically (i.e., (A), (B), (C), (D), instead of (0), (1), (2), (3)). The prompt for the GPT3.5-turbo model had a context section, while the long answer options were labeled alphabetically.

The results for this task are presented in Table 2. For every task, the best result is listed using bold font. In some cases, multiple examples are bolded for a single task because the differences between the results are marginal.

The FLAN-based models performed considerably better than the GPT3.5-turbo model for the three comprehension strategy tasks. Differences of 51% (Paraphrase presence), 33% (Elaboration presence), and 10% (Bridging presence), in terms of weighted F1 scores, were observed between the best FLAN-T5 performance and the best GPT3.5-turbo results.

The model size did not have a large influence in the case of the FLAN-T5 models. The differences between FLAN-T5 large, XL, and XXL are unclear. The best results were obtained with a FLAN-T5 XL model for the Bridging and Elaboration presence tasks (see Table 2). The best performance on the Paraphrase presence task was a tie between FLAN-T5 large and XL, while FLAN-T5 large performed the best for the Overall task.

The impact of providing the model with more examples via the prompt is unclear for the FLAN-T5 models as no clear pattern can be observed in this regard. In the case of the GPT3.5-turbo model, the impact of adding examples is considerably clearer. We can observe that the results for the three comprehension strategy tasks improve when switching from 0-shot to 1-shot prompting and further on when switching to the multi-shot setting. However, the reverse happens for the Overall quality task. The GPT3.5-turbo model performs worse as more examples are added.

Further exploration was undertaken for the prompting format used to query the GPT3.5-turbo API in the multi-shot scenario. The endpoint was queried using more examples, the context section, and extended descriptions of the classes. Adding more

Table 2. Out-of-the-box results for FLAN-T5 and GPT3.5-turbo (the best results for each task are listed in bold).

Task	Model	Weighted F1-score 0-shot	 1-shot	 multi-shot
Paraphrase	FLAN-T5 large	**75.23%**	66.30%	68.73%
	FLAN-T5 XL	28.24%	53.93%	**75.26%**
	FLAN-T5 XXL	0.69%	1.53%	1.65%
	GPT3.5-turbo	2.19%	14.39%	24.74%
Elaboration	FLAN-T5 large	50.09%	58.78%	56.94%
	FLAN-T5 XL	**89.9%**	**89.87%**	**89.90%**
	FLAN-T5 XXL	87.79%	78.52%	80.11%
	GPT3.5-turbo	44.53%	55.38%	56.13%
Bridging	FLAN-T5 large	45.24%	45.39%	45.04%
	FLAN-T5 XL	34.30%	**51.61%**	44.34%
	FLAN-T5 XXL	23.06%	22.76%	22.85%
	GPT3.5-turbo	19.26%	32.80%	41.18%
Overall	FLAN-T5 large	27.97%	9.73%	7.09%
	FLAN-T5 XL	2.84%	7.24%	6.64%
	FLAN-T5 XXL	10.68%	8.81%	12.44%
	GPT3.5-turbo	**30.18%**	28.07%	27.87%

examples on top of the multi-shot setting did not help. However, adding the context and the extended descriptions slightly improved the results for some tasks. These prompt changes resulted in a slight improvement for the Overall and Bridging classes, considerable improvement for the Paraphrase class, and a high drop in performance for the Elaboration class. The results for the best-performing prompt are listed in Table 3. The results of these prompts are referenced when presenting the confusion matrices and the qualitative analysis for the GPT3.5-turbo model.

Table 3. GPT3.5-turbo performance after exploring prompt variations.

Task	Model	Weighted F1-score
Paraphrase	GPT3.5	65.54%
Elaboration	GPT3.5	6.80%
Bridging	GPT3.5	44.07%
Overall	GPT3.5	30.67%

3.2. Fine-tuning

In this section, we analyze the performance of fine-tuning FLAN-T5 models using the LoRA method. Experiments were run using the publicly available FLAN-T5 models on HuggingFace; similarly, the FLAN-T5 small and base versions were excluded in this subsequent analysis, given their poor performance.

Three FLAN-T5 models were initially trained using a small learning rate for one epoch on the four tasks in the 0-shot, 1-shot, and multi-shot settings. The same prompt structure as in the 'out-of-the-box' scenario was used. All models were trained using a mini-batch size of 1. Experiments with larger mini-batch sizes lead to poorer results, probably because the learning rate must be adapted depending on the batch size.

