A Case Study on ChatGPT Question Generation

Winston Chan, Aijun An
Dept. of Electrical Engineering and Computer Science
York University
Toronto, Canada
winston.sh.chan@gmail.com, aan@yorku.ca

Heidar Davoudi
Faculty of Science
Ontario Tech University
Oshawa, Ontario
heidar.davoudi@ontariotechu.ca

***Abstract*—The advent of transformers and the subsequent development of Large Language Models (LLMs) based on these technologies has revolutionized the field of Natural Language Processing (NLP). These models are able to understand and generate coherent natural language and hold conversations with humans continuously. Meanwhile, ChatGPT has become famous among many LLMs for its general-purpose characteristics and versatility. With that in mind, we investigate the capabilities of ChatGPT, which is very successful in many downstream NLP tasks on the task of Question Generation (QG). In particular, our experiments show that appropriate context through our designed prompts makes ChatGPT an appropriate tool for accurately performing the QG task. We compare ChatGPT's question generation results with the state-of-the-art models, particularly on the SQuAD and car manual datasets. The results show that ChatGPT is able to compete with or even outperform some of the baseline models. Furthermore, we illustrate that we may improve ChatGPT through additional fine-tuning of the prompts. Finally, we also investigate the use of ChatGPT to evaluate QG models. While the use of ChatGPT for such purposes is still in its early stages, our results demonstrate that ChatGPT can potentially be a strong QG accuracy evaluator comparable to human evaluators.**

***Index Terms*—natural language processing, large language model, question generation**

I. Introduction

Natural Language Processing (NLP) focuses on analyzing and understanding natural language and its use in various applications such as conversational systems [1]. From language translations that are done on a daily basis [2] to chatbots that are able to communicate with humans in an effective manner [3] [4], the ability to understand and generate human-like language holds profound implications for various NLP tasks.

Question generation (QG) is one of the NLP tasks that involves creating coherent and contextually relevant questions based on natural language texts providing *context* and *answers* [5]. This task requires understanding of the *context* and *answers* and generating natural language responses according to the given context and answers. The QG task holds a pivotal role in numerous real-life applications, like its uses in the education setting [1], FAQ generations, and automatic customer support systems [6]. Modern transformer models, such as T5 (Text-To-Text Transfer Transformer) [7] and BART (Bidirectional and Auto-Regressive Transformer) [8], showcase the remarkable capabilities of contemporary NLP techniques in QG tasks.

With the emergence of transformers, LLMs have also gained significant traction, with the most notable example being ChatGPT, an LLM powered by the Generative Pre-trained Transformer 3 (GPT-3) [9]. The ChatGPT underlying transformer architecture allows it to understand the context, produce coherent text, and, most impressively, engage in a continuous interactive conversation [10]. With its ability to perform a wide variety of natural language tasks, ChatGPT is considered highly valuable for its general-purpose usage.

For the purpose of automatically generating the *knowledge base* for interactive question-answering (QA) systems, we have been working with iNAGO Corp.[1] on the automatic QG task. The knowledge base of iNAGO's interactive QA system (i.e., netpeople platform[2]) consists of question-and-answer (QA) pairs, which were traditionally curated by human based on domain documents. To alleviate human efforts, we have been working with iNAGO on developing sequence-to-sequence text generation models, such as T5[7], BART[8], and SRL-Seq2Seq [5] for the QG task.

The objective of this paper is to comprehensively assess the viability of utilizing ChatGPT for the task of question generation. To do so, we conducted experiments using ChatGPT to generate questions on some sample texts and evaluated their performance against state-of-the-art baseline models. Specifically, ChatGPT is utilized to generate highly relevant questions to the given context. We leverage the unique conversational abilities of ChatGPT through a prompt engineering mechanism. The investigation aims to compare ChatGPT's capabilities against baseline models such as T5 and BART. We also aim to learn the characteristics of ChatGPT, such as *fine-tuning* that can help users attain accurate responses from it in the QG task. Finally, we investigate the capability of ChatGPT in *assessing* generated questions in comparison to human evaluators.

To summarize, our work offers several noteworthy contributions:

- We present a comprehensive analysis of question generation using ChatGPT, a readily available Large Language Model (LLM) model, by employing our specifically de-

[1]https://www.inago.com/

[2]https://www.inago.com/products/#netpeople-assistant-platform

979-8-3503-2445-7/23/$31.00 ©2023 IEEE

signed prompts and comparing its performance with state-of-the-art transformer-based models.
- We introduce a set of meticulously crafted prompts tailored to assess various facets of the generated questions. These prompts have undergone multiple iterations of improvements to facilitate the evaluation task effectively. Our preliminary findings reveal that these evaluators closely align with human evaluators, enhancing the reliability and comprehensiveness of our assessments.
- We design prompting techniques designed to mitigate ChatGPT's limitations in question generation. These techniques aim to harness the model's strengths while addressing identified shortcomings, providing practical insights for improving performance in this specific task.
- We investigate the impact of fine-tuning when using ChatGPT for question generation, showcasing the effectiveness and implications of this technique in the context of question-generation tasks with limited training data.

