Let the LLMs Talk: Simulating Human-to-Human Conversational QA via Zero-Shot LLM-to-LLM Interactions

Zahra Abbasiantaeb
University of Amsterdam
Amsterdam, The Netherlands
z.abbasiantaeb@uva.nl

Yifei Yuan
The Chinese University of Hong Kong
Hong Kong, Hong Kong SAR
yfyuan@se.cuhk.edu.hk

Evangelos Kanoulas
University of Amsterdam
Amsterdam, The Netherlands
e.kanoulas@uva.nl

Mohammad Aliannejadi
University of Amsterdam
Amsterdam, The Netherlands
m.aliannejadi@uva.nl

ABSTRACT

Conversational question-answering (CQA) systems aim to create interactive search systems that effectively retrieve information by interacting with users. To replicate human-to-human conversations, existing work uses human annotators to play the roles of the questioner (student) and the answerer (teacher). Despite its effectiveness, challenges exist as human annotation is time-consuming, inconsistent, and not scalable. To address this issue and investigate the applicability of large language models (LLMs) in CQA simulation, we propose a simulation framework that employs zero-shot learner LLMs for simulating teacher–student interactions. Our framework involves two LLMs interacting on a specific topic, with the first LLM acting as a student, generating questions to explore a given search topic. The second LLM plays the role of a teacher by answering questions and is equipped with additional information, including a text on the given topic. We implement both the student and teacher by zero-shot prompting the GPT-4 model. To assess the effectiveness of LLMs in simulating CQA interactions and understand the disparities between LLM- and human-generated conversations, we evaluate the simulated data from various perspectives. We begin by evaluating the teacher's performance through both automatic and human assessment. Next, we evaluate the performance of the student, analyzing and comparing the disparities between questions generated by the LLM and those generated by humans. Furthermore, we conduct extensive analyses to thoroughly examine the LLM performance by benchmarking state-of-the-art reading comprehension models on both datasets. Our results reveal that the teacher LLM generates lengthier answers that tend to be more accurate and complete. The student LLM generates more diverse questions, covering more aspects of a given topic.

CCS CONCEPTS

• **Information systems** → Evaluation of retrieval results.

WSDM '24, March 4–8, 2024, Merida, Mexico
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635856

KEYWORDS

Dialogue simulation, Conversational question answering, Large language models.

ACM Reference Format:
Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2024. Let the LLMs Talk: Simulating Human-to-Human Conversational QA via Zero-Shot LLM-to-LLM Interactions. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3616855.3635856

1 INTRODUCTION

Over the years, the information retrieval (IR) community has strived to create an interactive and iterative search system that effectively retrieves information [4, 9, 13, 21]. Recent advancements in conversational question-answering (CQA) systems have been successful in achieving this goal by retrieving relevant information and engaging in back-and-forth interactions with users to fully understand their information needs [34, 38]. Under this case, existing work captures the iterative dynamics of conversations, where a set of annotators play the role of the questioner (student) and the answerer (teacher) over a pre-defined search topic [11, 24, 49].

Despite the effectiveness of previous efforts in this task, several drawbacks exist. One major challenge is the maintenance of a large team of annotators to generate a substantial number of conversations. This process can be time-consuming, resource-intensive, and expensive. Additionally, relying solely on human annotators may introduce variations in the quality and consistency of the generated conversations. Also, in many cases, the human student cannot effectively explore a given topic that is out of their background knowledge. For example, a person who has expertise in geography can better explore a related topic rather than a person who does not. In contrast, large language models (LLMs) can leverage their vast background knowledge to effectively play the role of a geography expert in a conversation. Therefore, it is crucial to explore automated approaches that can generate simulated conversations, reducing the dependency on human annotators and making the process more efficient and scalable.

User simulation is an important emerging research frontier for conversational search development and evaluation [7, 28], where the focus mainly is on simulating the user behavior under a certain condition, such as responding to system's actions [47], answering clarifying questions [42], and giving feedback on system answer [28]. The main drawback of existing research on user simulation is its reactive nature, where the simulated user just passively respond to the system's utterance. In real-world scenarios, however, users' actions are a mix of proactive and reactive actions, initiating and frequently guiding conversations by posing questions that stem from their underlying information need.