The performance for the Paraphrase, Bridging, and Overall quality tasks improves considerably when switching from 0-shot to 1-shot and then to multi-shot settings. In the case of Elaboration presence, the pattern is not as clear, but the best result is still obtained in a multi-shot setting. One exception is the performance of the FLAN-T5 XXL model on the Overall task, which required fine-tuning of the learning rate to get a good performance. The standard learning rate for the fine-tuning experiments was 3e-4, but this model obtained its best performance using 1.5e-4; most likely, the XXL model is more sensitive to the learning rate used for fine-tuning.

When looking at the impact of model size on performance, larger models tend to perform better for all the tasks. The best result for every task is obtained using a FLAN-T5 XXL model, and we can also observe that the FLAN-T5 XL outperforms the large variant in most scenarios.

Table 4. 1-epoch fine-tuned results for FLAN-T5.

Task	Model	Weighted F1-score		
		0-shot	**1-shot**	**multi-shot**
Paraphrase	FLAN-T5 large	21.45%	68.46%	82.53%
	FLAN-T5 XL	10.63%	37.54%	85.50%
	FLAN-T5 XXL	72.79%	74.98%	**86.76%**
Elaboration	FLAN-T5 large	83.99%	84.12%	74.26%
	FLAN-T5 XL	87.66%	81.58%	84.28%
	FLAN-T5 XXL	88.64%	88.63%	**89.80%**
Bridging	FLAN-T5 large	42.68%	45.32%	45.85%
	FLAN-T5 XL	24.37%	48.22%	61.26%
	FLAN-T5 XXL	53.13%	76.32%	**79.06%**
Overall	FLAN-T5 large	1.34%	2.15%	36.22%
	FLAN-T5 XL	11.53%	25.70%	40.02%
	FLAN-T5 XXL	59.68%	**64.39%**	61.25% [1]

[1] Obtained after extra hyper-parameter tuning.

As previously mentioned, the models were evaluated in a scenario, where badly formatted answers were labeled as the low-quality, 0 class. For this reason, it is also important to consider the percentage of correctly formed answers. Figure 4 shows that the percentage of correctly formed answers increases as more examples are added to the prompt. The results for FLAN-T5 large are dramatically low in the 0-shot and 1-shot settings, but they considerably improve for multi-shot. The same trend is visible for the FLAN-T5 XL and XXL models. We can also observe that the larger models tend to better format the answers correctly.

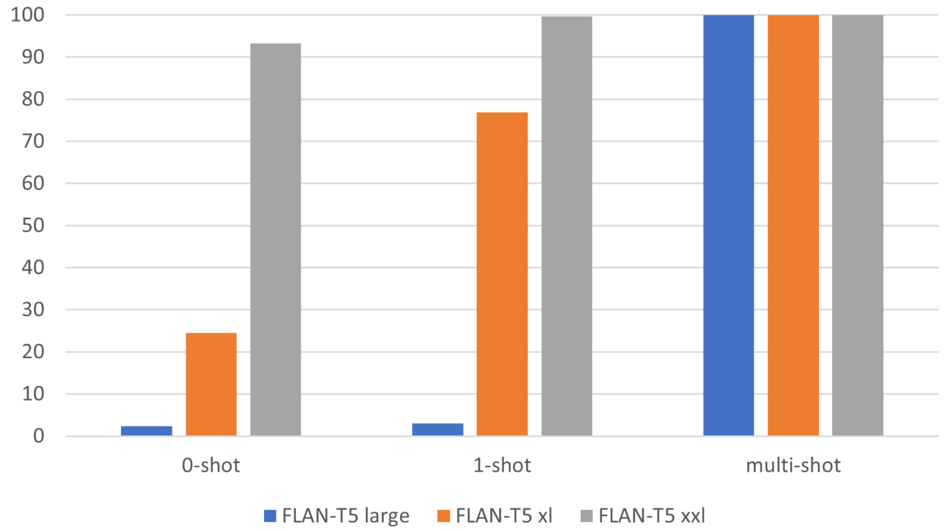


Figure 4. Percentage of correctly formed answers for fine-tuned models on the Overall task

Lastly, experiments were performed to observe whether model performance improves if fine-tuning for more epochs. Preliminary experiments indicated that the test loss would reach a plateau after three epochs of fine-tuning. In order to reduce the number of experiments, we evaluated the FLAN-T5 large, XL, and XXL models only in the multi-shot setting.