The remainder of this paper is organized as follows. In Section II, we describe related work. Section III discusses our methodology, and Section IV presents the results of our experiments.

II. Related Work

Question Generation (QG) has evolved from the use of rule-based techniques to data-driven machine learning approaches (e.g., sequence-to-sequence models [5]).

Rule-based question generation is an approach of using predefined rules and patterns to generate questions from a given text automatically [11]. These rules are based on capturing the given text's linguistic patterns and syntactic structures and converting the identified information into questions [12]. However, considering the nuances of the human language, it is impossible to create a complete set of rules. That is, creating a set of rules to mimic how humans curate questions from a given text is an almost impossible task.

The Sequence-to-sequence (Seq2Seq) approach offers an improvement over the rule-based approach, addressing the limitations associated with the resource-intensive rule creation process. Early Seq2Seq work involves the use of vanilla models or ones based on Recurrent Neural Networks[13], [14].

Most recently, the transformer-based models are successfully applied to question generation. Many of these models are able to achieve state-of-the-art performances, including ERNIE-GEN [15], BART TextBox 2.0 [16], ProphetNet [17], and UniLMv2 [18], to name a few. The following paragraphs describe prominent models employed in the QG task:

1) Text-To-Text Transfer Transformer (T5): The model is introduced by Google Research as a framework built upon the transformer architecture [7]. It sets itself apart with its "text-to-text" nature, where it can accept input data in the format of a text prompt that uses natural language to describe a task in a human-like manner. The model was then trained to produce a natural language response.

2) Bidirectional and Auto-Regressive Transformers (BART): The BART, by Meta Research, is also a model based on the transformer architecture [8]. Its unique approach of combining the bidirectional and auto-regressive capabilities allowed for the completion of various NLP tasks. This dual approach utilizes the bidirectional encoder to understand context alongside the auto-regressive decoder, which is responsible for the generation of a human-readable response.

3) Large Language Models (LLMs): These models have transformed how to approach NLP tasks, including question generation. The recent emergence of ChatGPT by OpenAI has further expanded the capabilities of Large Language Model (LLM) in regard to human-like text generation and conversation [9]. One of ChatGPT's standout capabilities are its ability to maintain conversational context, make continuous interactions between machines and humans. It is also known for its ability to perform a wide variety of tasks with limited to no additional training.

4) ProphetNet: ProphetNet is a transformer-based model developed by Microsoft Research that focuses on tackling sequence-to-sequence text generation and language understanding tasks [17]. The future n-gram prediction, which is a unique self-supervised objective, is incorporated to reward the planning of upcoming n tokens and reduces the risk of overfitting to strong local correlations. This is unlike conventional Seq2Seq models, where only one step in the future is predicted. ProphetNet achieved state-of-the-art QG results on benchmark datasets like MS MARCO and NewsQA [19], [20].

5) Unified Pre-trained Language Model (UniLM): The UniLM can be fine-tuned for both language understanding and generation tasks [21]. The unified capability of bidirectional, unidirectional, and sequence-to-sequence language models is achieved through a shared transformer network and self-attention masks, which enables precise control over contextual information. In addition to using UniLM alone, the ASGen pre-training approach is successfully applied to UniLM, and the experimental result shows that it is able to compete with state-of-the-art models on several natural language generation datasets [19], [21].

The primary objective of this paper is to explore the feasibility and effectiveness of using ChatGPT for the task of question generation.

III. Methdology

In this section, We are going to detail the various tasks for which we use ChatGPT. (Note that by ChatGPT, we are referring to the OpenAI model GPT-3.5-turbo).

A. Question Generation

The idea of QG with ChatGPT is to give it a context passage and instruct it to generate questions. As simple as it may sound, such a task involves several stages of fine-tuning. The main method of fine-tuning is prompt engineering - a method of carefully constructing and improving the prompt to guide the model's behavior [22]. The prompt is often constructed in a way that is easy to understand for ChatGPT but not for

humans, which can be a very counter-intuitive process of fine-tuning. We will outline the high-level idea of our approach to tune the QG prompt.

At first, an initial prompt that states the general purpose of the task was created. It can be as simple as two to three sentences. The purpose is for us to understand whether ChatGPT is able to perform the task of QG in the first place.

Then, we perform iterative refinement runs by manually reviewing the initial prompt results and making appropriate adjustments to the prompt. In earlier runs, it would typically involve making incremental changes to the prompt by providing more and more specific instructions and seeing if ChatGPT is able to follow the instructions.

In this whole process of testing whether ChatGPT is able to give us our expected results, the model *temperature* needs to be configured accordingly. Note that temperature is the main configurable parameter of ChatGPT, which affects the response it gives. It is a parameter that controls the randomness and diversity of the generated text [23]. The initial runs set the temperature to zero or a very low value (lower than 0.3). Once we have improved the prompt to a point where we get desirable results, the temperature can be momentarily increased to increase ChatGPT's creativity, allowing it to come up with a variety of questions in different runs.