In this work, we aim to explore LLMs' effectiveness in simulating a proactive user, exploring a pre-defined topic in a conversational setting. To this aim, we replicate the `teacher–student` conversational simulation adopted by Choi et al. [11] while replacing both human parties with LLMs, enabling us to effectively evaluate and compare the performance of LLMs with human annotators. This leads us to our first research question, **RQ1:** *how can we employ LLMs to generate such simulated conversations effectively and automatically?* We answer this question by proposing a zero-shot LLM-to-LLM simulation framework where the `student` LLM aims to explore a topic by posing various questions and the `teacher` LLM's goal is to provide complete and correct answers to the questions. We implement both the `student` and `teacher` by zero-shot prompting GPT-4 [27].

The usage of LLMs in this setting leads us to the next research questions **RQ2:** *how can we evaluate the role of LLMs in CQA simulation?* and **RQ3:** *how do LLM- and human-generated conversations compare?* To address these questions:

(i) We first conduct an extensive independent evaluation of the `teacher`, measuring its effectiveness in this task. To this aim, we conduct an extensive human evaluation task where the annotators compare LLM- and human-generated answers on the same questions side by side.
(ii) We then evaluate the performance of the `student`. To this aim, we compare the patterns and question-asking behavior of the LLM and human from various perspectives, discovering interesting patterns. For example, we find that LLM-generated questions lead to more topical coverage.
(iii) Finally, we conduct extensive analyses to thoroughly examine the performance of the LLM by benchmarking state-of-the-art reading comprehension models on both datasets.

We find that LLM-generated answers are generally lengthier and more comprehensive. Also, they are more consistent and fluent. Moreover, our human evaluation reveals that the LLM `teacher` is more accurate in providing correct answers. Upon benchmarking state-of-the-art reading comprehension models, we find that pre-trained models exhibit more effective performance on LLM-generated data. This efficacy may result from certain biases in the generated conversations and the enhanced consistency within it.

Overall, our contributions can be summarized as follows:

- We leverage LLMs to mimic human-to-human interaction in a CQA setting using zero-shot prompting. We prompt two LLMs to conduct `teacher–student` simulation and propose an LLM-generated dataset, called SimQuAC (Code and data available at: https://github.com/ZahraAbbasiantaeb/SimQUAC.git).
- We propose and perform a comprehensive automatic and human evaluation framework, as well as linguistic analysis to evaluate the LLM's effectiveness in this setting on the `teacher` and `student` level.

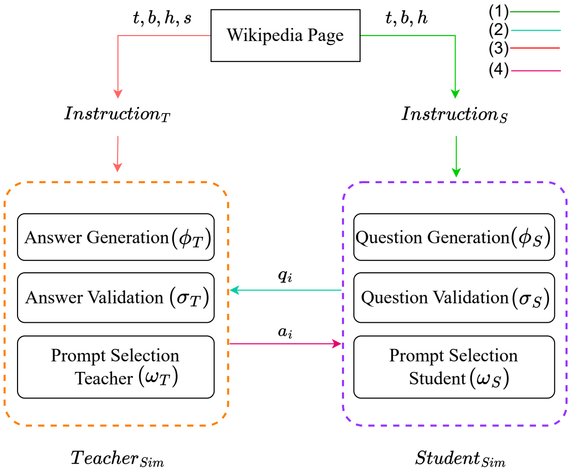

t, b, h, s
Wikipedia Page
t, b, h
(1)
(2)
(3)
(4)
$Instruction_T$
$Instruction_S$
Answer Generation (ϕ_T)
Answer Validation (σ_T)
Prompt Selection Teacher (ω_T)
Question Generation (ϕ_S)
Question Validation (σ_S)
Prompt Selection Student (ω_S)
q_i
a_i
$Teacher_{Sim}$
$Student_{Sim}$

Figure 1: A high-level view of the architecture of our Simulation framework.