Table 5. 3-epoch fine-tuned results for FLAN-T5.

Task	Model	Scenario	Weighted F1-Score
Paraphrase	FLAN-T5 large	multi-shot	86.70%
	FLAN-T5 XL	multi-shot	**86.76%**
	FLAN-T5 XXL	multi-shot	86.21%
Elaboration	FLAN-T5 large	multi-shot	89.33%
	FLAN-T5 XL	multi-shot	**89.88%**
	FLAN-T5 XXL	multi-shot	89.54%
Bridging	FLAN-T5 large	multi-shot	63.72%
	FLAN-T5 XL	multi-shot	**79.02%**
	FLAN-T5 XXL	multi-shot	**79.02%**
Overall	FLAN-T5 large	multi-shot	58.49%
	FLAN-T5 XL	multi-shot	69.85%
	FLAN-T5 XXL	multi-shot	**72.12%**

In this scenario, FLAN-T5 XL performs better in 2 out of 4 cases, while FLAN-T5 XXL considerably outperforms the other two on the Overall task. The FLAN-T5 large model obtains good results on the Paraphrase and Elaboration tasks but has worse results on the remaining two. For the three comprehension strategy tasks, the results are close when comparing the XL and XXL models, with the XL model having a slight advantage. It must be noted that because the FLAN-T5 XXL model was the best-performing model extra hyper-parameter tuning was done to maximize its potential. In the end, this model was trained for all tasks using a smaller learning rate of 1.5e-4, as opposed to the standard 3e-4 used for the other models.

The fine-tuned FLAN-T5 XXL model obtained the best performance on the Overall task, surpassing even the results from previous work (see [7]). The best models in that study were single-task (STL) and multi-task (MTL) neural network architectures based on a pretrained RoBERTa model. The LLM-based methods obtained a better result for the Overall, Paraphrase, and Elaboration presence tasks, while the MTL/STL models still hold a narrow edge over them on the Bridging presence task (see Table 6).

Table 6. Best results across the 2 studies.

Task	Previous results [7]		Current study		
	Best Model	Scenario	Best Model	Scenario	Improvement
Paraphrase	84.3%	STL	86.76%	Fine-tuned XXL multi-shot	2.46%
Elaboration	78.50%	STL	89.88%	Pretrained XL	11.38%
Bridging	89.90%	STL	79.02%	Fine-tuned XXL multi-shot	-10.88%
Overall	69.90%	MTL	72.12%	Fine-tuned XXL multi-shot	2.12%

4. Discussion

This study evaluated the performance of LLMs on scoring self-explanations using multiple employed strategies in either out-of-the-box or fine-tuned setups. In the out-of-the-box scenario, a comparison was made between the performance of FLAN-T5 models and the GPT3.5-turbo API. The FLAN-T5 models obtained better results on three comprehension strategy tasks. The model performance did not scale with the model size and the number of examples listed in the prompts. The GPT3.5-turbo model obtained better results on the Overall quality task and showed a clearer improvement on the other tasks with the addition of more examples to the prompt.

When analyzing the correctness of the responses generated by the LLMs, it was also observed that GPT3.5-turbo and FLAN-T5 large were more likely to generate answers in the correct format. This capability improved for all the models if more examples were provided in the prompt.

Table 7. Confusion matrices for the 'out-of-the-box' models on the Overall task.

FLAN-T5 large 0-shot	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	28	7	14	12
Actual 1	283	254	30	121
Actual 2	428	130	60	308
Actual 3	182	21	54	210
GPT3.5 multi-shot	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	18	10	13	20
Actual 1	274	107	109	198
Actual 2	207	134	222	363
Actual 3	47	27	104	289

When looking at the confusion matrix for the Overall task, the two best-performing out-of-the-box models tend to misclassify multiple examples, not only in adjacent classes but in other classes as well. Numerous instances of class 0 examples are classified as class 3 and vice-versa. This indicates that the models are not able to reliably identify content that had been copied and pasted. In addition, high class imbalance (i.e., class 0 has almost 9 times fewer examples than class 2) influences the predictions.