Regarding the quality of the questions, we have defined three criteria of what makes a good question:

- *Relatedness*: The generated questions must be related and relevant to the corresponding context passage. This is the most important criterion to rate the questions on. Note that relatedness and relevance here are distinct and equally important. For instance, a question about a Honda Civic is generated from a Tesla car manual, which can be considered related because both are about automobiles; it is irrelevant since the context and the question are about different brands of cars respectively.
- *Conciseness*: The generated questions must be clear and brief. While not as important as relatedness, a question that is readable is preferred. Although ChatGPT can generate human-like responses, our preliminary findings showed without tuning the QG prompt, ChatGPT tends to generate convoluted questions, most often by combining several questions as one single question. This is one of the examples of unconcise questions which we want the generator to be able to avoid.
- *Completeness*: The set of questions generated from a given context must cover all information in the context. Unlike relatedness and conciseness, which rate each question individually, we would like to rate the set of questions as a whole with completeness. That is because each question can only cover some information in the context, meaning rating individual questions' completeness would be meaningless. Instead, our goal is to observe whether all the questions can cover all the information in the context.

The instructions in the prompt are designed around these three criteria, where the individual questions are expected to be related to the context and concise, and the entire set of questions is expected to be complete.

B. Question Evaluator

Having the QG prompt in place, we require a way to evaluate these questions. Many question-generation systems are trained on datasets that include ground truth questions, meaning during the evaluation phase, the systems' predictions would typically be compared against the references, which would be able to show the quality of the generated questions. Existing automatic evaluation metrics like BLEU, ROUGE-L, and METEOR are widely used in a realm of NLP, where the predictions, which are the questions generated, are evaluated on how well they align with the ground truth references [5].

However, the composition of QG datasets can be very costly. For the ground truth reference questions to be reliable, it would require careful curation by a qualified person (usually a linguist). They would read through the textual materials that the system is based on and manually write questions that are related to the materials, which would act as the ground truth reference. This task is often seen to be extremely time-consuming because it requires manual crafting of questions based on the context and understanding of the given text.

An alternative to using automatic evaluation metrics is to perform human evaluation, requiring a qualified human evaluator to inspect every question and rate its quality. This method does not require ground truth questions, meaning that the time spent constructing a dataset would otherwise be allocated to manually evaluating the questions. Again, this task requires manual effort, which can be time-consuming.

To solve the problem of inefficient curation, Amazon released the service Amazon Mechanical Turk in 2005, an online crowd-sourcing marketplace allowing tasks to be outsourced to a global network of workers. These tasks often require human intelligence and crowd workers to hold certain qualifications to be considered fit for a specific task. This service shortened the time needed for tasks like question curation because tasks can be divided into smaller sub-tasks, which can be done virtually and parallelly. Although this cuts the amount of time needed for question curation, considering the substantial task volume, the cost of such a task is expected to be very high.

With that in mind, we set out to design a reference-less evaluation metric. Specifically, we developed a ChatGPT-based Question Evaluator, where the questions generated and their corresponding context are provided to ChatGPT, which would then rate the questions based on the criteria - relatedness, conciseness, and completeness. Theoretically, by leveraging this method, the cost of evaluation can then be reduced, and we would have a certain standard for QG. This evaluator would be suitable for any reference-less QG and require significantly less manual effort.

Jiao et al. previously explored this method of using ChatGPT for NLP task evaluation [24]. In their work, the task of machine translation was evaluated, and the authors showcased the potential of using ChatGPT as an evaluation method for

text generation. The goal here is to investigate whether a ChatGPT evaluator is usable for QG.

To design the prompt for the question evaluator, the process is identical to that of QG. However, the tuning of the question evaluation prompts is more complicated. For one, there needs to be a prompt for each criterion, and each prompt also requires independent tuning. Also, unlike QG, which is an open-ended task that can generate more than one acceptable response, the question evaluator requires a definitive rating for a question, meaning that there can only be one answer. Hence, the prompt tuning process needs to be performed with more effort. As for the temperature, it is set to zero throughout the fine-tuning process because we expect the ratings to be predictable.

In Section IV-C we will evaluate the performance of ChatGPT as a question evaluator by comparing its evaluation with human evaluation.

IV. Experimental Results

A. Datasets

In order to evaluate the questions generated by ChatGPT, two datasets are used. The first dataset is the Stanford Question Answering Dataset 1.1 (SQuAD 1.1), which contains QAs created by Amazon Mechanical Turk Crowd-workers from Wikipedia articles. Each example in the dataset consists of a context, an answer, and the corresponding ground-truth question [25]. The dataset has both training and test datasets. Table I shows the sizes of the datasets.

The second dataset is the Car Manuals dataset, a dataset created by iNAGO Corp., which consists of answer-and-question pairs without a context paragraph in each example. The dataset also contains a training and a test dataset, whose sizes are shown in Table I. By comparing the results on both datasets, we can investigate the effect the context paragraph has on an LLM.

TABLE I
Sizes of the datasets.

Dataset	training set	testing set
SQuAD 1.1	70,484	11,877
Car Manual	9,184	1,869

B. Question Generation without Fine-tuning ChatGPT

We report the results of question generation without fine-tuning ChatGPT. Only the test data of each dataset is used when evaluating ChatGPT in this experiment. On SQuAD, the context and the answer to a question are provided to ChatGPT in the form of a prompt, and the questions generated by ChatGPT are compared with the ground-truth questions in the test dataset. Only the answer is provided to ChatGPT on Car Manuals without a context paragraph.