- We conduct extensive analyses on LLM- and human-generated conversations, discovering many interesting patterns exhibited by humans and LLMs during CQA.

2 METHODOLOGY

2.1 Problem Setting

Our experimental setup involves simulating an information-seeking conversation, where a `student` interacts with a `teacher` in a question-answering conversation. Based on that, we adopt the setting established by the **QuAc!** (**QuAc!**) dataset [11], which serves as a widely recognized benchmark for evaluating the effectiveness of CQA models. The dataset revolves around discussions based on Wikipedia articles. It consists of conversation contexts where a crowdworker plays the role of a questioner (`student`) and engages in a conversation with another crowdworker who acts as an answerer (`teacher`). Specifically, the `teacher` is given access to the entire Wikipedia section article and aims to generate responses to questions posed by the `student`. To ensure fairness, the `teacher`'s responses are limited to selecting the appropriate answer span from the article. In contrast, the `student` is only provided with the article's title and tries to use this limited information to ask relevant questions and explore the topic. As the `students` engage in the conversation, they explore the topic by asking questions, and the conversation unfolds accordingly.

2.2 Task Formulation

As is mentioned in Section 2.1, the conversation evolves around a Wikipedia article titled t. The `student` is only provided with limited access to information, including the section header h as information need and the first paragraph of the main article b, which serves as the background information. The `teacher`, on the other hand, has access to the additional information, including the full text of the section s. The conversation begins when the `student` raises an initial question q_0 and the teacher provides an answer, denoted as a_0. After receiving the answer from `teacher`, `student` continues to ask more questions until some stoppage criteria are met. Specifically, following previous work [11, 37, 49], instead of answering with free-text, the `teacher` must select one or several contiguous spans from text as the answer. Note that although limiting the LLM to select text spans restricts the `teacher`'s ability to freely provide answers, it offers the advantage of simplified answer evaluation and prevents hallucination. This setting enables us to examine the proficiency of LLMs in tasks like CQA and reading comprehension (RC) by comparing their performance against existing methods.

2.3 Model Framework Overview

In order to have a better understanding of **RQ1**, we propose a LLM-based framework. Figure 1 illustrates the overall architecture of our model, showcasing the interactions between the two LLMs. The entire process evolves around a Wikipedia page. The purple box on the right plots the simulated `student`, named $student_{Sim}$, while the orange box on the left plots the simulated `teacher`, named $teacher_{Sim}$. The $teacher_{Sim}$ and $student_{Sim}$ contain several components to generate acceptable answers and questions, respectively.

The process of generating the conversation starts with initializing $student_{Sim}$ by giving the instruction prompt $Instruction_S$. The $Instruction_S$ prompt aims to guide the `student` LLM $student_{Sim}$ in generating the first question q_0 in the *question generation* component (ϕ_S). Then, we pass the generated question q_0 to the *question validation* component (σ_S). This component plays a critical role in ensuring the structural integrity of the generated questions. If it determines that the structure of the question is not acceptable, $student_{Sim}$ will prompt ϕ_S again to regenerate q_0. After that, we forward q_0 to $teacher_{Sim}$, which concatenates it with the instruction prompt $Instruction_T$, forming a combined input. This combined input is then fed to the *answer generation* component (ϕ_T) for generating the answer a_0. To ensure that the generated answers adhere to our defined setting (i.e., corresponding to one or multiple segments in the section text s), an *answer validation* (σ_T) component is leveraged to check the validity of a_0. If a_0 is determined as an invalid answer, a *prompt selection teacher* component ω_T will select the appropriate prompt (p_T) and pass it to ϕ_T to regenerate the answer a_0. This step continues when a_0 is determined to be valid by σ_T, it is passed back to $student_{Sim}$.