In the fine-tuning scenario, only FLAN-T5 models were targeted. Initially, the models were fine-tuned 1-epoch using the LoRA method. After this fine-tuning, the performances drastically improved and scaled better with model size and number of examples provided. When the models were trained for 3 epochs, the differences between the FLAN-T5 XL and XXL models decreased.

Table 8. Confusion matrices for the best fine-tuned FLAN-T5 model on the Overall task.

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	18	21	18	4
Actual 1	39	512	132	5
Actual 2	1	110	735	80
Actual 3	0	5	179	283

The confusion matrix generated for the best-performing, fine-tuned model on the Overall task shows improved results when compared with the out-of-the-box models. For almost all classes, except the underrepresented class 0, we observe that most predictions coincided with the ground truth. Furthermore, even when errors occurred, they appeared in the vicinity of the correct options; only four instances of errors occurred at a distance of three classes (i.e. class 0 examples evaluated as being class 3).

Table 9. Training time per model.

Model	**Num epochs**	**Total training time (minutes)**	**GPU type**
MTL	25	20	Tesla P100
FLAN large	1	23	Tesla P100
FLAN XL	1	100	Tesla A100 40GB
FLAN XXL	1	180	Tesla A100 40GB

The FLAN fine-tuned models and the previous MTL approach can also be compared in regards to training time, as reported in Table 9. We can observe that the MTL model required the least training time while using less performant hardware. For the FLAN models, the training time listed is for 1-epoch, so the 3-epoch fine-tuned model would take roughly 3 times more time to train. The previous MTL model performance was surpassed by our best-performing 3-epoch-trained FLAN XXL, but that model required 540 minutes to train (a 27x increase), and also more expensive hardware.

4.1. Error analysis

Table 10 lists 10 randomly selected inputs, at least one per class, on which the following three models were evaluated: the best-performing model (FLAN-T5 XXL multi-shot), the best performing out-of-the-box FLAN-T5 model (FLAN-T5 large 0-shot), and GPT3.5-turbo (prompted in a multi-shot setting). Our evaluation parser considered all listed outputs valid, despite GPT3.5's extra verbosity in listing the class description along with the class name or the lack of parenthesis for FLAN-T5 large on example 1. The models performed better when prompted with alphabetical classes as options instead of numerical ones. For this reason, the classes appear with a different naming convention in this table compared to previous mentions. However, the correspondence is easy to understand as class (A) corresponds to class 0, (B) to class 1, (C) to class 2, and (D) to class 3.

Examples 2 and 3 show all models answering correctly when classifying input belonging to classes 2 and 3. There are also cases (examples 4 and 5) of minor errors where the models classify a good example as having high quality. One possible explanation is that the self-explanations were particularly verbose and the models had trouble keeping track of all information and comparing it with the source text.

Both out-of-the-box models have an example three classes away from the ground truth (see example 6). This is a classic example of copy-pasted content, labeled as low-quality because the reader did not make an effort to self-explain the source text. The fine-tuned FLAN-T5 XXL model manages to detect this and correctly rate the example as Poor quality.

The performance obtained by these models on this 10-example subset is consistent with the previously presented results. The best-performing FLAN-T5 XXL model was correct in 9 out of 10 situations, and it was 1 class away from the correct answer in the erroneous case. The out-of-the-box models managed to correctly answer in 2 or 3 out of 10 cases, exhibited errors 2-class or 3-class away from the ground truth, and they also had minor issues in correctly formatting the output.

4.2. Limitations

The class imbalance was an impediment in some cases, especially for the Elaboration presence task, where class 0 accounted for roughly 93% of samples. This made it tempting for the fine-tuned models to disproportionately label new examples as low quality since it seemed like a sure bet. For that reason, more focus in the analysis was put on the Overall quality task which, in addition to being more complex, was also one of the more balanced tasks.

Especially in regards to the GPT models our experiments were restricted by the costs of using the API. Because we sought to explore multiple scenarios (i.e. 0-shot, 1-shot, multi-shot) for all four tasks, and also explore different variations of prompts, doing so for multiple OpenAI endpoints, would have increased our cost, especially considering the fact that the GPT4 model is roughly 20-30x more expensive than GPT3.5-turbo.

The experiments presented in this study have not been evaluated in an iterative setting, where the request is not modeled as a monolithic prompt but as a dialogue with multiple short requests. The initial requests could have provided the context and the examples, while the last request could have been solely focused on the actual classification task. This would have been more advantageous for the GPT3.5-turbo model which was targeted more towards usage in a conversational setting.