1) Results on SQuAD Dataset: Table II shows the automatic evaluation results of ChatGPT on the SQuAD 1.1 dataset against various state-of-the-art paragraph-based question generation models[3]. The BLEU-4 and ROUGE-L results show that while ChatGPT's precision scores are on the lower side, the METEOR scores are significantly better than all models. Considering the different natures of the evaluation metrics, we believe GPT-3.5 performed worse in BLEU-4 and ROUGE-L but better in METEOR because the generated questions were phrased differently from the ground-truth questions, but they essentially convey the same meaning, i.e., they both ask about the same thing. The generated questions being rephrased or restructured variance of the references would explain the relatively low BLEU-4 and ROUGE-L scores. The use of synonyms and rephrasing would explain the high METEOR scores.

Fig. 1 shows the prompt used for SQuAD question generation. Fig. 2 shows the prompt used by iNAGO for question generation.

TABLE II
Comparison of automatic evaluation results of ChatGPT on the SQuAD dataset against various state-of-the-art paragraph-based question generation models [16], [17], [19]–[21]. Numbers in bold indicate the best result in each column.

QG Method	BLEU-4	ROUGE-L	METEOR
BART (TextBox 2.0)	**25.1**	52.6	26.7
ProphetNet+ASGen	24.4	**52.8**	26.7
ProphetNet+syn.mask+localness	24.4	**52.8**	26.3
ProphetNet	23.9	52.3	26.6
UniLM+ASGen	23.7	52.3	25.9
UniLM	22.8	51.1	25.1
GPT-3.5-turbo	17.4	40.3	**28.0**

It is interesting to see that while ChatGPT is not fine-tuned on the SQuAD dataset, its ability to generate questions is able to compete with state-of-the-art models that were fine-tuned on the SQuAD training data except for BART (TextBox 2.0), which may only have been the pre-trained model. This shows promising potential for ChatGPT to be used for language generation tasks and tasks that require an understanding of text. In our case, ChatGPT is able to understand the context paragraph alongside the answer and form a set of suitable questions.

2) Results on Car Manuals Dataset: For the experiment on the car manual dataset, in addition to measuring how well the generated questions match the ground-truth questions, we also measure the coverage of the generated questions over the ground-truth questions since, for each input sentence, there may be multiple ground-truth questions. For this purpose, for each performance measure (i.e., BLEU-4, ROUGE-L, and

[3]Data gathered from https://paperswithcode.com/sota/question-generation-on-squad11

Given the following context paragraph and answer, generate a question that can be answered by the provided answer:

Context: {put context here}
Answer: {put answer here}

Fig. 1. This is the prompt used for SQuAD QG. Note that for car manual QG, simply remove the lines including "context" in the prompt.

Given a passage from the user manual of a specific machinery, imagine you are operating said machinery, and generate a set of questions that a user operating it might ask.

Passage: {put passage here}

Each question should be closely related to the passage.
The set of questions should be sufficient to completely cover the information in the passage, no more or less.
Questions must not be about the passage location.
Each question should not be a combination of two or more questions.
Each question should focus on practical application, problem-solving, or inquiry for clarification/definition, rather than being a traditional quiz question.
Each question must be answerable by the information in the passage.
Each question should be clear and brief.
Each question should not be overly-specific.

Fig. 2. This is the main prompt used by iNAGO. Note that this prompt is tailor-made for iNAGO's preference for QG. For your own use, some degree of tuning is required.

METEOR), $precision$, $recall$, and $F\text{-}score$ are used to assess the quality of the generated questions [26]. In the context of question generation, $precision$ measures how accurate and relevant the questions are; $recall$ measures the degree of relevant information coverage of the generated set of questions. If $precision$ is prioritized, generated questions can be accurate, but the model misses out on generating questions for other relevant information. Conversely, if $recall$ is prioritized, many questions are generated, but some might be unrelated to the given text.

Specifically, the $precision$ and $recall$ are calculated below [5]:

$$precision(G, T, s) = \frac{1}{|G|} \sum_{g \in G} \max_{t \in T} s(g, t)$$

$$recall(G, T, s) = \frac{1}{|T|} \sum_{t \in T} \max_{g \in G} s(g, t)$$

where G is the set of generated questions, T is the set of ground-truth questions, and s is the scoring functions that are BLEU-4, ROUGE-L, and METEOR. The $F\text{-}score$ is calculated as the harmonic mean of the $precision$ and $recall$ [5].

Table III shows the automatic evaluation results of ChatGPT on the car manual dataset against BART[4], T5[5], and two versions of SRL-Seq2Seq [5]. Based on the results, ChatGPT is significantly inferior to most other models, with exceptions to the METEOR scores. Note that the difference between the SQuAD dataset and car manual dataset set experiments is that while both the context and the answer are provided to ChatGPT in the SQuAD experiment, only the answer is given to ChatGPT for QG in the car manual experiment. This shows that the context is a piece of important information to provide to ChatGPT, and the lack of said context can heavily and negatively limit its capability. This would also align with our expectations. The non-GPT QG models are trained with a domain-specific purpose, and these models will perform better on the task they are designed for. Table IV shows an example where the model understands the context, while ChatGPT cannot do so. We speculate that the context of the manual is trained into the model, allowing it to generate questions with only the answer. Again, this showcases the importance of context for ChatGPT, especially when we are prompting it to perform an untrained task.