Similarly, $student_{Sim}$ incorporates the *prompt selection for student* component (ω_S) to select the optimal prompt p_S for generating the subsequent question q_i. Once chosen, it transfers p_S to the question generation ϕ_S module again, where p_S is then employed to generate the next question q_i. This back-and-forth question–answering process continues until the stoppage criteria are met. In each turn, the generated question q_i and answer a_i will be stored. Algorithm 1 shows the detailed simulation process of our model. In the following sections, we will provide detailed explanations of each component to further elucidate its functionality.

2.4 Teacher Simulation

Answer generation (ϕ_T). This component belongs to $teacher_{Sim}$ and is initialized with $Instruction_T$ in a zero-shot manner. The $Instruction_T$ includes the instruction to copy the exact spans from s to answer the given question and some information about the Wikipedia page including the title t, background b, and section text s. We instruct $teacher_{Sim}$ to generate the sentence *"I cannot find the answer."* when s does not contain the answer. Additionally, to prevent the generation of excessively long answers that could potentially impede readability, we implement a two-step mechanism to control the length of the generated answers: (i) we specify in the prompt that the selected span should not exceed a maximum of 40 tokens; (ii) we include the statement *"Remember that you should select the shortest possible span from the text,"* at the end of each question, making $teacher_{Sim}$ itself decide on the length of the sentence within the maximum limit.

Algorithm 1: Data simulation algorithm

```
Data: $t$, $b$, $s$, $h$, $N$, $patience$
Result: $q_{0...N}$, $a_{0...N}$
$Instruction_S \leftarrow$ StudentInitialPrompt($t$, $b$, $h$);
$Instruction_T \leftarrow$ TeacherInitialPrompt($t$, $b$, $h$, $s$, $q_0$);
$i \leftarrow 0$ ;
while $i < N$ do
    $m \leftarrow$ patience ;
    if $i == 0$ then
        $q_i \leftarrow \phi_S$ ($Instruction_S$) ;          /* student_Sim */
    else
        $p_S \leftarrow \omega_S$ ($a_i$);
        $q_i \leftarrow \phi_S$ ($p_S$);
    end
    while $\sigma_S$ ($q_i$) is False do
        $p_S \leftarrow$ updatePromptToAskShortQuestion($p_S$);
        $q_i \leftarrow \phi_S$ ($p_S$);
    end
    if $i == 0$ then
        $a_i \leftarrow \phi_T$ ($Instruction_T$) ;          /* teacher_Sim */
    else
        $a_i \leftarrow \phi_T$ ($q_i$)
    end
    while $\sigma_T$ ($a_i$, $b$, $s$) is False and $m > 0$ do
        $p_T \leftarrow \omega_T$ ($a_i$, $b$, $s$);
        $a_i \leftarrow \phi_T$ ($p_T$);
        decrement m;
    end
    increment $i$;
end
```

Answer validation & regeneration (σ_T). Rather than solely rely on the instruction prompt $Instruction_T$ for one-time answer generation, we adopt an iterative model σ_T to validate and refine the generated answers in succession to ensure they are in line with the request of $Instruction_T$. This component serves as a reminder to ϕ_T of the validation criteria and prompts `teacher` to generate an answer that aligns with the given section.

We define that a *valid answer* (a_i) should include exact copies of contiguous spans in the section text s, or it should be the phrase *"I cannot find the answer,"* if the question (q_i) cannot be answered from the text. Therefore, we verify an answer's validity based on two criteria: (i) *whether a_i contains one or multiple exact copies of the text spans in s or being "I cannot find the answer"*; and (ii) *whether a_i is copied from the text section s, rather than the background b.*

Table 1: The template for constructing *Instruction$_T$* (left side) and the *Instruction$_S$* (right side). The variables inside "[ ]" would be filled based on the input Wikipedia page.