5. Conclusions

This study, corroborated with the previous work of Nicula *et al.* [7], indicates that the task of evaluating reading strategies and assessing overall self-explanation quality can be

Table 10. Sample outputs.

ID	Self-explanation	Source sentence	FLAN-T5 XXL multi-shot	FLAN-T5 large 0-shot	GPT3.5-turbo	Ground Truth
1	This sentence explains that the circular shape of the red blood cells result in a big surface area, which lets them be efficient at gas diffusion.	The disk shape of red blood cells results in a large surface area, which enables them to be efficient at gas diffusion	(B)	C	(C) Good quality	(B)
2	Red blood cells gets its color from the Hemoglobin.	Hemoglobin also contains iron, which gives blood its red color	(B)	(B)	(B) Fair quality	(B)
3	This sentence explains how hemoglobin, a complex protein in red blood cells, binds to the oxygen and carbon dioxide that the red blood cells transport.	Hemoglobin binds to the oxygen and carbon dioxide that the red blood cells transport	(C)	(C)	(C) Good quality	(C)
4	This sentence is saying that red blood cells have essentially two jobs, the second of which being the removal of carbon dioxide that is no longer needed. Oxygen enters the body, and waste carbon dioxide leaves the body with the help of red blood cells.	They also pick up waste carbon dioxide for removal	(C)	(C)	(C) Good quality	(D)
5	Red blood cells carry oxygen to the cells and remove waste. The way they are shaped allows gas diffusion to go well. Once the red blood cells have the oxygen and carbon dioxide waste, hemoglobin binds them.	Hemoglobin binds to the oxygen and carbon dioxide that the red blood cells transport	(D)	(C)	(C) Good quality	(D)
6	the shape of the cells causes them to clog blood vessels, preventing oxygen from reaching muscles and other tissues	The shape of the cells causes them to clog blood vessels, preventing oxygen from reaching muscles and other tissues	(A)	(D)	(D) High quality	(A)
7	When low amounts of oxygen are transported, a person can feel tired or weak due to the body not being replenished completely.The heart, lungs, and muscles rely on oxygen to function, so if there is a deficiency of that a person would become fatigue.	This makes a person feel tired and weak	(D)	(C)	(B) Fair quality	(D)
8	if you have a lot of iron, it will make your blood red	Hemoglobin also contains iron, which gives blood its red color	(B)	(C)	(A) Poor	(B)
9	This means that because of the red blood cells shape being like a disk it helps the body with gas diffusion. Like if the body has a lot of gas build up in it then the red blood cells help get rid of the gas.	The disk shape of red blood cells results in a large surface area, which enables them to be efficient at gas diffusion	(C)	(C)	(B) Fair quality	(C)
10	As a result, the person feels sluggish and has less energy. They are lacking the oxygen which presumably messes up their oxygen:carbon dioxide ratio.	This makes a person feel tired and weak	(C)	(A)	(B) Fair quality	(C)

solved using deep learning models. This work shows that, with fine-tuning, pretrained LLMs surpass the performance of more specialized medium-sized neural network architectures. The LLM models require a more expensive hardware setup for fine-tuning and can have more inference latency than shallower medium-sized models, but they are easier to adapt to a new task than a specialized medium-sized model.

These approaches can be leveraged to develop systems that can either evaluate readers' existing text comprehension abilities or even gradually guide them to improve their performance.

Author Contributions: Conceptualization, M.D. and D.M.; methodology, B.N., M.D., and D.M; software, B.N.; validation, B.N., M.D. and D.M.; formal analysis, M.D.; investigation, D.M.; resources, R.B. and T.A.; data curation, R.B. and T.A.; writing—original draft preparation, B.N.; writing—review and editing, M.D., R.B., T.A and D.M.; visualization, B.N.; supervision, M.D. and D.M.; project administration, M.D. and D.M.; funding acquisition, M.D. and D.M.. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Research, Innovation, and Digitalization, project CloudPrecis, Contract Number 344/390020/06.09.2021, MySMIS code: 124812, within POC, the Ministry of European Investments and Projects, POCU 2014-2020 project, Contract Number 62461/03.06.2022, MySMIS code: 153735, the IES (R305A130124, R305A190063), the U.S. Department of Education, and the NSF (NSF REC0241144; IIS-0735682).