[4]The BART-base model is used.

[5]T5-small is used.

C. Question Evaluator

To see how ChatGPT performs as a question evaluator, a subset of 150 questions is randomly chosen for a linguist and ChatGPT to rate. After that, the results are compared. The linguist and ChatGPT are asked to rate the questions based on the criteria on a scale from 1 to 3, with 1 being bad and 3 being good. The evaluation performed for the question evaluator is only meant to show whether using ChatGPT to evaluate question quality is viable.

TABLE III

COMPARISON OF AUTOMATIC EVALUATION RESULTS OF CHATGPT ON THE CAR MANUAL DATASET AGAINST BART, T5, AND SRL-SEQ2SEQ [5]. P, R, AND F REPRESENT PRECISION, RECALL, AND F-SCORE, RESPECTIVELY. NUMBERS IN BOLD INDICATE THE BEST RESULT IN EACH COLUMN.

QG Method	BLEU-4			ROUGE-L			METEOR		
	F	P	R	F	P	R	F	P	R
BART	57.2	63.7	51.9	71.7	75.9	68.0	57.6	63.7	52.6
SRL-Seq2Seq with Soft+C	88.3	**85.4**	91.4	94.3	**94.0**	94.6	63.0	**62.1**	63.9
SRL-Seq2Seq with SRLSoft+L	**89.0**	85.1	**93.2**	**94.8**	93.7	**95.9**	**63.6**	61.8	**65.5**
T5	45.0	50.3	40.7	62.4	66.0	59.2	46.3	50.0	43.1
SRL-Seq2Seq with Soft+C	85.9	84.1	87.8	91.9	91.5	92.4	59.9	59.3	60.5
SRL-Seq2Seq with Soft+L	84.7	82.8	86.6	91.0	90.1	92.0	58.8	57.6	60.0
GPT-3.5-turbo	21.5	18.7	25.2	28.3	30.6	26.4	42.9	42.5	43.3

TABLE IV

AN EXAMPLE SHOWING HOW LACKING CONTEXT CAN AFFECT CHATGPT'S PERFORMANCE. FROM THE PROVIDED ANSWER, THE T5 MODEL, WHICH IS TRAINED ON THE CAR MANUAL DATASET, UNDERSTANDS THAT THE "CAMERA LENS" REFERS TO THE REARVIEW CAMERA, WHILE CHATGPT DOES NOT UNDERSTAND IT BECAUSE OF THE LACK OF CONTEXT.

Provided answer: "The camera lens is located in the handle of the trunk lid."
Ground truth question: "Where is the **rearview** camera located?"
T5 generated question: "Where is the **rearview** camera?"
ChatGPT generated question: "Where is the camera located?"

Out of the 150 questions, there are:

- 81 entries where ChatGPT's rating matches the linguist's rating
- 14 entries where GPT's rating is higher than that of the linguist
- 55 entries where GPT's rating is lower than that of the linguist

While the majority of ChatGPT's ratings align with that of the linguist, this is far from perfect. As a follow-up investigation, a few of the questions that are poorly rated are passed to GPT-4 for further testing. Table V shows the result comparison on rating the questions' relatedness to the passage, which is a passage extracted from a Tesla Model 3 owner's manual. Judging from this small set, GPT-4 is able to rate questions more accurately. Although the results of GPT-3.5-turbo are far from perfect, GPT-4 shows that ChatGPT has the potential to be used as an evaluator. We expect later versions of ChatGPT to be able to handle such a task more accurately.

Fig. 3 shows the prompts for individual question evaluation; Fig 4 shows the prompts for set question evaluation.

D. Question Generation with Fine-tuned GPT-3.5

The fine-tuning feature of GPT-3.5-turbo has recently been released by OpenAI, which allows for fine-tuning of the GPT model with training samples in the conventional GPT chat format. This feature is mainly introduced to address the limited amount of training examples that can fit into the prompt. As such, the idea of this experiment is to evaluate the effectiveness of GPT fine-tuning. We first gather 2,000 training samples from the car manual dataset and use them to fine-tune GPT-3.5 for question generation. Then, we pick 200 testing samples from the car manual dataset, which we use to compare the questions generated by GPT-3.5, GPT-4, and fine-tuned GPT3.5.

Table VI shows the fine-tuned results against that of the un-tuned GPT models. Our experimental results provide evidence of the efficacy of fine-tuning GPT-3.5 on a limited training set. When evaluated against both the un-tuned GPT-3.5 and GPT-4 models, the fine-tuned GPT-3.5 model consistently outperforms them across multiple key metrics. This outcome highlights the potential of fine-tuning as a valuable technique for optimizing the performance of on-the-shelf models for more specific usages.