Instruction$_T$	*Instruction$_S$*
Topic: [t] Background knowledge [b] In this task, you will be given a text about the topic explained above. You will answer my questions from this text. Please remember that you cannot generate the answer on your own but should only copy a continuous span from the original text and the copied answer should not exceed 40 tokens. If you cannot find the answer in the text, please generate 'I cannot find the answer'. Section header: [h] Section text: [s]	In this task, I am a teacher and have a document, you are a curious student who wants to explore this document by asking questions. The main objective is to learn most of the documents that I have. I will explain to you the topic and background knowledge of the document. Then I will give you the title of the document and you should ask questions about this title one by one. When you ask a question, I give you the answer, and then you ask your next question. I'm only allowed to find the answer to your questions from this document, so if I cannot find the answer, I will say "I cannot find the answer, please ask your next question". You shouldn't ask questions that can be answered from my previous answers to your previous questions. You should sometimes ask follow-up questions from my previous answers. Topic: [t] Background knowledge [b] Please start asking question about: [h]

We follow the steps below to address the two validation criteria. First of all, to address criterion (i), we conduct a simple text search and see if a_i (or each sentence of a_i) is from s. Notably, we notice that most of the time LLMs do not copy the texts inside the brackets and neglect the extra white spaces within the text. Therefore, we generate two normalized versions of s by (i) removing the extra white spaces and (ii) texts inside the brackets. If a_i is not found in s and its normalized versions, we issue a second prompt (p_T) *'Please copy the answer exactly from the given text,"* reminding ϕ_T where it has failed. To address criterion (ii), we also perform a text search to see if a_i is selected from b. In such a case, we issue a second prompt (p_T) *"Please answer from the given section not the given background description,"* to remind ϕ_T of this criterion. We continue these steps until the generated answer satisfies both validation criteria.

Finally, once the valid a_i has been confirmed by σ_T, it is passed on to *student$_{Sim}$*, which utilizes a_i to formulate the next question (q_{i+1}) in the conversation. However, there are cases where the loop continues for an excessive number of iterations. We terminate the loop in such cases, assuming that *teacher$_{Sim}$* fails in finding the answer from s, or the question is not answerable. Similar to QuAC, we set the answer to such questions to "*I cannot find the answer.*" This is necessary to prevent an infinite loop and ensure that the system remains efficient and responsive.

2.5 Student Simulation

Question generation (ϕ_S). To simulate the student, we prompt the Question Generation ϕ_S component of *student$_{Sim}$* in a zero-shot manner. With *Instruction$_S$*, we instruct *student$_{Sim}$* to explore the given information (h and b) by posing questions, under the assumption that it does not possess knowledge of s. As shown in Table 1, we include the topic t and b as well as the section header h in *Instruction$_S$* to ensure that *student$_{Sim}$* has some basic knowledge about the given topic.

Question validation (σ_S). To ensure that an LLM-generated question q_i is structurally sound, we employ a validation step called σ_S. This component serves the purpose of verifying and validating the syntactical correctness and coherence of the generated question. We observe that while q_i is supposed to be exactly one question in our setting, sometimes the LLM tends to generate multiple questions in one go. To address this issue, we consider a question valid if it adheres to the following criteria: (i) it should not exceed 25 words in length and (ii) should not contain a newline character or enumerated items (e.g., 1, 2, 3). This simple yet effective validation helps to filter lengthy and intricate questions, including those containing multiple sub-questions.

Prompt selection for student (ω_S). As the conversation progresses, there may be instances where the generated question q_i remains unanswered from the given text (s) despite being relevant to the information need (h) and topic (t). For instance, students tend to ask very specific follow-up questions that cannot be answered from s (e.g., "Was Newsom's mayoralty generally well-received by the citizens of San Francisco?"). To address this issue, it is crucial to continuously assess the ability of the teacher simulator to answer the generated question q_i and make necessary adjustments to the student prompt p_S to enhance the quality of the question. The refined p_S aids the generative component ϕ_S in generating questions that can be answered from the given information s. For instance, if the response a_i is "I cannot find the answer," there is a higher chance that the subsequent question q_{i+1} might be overly specific and cannot be answered directly from s. To solve this issue, ω_S randomly selects one of the following guiding prompts as p_S and passes it to ϕ_S. These guiding prompts include: (i) Ask a general question and do not ask a too specific question; (ii) Ask a question starting with where, when, or who; (iii) Ask a question about what is interesting in this article; (iv) Ask a question about another aspect of the topic. By utilizing these guiding prompts, we can effectively prevent the generation of overly specific questions and guide *student$_{Sim}$* by offering additional clues and information. This approach allows for more efficient exploration of the given information need (h) by the *student$_{Sim}$*, ultimately enhancing its overall understanding.