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki and approved by the Institutional Review Board of Arizona State University.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The code, as well as the links to models published on HuggingFace, can be found at https://github.com/readerbench/self-explanations (accessed on 12 September 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BERT	Bidirectional Encoder Representations from Transformers
CoT	Chain-of-thought
LLM	Large Language Model
LoRA	Low-Rank Adaptation
NLP	Natural Language Processing
SE	Self-explanation
SERT	Self-explanation Reading Training

References

1. McNamara, D.S. Self-explanation and reading strategy training (SERT) improves low-knowledge students' science course performance. *Discourse Processes* **2017**, *54*, 479–492.
2. McNamara, D.S. SERT: Self-explanation reading training. *Discourse processes* **2004**, *38*, 1–30.
3. Anders Ericsson, K. Deliberate practice and acquisition of expert performance: a general overview. *Academic emergency medicine* **2008**, *15*, 988–994.
4. McNamara, D.S.; Arner, T.; Butterfuss, R.; Fang, Y.; Watanabe, M.; Newton, N.; McCarthy, K.S.; Allen, L.K.; Roscoe, R.D. iSTART: Adaptive Comprehension Strategy Training and Stealth Literacy Assessment. *International Journal of Human–Computer Interaction* **2023**, *39*, 2239–2252.
5. McNamara, D.S.; O'Reilly, T.; Rowe, M.; Boonthum, C.; Levinstein, I.B. iSTART: A web-based tutor that teaches self-explanation and metacognitive reading strategies. *Reading comprehension strategies: Theories, interventions, and technologies* **2007**, pp. 397–421.
6. Jurafsky, D.; Martin, J.H. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*; Pearson Prentice Hall: Upper Saddle River, N.J., 2009.
7. Nicula, B.; Panaite, M.; Arner, T.; Balyan, R.; Dascalu, M.; McNamara, D., Automated Assessment of Comprehension Strategies from Self-explanations Using Transformers and Multi-task Learning; 2023; pp. 695–700. https://doi.org/10.1007/978-3-031-36336-8_107.
8. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent Abilities of Large Language Models, 2022, [arXiv:cs.CL/2206.07682].
9. Chiesurin, S.; Dimakopoulos, D.; Cabezudo, M.A.S.; Eshghi, A.; Papaioannou, I.; Rieser, V.; Konstas, I. The Dangers of trusting Stochastic Parrots: Faithfulness and Trust in Open-domain Conversational Question Answering, 2023, [arXiv:cs.CL/2305.16519].
10. Perez, F.; Ribeiro, I. Ignore Previous Prompt: Attack Techniques For Language Models, 2022, [arXiv:cs.CL/2211.09527].

11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, ; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in neural information processing systems. Curran Associates Inc., 2017, Vol. 30, p. 5998–6008.
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Proceedings of the 2019 Conference of the NAACL, Volume 1, 2019, pp. 4171–4186.
13. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* **2019**, *abs/1907.11692*, [1907.11692].
14. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners, 2020, [arXiv:cs.CL/2005.14165].
15. OpenAI. GPT-4 Technical Report, 2023, [arXiv:cs.CL/2303.08774].
16. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* **2020**, *21*, 1–67.
17. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling Instruction-Finetuned Language Models, 2022. https://doi.org/10.48550/ARXIV.2210.11416.
18. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, 2019, [arXiv:cs.CL/1804.07461].
19. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems, 2020, [arXiv:cs.CL/1905.00537].
20. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling Language Modeling with Pathways, 2022, [arXiv:cs.CL/2204.02311].
21. Christiano, P.F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; Amodei, D. Deep Reinforcement Learning from Human Preferences. In Proceedings of the Advances in Neural Information Processing Systems; Guyon, I.; Luxburg, U.V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R., Eds. Curran Associates, Inc., 2017, Vol. 30.
22. McNamara, D.S.; Newton, N.; Christhilf, K.; McCarthy, K.S.; Magliano, J.P.; Allen, L.K. Anchoring your bridge: the importance of paraphrasing to inference making in self-explanations. *Discourse Processes* **2023**, *60*, 337–362, [https://doi.org/10.1080/0163853X.2023.222575 https://doi.org/10.1080/0163853X.2023.2225757.
23. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, 2021, [arXiv:cs.CL/2107.13586].
24. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models, 2021, [arXiv:cs.CL/2106.09685].

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.