V. CHARACTERISTICS OF CHATGPT

In this section, we will go over the characteristics of ChatGPT. Note that different prompts are required for utilizing ChatGPT to perform certain tasks. Also, considering the fact that a prompt that makes the most sense to ChatGPT does not necessarily mean it is the same case for humans, designing a good prompt can be very counter-intuitive. With that said, some subsections also showcase ways to get stable results based on our experience with prompt tuning.

A. Prompt-sensitive Nature

ChatGPT is extremely prompt-sensitive. During our experiments, we submitted two seemingly identical question evaluation prompts to ChatGPT with the temperature set to

TABLE V
AN EXAMPLE SHOWING THE DIFFERENCE BETWEEN GPT-3.5-TURBO AND GPT-4 ON RATING THE QUESTIONS' RELATEDNESS. JUDGING FROM THE RESULTS HERE, IT SEEMS GPT-4 IS ABLE TO SPOT THE BAD QUESTIONS BETTER THAN GPT-3.5-TURBO, AS BOTH QUESTIONS RATED AS "1" BY GPT-4 ARE NOT RELATED TO TESLA MODEL 3.

Question	GPT-3.5's ratings	GPT-4's ratings
How do I use my phone to access my Model 3?	3	3
What do I need to do to authenticate my phone as a key for my Model 3?	3	3
Can I use any phone to access my Model 3?	3	3
What is today's date?	3	1
Is it necessary to keep my phone's Bluetooth on at all time to access my Model 3?	3	3
How do I open doors?	3	1

TABLE VI
COMPARISON OF AUTOMATIC EVALUATION RESULTS OF FINE-TUNED GPT-3.5 ON A SUBSET OF CAR MANUAL DATASET AGAINST GPT-3.5 AND GPT-4. NUMBERS IN BOLD INDICATE THE BEST RESULT IN EACH COLUMN.

QG Method	BLEU-4			ROUGE-L			METEOR		
	F	P	R	F	P	R	F	P	R
GPT-3.5	33.7	26.8	45.3	47.6	47.2	47.9	28.6	30.8	26.8
GPT-4	34.0	27.1	45.2	50.6	49.3	51.8	30.1	31.6	28.7
Fine-tuned GPT-3.5	**46.0**	**42.1**	**50.7**	**60.1**	**57.0**	**64.3**	**35.2**	**33.2**	**37.6**

0, but the results that came back were completely different. Specifically, the formatting of the output and the ratings are different. We later discovered that the prompts were not actually identical - one had an addition period at the end of one of the sentences. This shows that even the distinction of a single character can lead to ChatGPT returning completely different results.

Also, regarding the ChatGPT question evaluator, we discovered that giving GPT multiple questions to rate tends to result in more reasonable ratings than giving ChatGPT one question at a time. We speculate that this is because, with more questions visible to ChatGPT in the same context window, it is able to compare the ratings of the questions and revise some of the ratings it thought to be unreasonably rated. On the other hand, if we pass one question at a time to ChatGPT, it can only analyze one question in each context window, which means it will not be able to make any comparison to other questions.

B. Vague vs. Specific Prompt

In most cases, if more information about the task is provided, ChatGPT gives better responses. From our methodology, we started out with a vague prompt and then made incremental changes to the prompt to include more specific instructions. In most of our experiments, ChatGPT will only get better by making the prompt more detailed. However, there did come a point when we realized the results generated by ChatGPT were starting to be worse because of the prompt being overly specific.

C. Few-shot Learning's Effectiveness

Few-shot learning is a method of giving ChatGPT examples of the response you desire [10]. For instance, if you would like ChatGPT to write a poem in a certain style, it would help immensely for you to give it an example of the poem in said style. For QG, few-shot learning can be performed by including example context and corresponding example questions in the prompt. For the question evaluator, it can be done by providing a set of questions and their correct ratings.

However, it is impossible to comment on few-shot learning's effectiveness generally. While it worked to some extent for QG, it was proven to make the question evaluator worse because it became too strict. In short, few-shot learning's effectiveness varies from case to case.

D. Effect of Ordered Instructions

There are different ways to give ChatGPT instructions. Normally, if your task is simple enough, the instructions can be as short as a few sentences. The prompt could include up to a few paragraphs of instructions for more complicated tasks. In some cases, ChatGPT would only follow some instructions but ignore others. This problem needs to be addressed because, for some deterministic tasks, like the question evaluator, where a question can only have one correct rating, we want ChatGPT to follow the instructions strictly.

To tackle this issue, we tested different instruction formats in the prompt and discovered that putting instructions in an ordered list can help with this issue. Still, this does not guarantee that all important instructions are strictly followed,

You will be provided with a passage extracted from a PDF manual and its section title.

Instructions:
1. Read the passage carefully.
2. Read through the questions.
3. Assess the relatedness of each question to the passage content without assuming any information not explicitly stated.
4. Rate the questions on a scale of 1 to 3, with 1 meaning "unrelated," 2 meaning "somewhat related," and 3 meaning "closely related."
5. If the passage does not include the answer to the question, the question should be deemed "unrelated".
6. If the passage states the indirect or half answer to the question, and the focuses of the question and the passage are slightly mismatched, the question should be deemed "somewhat related".
7. If the passage explicitly states the answer to the question, and the focuses of the question and the passage are identical, the question should be deemed "closely related".