Table 2: Examples of cases where answers generated by *teacher*$_{Sim}$ win the original QuAC answers in each aspect.

Correctness
1) How old was he when he went on pilgrimage? *answer*$_{QuAC}$: In 1897, *answer*$_{Sim}$: He was twenty-eight, had been married ten years, and had an infant son with another child on the way.
Completeness
2) What shows did David Frost have? *answer*$_{QuAC}$: Sunday morning interview programme Breakfast *answer*$_{Sim}$: Sunday morning interview programme Breakfast; Through the Keyhole; Al Jazeera English.
Naturalness
3) Did he perform in the later 30s? *answer*$_{QuAC}$: Agency. Mills though continued to record Ellington *answer*$_{Sim}$: In 1937, Ellington returned to the Cotton Club which had relocated to the mid-town Theater District.

3 TEACHER EVALUATION

In this section, we describe our experimental methodology to evaluate the performance of *teacher*$_{Sim}$ from various perspectives, which addresses **RQ2** and **RQ3** from the `teacher` perspective. Firstly, we describe the data source to perform `teacher` evaluation. We then introduce the human evaluation process of the `teacher`, with a particular focus on assessing the generated answers by comparing them against human-generated answers.

3.1 Experimental Setup

Data for evaluating *teacher*$_{Sim}$. To simulate the `teacher` and ensure a fair comparison between the LLM- and human-generated answers, we maintain consistency in the conversation topic and questions across the comparison. In detail, we randomly select 50 conversations from the training set of QuAC [11]. From each conversation in the sampled data, we borrow the topic information and all associated questions. Following this, we pass the questions to our *teacher*$_{Sim}$ to generate the answers and then compare them with the original answers from QuAC.

Parameters. In our experiment, we adopt GPT-4 as our base `teacher` and `student` LLM. In our preliminary experiments, we explored using other LLMs such as GPT-3.5 and LLaMA [48] as `teacher`. However, we found that only GPT-4 can copy an exact segment of the text in a zero-shot manner (we later discuss it as a direction for future work in Section 6). Other models failed in this task by either generating broken or free-text sentences that did not satisfy our requirements. In our model, we set the patience parameter of σ_T to a fixed value of 4, which means the `teacher` validation loop breaks after a maximum number of 4 iterations.

Human evaluation. To evaluate the performance of the `teacher` in our task, we conduct human evaluation on a professional crowdsourcing platform Prolific.[1] We ask the crowd-workers to compare the answers generated by *teacher*$_{Sim}$ (i.e., *answer*$_{Sim}$) with the answers of QuAC (i.e., *answer*$_{QuAC}$) in terms of *correctness*, *completeness*, and *naturalness*. We explain each aspect in detail:

- *Correctness* aims to determine whether the selected text span accurately serves as a correct answer to the question, based on the context of the conversation.
- *Naturalness* measures the fluency and human-likeliness of a text span. Although both QuAC and *teacher*$_{Sim}$ contain a selected text span as a response, we observe that in many cases of QuAC, the selected spans are unnatural and do not form complete sentences.
- *Completeness* measures whether the provided answer is complete and comprehensive. It is important to note that an answer can be correct but incomplete. For example, if the question is about the albums of an artist, a more complete answer is the one that lists more albums, if not all.

Additionally, we ask the crowd-workers to indicate which system (human in QuAC vs. *teacher*$_{Sim}$) they would *prefer* to interact with by providing a short justifying, aiming to capture the overall quality of the generated data in a conversation.