Questions: {List of questions}
Title: {Title of the passage}
Passage: {Passage}
Return the ratings in json format.
Output the question as the key, and the list of ratings as the value

---

You will be provided with a passage extracted from a PDF manual and its section title.
Instructions:
1. Read the passage carefully.
2. Read through the questions.
3. Assess the conciseness of each question to the passage content without assuming any information not explicitly stated.
4. Rate the questions on a scale of 1 to 3, with 1 meaning "unconcise," 2 meaning "somwhat concise," and 3 meaning "concise."
5. If the question is unclear or contain unnecessary information, the question should be a 1.
6. If the question includes information that is not necessary, but it somewhat relates to the focus of the question, the question should be a 2.
7. If the question is clear and effective, the question should be a 3.
8. Give a strict 1 if the question is a combination of at least two question.

Questions: {List of questions}
Title: {Title of the passage}
Passage: {Passage}
Return the ratings in json format.
Output the question as the key, and the list of ratings as the value

Fig. 3. These are the main prompts used by iNAGO for individual question evaluation. Again, note that these prompts are designed specifically for the usage that fits iNAGO. These prompts underwent multiple iterations of improvement for them to give reasonable results. To use the prompts for your own purpose, you are advised to use these prompts as a reference, and design your own prompt based on your needs.

only that it will help ease the issue. We suspect this is an inherent flaw of GPT.

E. Asking ChatGPT for Explanation

Asking ChatGPT for an explanation of why it gave you a certain response is the best way to get more accurate results. This was discovered when the prompt was being tuned, where we were not able to get the desired results after multiple attempts of tuning the instructions. For QG, where the output is only supposed to be questions, asking ChatGPT for explanations is only meant to understand why ChatGPT is giving you a certain response. It would help users understand why ChatGPT is generating whatever the response is.

F. Effect of the Multi-prompt Approach

In many situations, giving ChatGPT a long list of instructions is preferred. Considering the fact that it is highly prompt-sensitive, it is common for ChatGPT not to follow all instructions. In this case, the multi-prompt approach can be used. It refers to breaking down instructions into more

You will be provided with a passage extracted from a PDF manual and its section title.

Instructions:
1. Read the passage carefully. Identify the number of major aspects in the passage as ”x”, do not change ”x”.
2. Read through the questions.
3. Assess the completeness of the question set as a whole, meaning you are rating the question set as a collective, but not the questions individually.
4. Note that each question in the set can cover only some aspect of the passage, the goal is to evaluate if all major aspects of the passage are covered.
5. Note that the rating correlates with the number of questions. In most case, more questions = higher rating, fewer questions = lower rating.
6. Questions do not need to cover unnecessary or extraneous detail, just the major aspects.
7. Identify the number of major aspects covered by the questions as ”y”.
8. Show me ”x” and ”y” in json format.

Questions: {List of questions}
Title: {Title of the passage}
Passage: {Passage}

Given value of x and y,
Compare x and y:
If y is larger than or equal to x, set ”z” as 3
Else if y is larger than or equal to half of x, set ”z” as 2
Else, set ”z” as 1

Return z only without telling me your explanation.

Fig. 4. This is the main prompt used by iNAGO for completeness question evaluation. Note that this showcases the usage of the multi-prompt approach. This would first prompt ChatGPT to identify the number of major aspects in the passage and in the questions as ”x” and ”y” respectively, and then these two values will be passed as another prompt to get the final rating.

than one part and then prompting ChatGPT for responses sequentially. Under this condition, the multi-prompt approach can help immensely to get more accurate and desirable results. However, assuming the OpenAI API is used for prompting instead of the web version, it would mean a higher cost of usage. Fig. 4 showcases an example of the multi-prompt approach.

VI. Conclusion

We consider ChatGPT to be fit for the task of QG. The experiment on the SQuAD dataset shows that ChatGPT is able to compete against current state-of-the-art QG models, given that the context is provided. The experiment on the car manual dataset shows the importance of providing context when asking ChatGPT to perform an untrained task. Also, for the ChatGPT question evaluator, we are confident that later versions of GPT will be able to bring its evaluation potential to the fullest, as displayed by the difference between GPT-3.5 and GPT-4. Also, our experimental results demonstrate the potential benefits of fine-tuned models for specific use cases.

For future work, it is important to explore the difference between GPT-3.5 and GPT-4; that way, we will know what other natural language generation tasks can be reliably performed with ChatGPT. For QG, admittedly, ChatGPT's questions tend to be quiz-like, i.e., something that would be asked on a quiz show. Questions are sometimes phrased in a way that appears to test the understanding of a person instead of their purpose being for inquiry. In some applications where questions are expected to be user-like, more effort will need to be put into making questions more natural. The investigation is ongoing with iNAGO to discover ways to generate more user-like questions.

Acknowledgment

We would like to thank iNAGO for providing the various resources and their collaboration in executing this case study. We would like to thank Mr. Ron Di Carlantonio and Mr. Gary Farmaner for their guidance on the project. We would also like to thank Dr. Gavin Bembridge and Ms. Machiko Okano for their help with human evaluations.

References

[1] G. Chen, J. Yang, C. Hauff, and G.-J. Houben, “Learningq: A large-scale dataset for educational question generation,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, no. 1, Jun. 2018. DOI: 10.1609/icwsm.v12i1.14987. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14987.

[2] P. Gao, S. Geng, Y. Qiao, X. Wang, J. Dai, and H. Li, *Scalable transformers for neural machine translation*, 2021. arXiv: 2106.02242 [cs.CL].

[3] C. Chang, W.-C. Peng, and T.-F. Chen, *Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms*, 2023. arXiv: 2308.08469 [cs.LG].

[4] C. Li, Z. Leng, C. Yan, *et al.*, *Chatharuhi: Reviving anime character in reality via large language model*, 2023. arXiv: 2308.09597 [cs.CL].

[5] A. Naeiji, A. An, H. Davoudi, M. Delpisheh, and M. Alzghool, "Question generation using sequence-to-sequence model with semantic role labels," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 2830–2842. [Online]. Available: https://aclanthology.org/2023.eacl-main.207.

[6] S. Wei, W. Lu, X. Peng, S. Wang, Y.-F. Wang, and W. Zhang, *Medical question summarization with entity-driven contrastive learning*, 2023. arXiv: 2304.07437 [cs.CL].

[7] C. Raffel, N. Shazeer, A. Roberts, *et al.*, *Exploring the limits of transfer learning with a unified text-to-text transformer*, 2020. arXiv: 1910.10683 [cs.LG].

[8] M. Lewis, Y. Liu, N. Goyal, *et al.*, *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, 2019. arXiv: 1910.13461 [cs.CL].

[9] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL].

[10] S. Thakur, B. Ahmad, H. Pearce, *et al.*, *Verigen: A large language model for verilog code generation*, 2023. arXiv: 2308.00708 [cs.PL].

[11] M. Flor and B. Riordan, "A semantic role-based approach to open-domain automatic question generation," in *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 254–263. DOI: 10.18653/v1/W18-0530. [Online]. Available: https://aclanthology.org/W18-0530.

[12] D. Lindberg, F. Popowich, J. Nesbit, and P. Winne, "Generating natural language questions to support learning on-line," in *Proceedings of the 14th European Workshop on Natural Language Generation*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 105–114. [Online]. Available: https://aclanthology.org/W13-2114.

[13] F. J. Muis and A. Purwarianti, *Sequence-to-sequence learning for indonesian automatic question generator*, 2020. arXiv: 2009.13889 [cs.CL].

[14] S. Yanan, T. Yanxin, F. Fangxiang, Z. Chunping, and W. Xiaojie, "Category-based strategy-driven question generator for visual dialogue," English, in *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, Huhhot, China: Chinese Information Processing Society of China, Aug. 2021, pp. 1000–1011. [Online]. Available: https://aclanthology.org/2021.ccl-1.89.

[15] D. Xiao, H. Zhang, Y. Li, *et al.*, *Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation*, 2020. arXiv: 2001.11314 [cs.CL].

[16] T. Tang, J. Li, Z. Chen, *et al.*, *Textbox 2.0: A text generation library with pre-trained language models*, 2022. arXiv: 2212.13005 [cs.CL].

[17] W. Qi, Y. Yan, Y. Gong, *et al.*, *Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training*, 2020. arXiv: 2001.04063 [cs.CL].

[18] H. Bao, L. Dong, F. Wei, *et al.*, *Unilmv2: Pseudo-masked language models for unified language model pre-training*, 2020. arXiv: 2002.12804 [cs.CL].

[19] S. Back, A. Kedia, S. C. Chinthakindi, H. Lee, and J. Choo, "Learning to generate questions by learning to recover answer-containing sentences," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online: Association for Computational Linguistics, Aug. 2021, pp. 1516–1529. DOI: 10.18653/v1/2021.findings-acl.132. [Online]. Available: https://aclanthology.org/2021.findings-acl.132.

[20] Z. Wu, X. Jia, F. Qu, and Y. Wu, *Enhancing pre-trained models with text structure knowledge for question generation*, 2022. arXiv: 2209.04179 [cs.CL].

[21] L. Dong, N. Yang, W. Wang, *et al.*, *Unified language model pre-training for natural language understanding and generation*, 2019. arXiv: 1905.03197 [cs.CL].

[22] X. He, S. Zannettou, Y. Shen, and Y. Zhang, *You only prompt once: On the capabilities of prompt learning on large language models to tackle toxic content*, 2023. arXiv: 2308.05596 [cs.CL].

[23] J. Li, L. Yu, and A. Ettinger, *Counterfactual reasoning: Testing language models' understanding of hypothetical scenarios*, 2023. arXiv: 2305.16572 [cs.CL].

[24] J. Wang, Y. Liang, F. Meng, *et al.*, *Is chatgpt a good nlg evaluator? a preliminary study*, 2023. arXiv: 2303.04048 [cs.CL].

[25] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. [Online]. Available: https://aclanthology.org/D16-1264.

[26] M. S. Schlichtkrull and W. Cheng, *Evaluating for diversity in question generation over text*, 2020. arXiv: 2008.07291 [cs.CL].