Crowdsourcing task design. We design a crowdsourcing task accordingly for the assessment between two conversations. The annotators begin by comparing the responses from both systems for each question. We display the background information (b) and the section text (s) on the left side of the page. On the right side, we include each question along with the simulated answer (*answer*$_{Sim}$) and the original QuAC answer (*answer*$_{QuAC}$). For each annotation aspect, we ask the annotators to indicate which system is better by choosing from the four options, namely, "System A," "System B," "Neither A nor B," and "Both A and B." The annotators can easily locate the selected text spans by clicking on the answers. The text will be highlighted in s, enabling them to compare the two spans efficiently and easily. Note that we do not ask the annotators to evaluate the questions when the answers from *answer*$_{QuAC}$ and *answer*$_{Sim}$ are identical. However, we still include them in the interface as they can contribute to the context of the conversation. Also, when one of the answers is "I cannot find the answer," we only ask the annotators to evaluate its correctness, as other metrics cannot be evaluated for these cases.

Annotation and quality check. We randomly sampled 50 conversations from two datasets and divided them into 10 batches, each containing five conversations for evaluation. To ensure reliable assessments, we have a minimum of three crowd-workers evaluate each conversation independently. We consider one system to have *won* over the other when the majority of the crowd-workers choose it. However, we acknowledge that there may be instances where the two systems perform equally well. In such cases, no system receives the majority vote, leading to a *tie*. In Table 2, we provide several cases when the LLM answer *answer*$_{Sim}$ wins the human answer *answer*$_{QuAC}$ under different aspects.

To avoid any position bias in the annotations, both QuAC and *teacher*$_{Sim}$ examples are randomly switched and positioned as System A and System B for each conversation. Also, to ensure English proficiency, we made the task visible only to native English speakers. Additionally, before starting the annotation task, we asked the crowd-workers to complete an onboarding test, consisting of some

[1] https://prolific.co

Table 3: Statistics of comparison on $answer_{QuAC}$ and $answer_{Sim}$ under different conditions based on their answer span and type. The "I cannot find the answer" answers are represented by 'None.' We refer to the single-span answers generated by $answer_{Sim}$ by 'single'.

Ans. Span	Condition	Count	Total
Overlap	$answer_{Sim}$ is single	87	106 (29.5%)
	$answer_{Sim}$ is not single	19	
Different	$answer_{QuAC}$ = None AND $answer_{Sim}$!= None	18	176 (49.0%)
	$answer_{Sim}$ = None AND $answer_{QuAC}$!= None	54	
	$answer_{Sim}$ is single	85	
	$answer_{Sim}$ is not single	19	
Same	$answer_{Sim}$ = None AND $answer_{QuAC}$ = None	41	77 (21.4%)
	$answer_{Sim}$ is single	33	
	$answer_{Sim}$ is not single	3	

questions about the task itself (e.g., (i) What does "System A is correct" mean? and (ii) Is a correct answer always natural?). We also provided around 10 sample annotations for the crowd-workers to refer to. Upon completion, we evaluated their responses, and only if they answered at least 75% of the onboarding questions correctly, they were allowed to start the main annotation task. This approach helps to guarantee that crowd-workers are adequately prepared and knowledgeable before undertaking the annotation tasks. Moreover, we manually check the consistency of preference justifications with the labels by reading their open comments. We noticed that in some cases (7%) they do not match, so we removed them from our dataset.

3.2 Experimental Results

In this section, we evaluate the performance of $teacher_{Sim}$.

Answer comparison: QuAC vs. $teacher_{Sim}$. We report the performance on 359 questions extracted from the 50 sampled conversations. For 77 questions $answer_{Sim}$ is identical to $answer_{QuAC}$. Furthermore, for 106 of the questions, there is an overlap between $answer_{Sim}$ and $answer_{QuAC}$, indicating that one is a substring of the other. For 176 questions, $answer_{QuAC}$ and the $answer_{Sim}$ do not overlap. Notably, for 41 questions, $teacher_{Sim}$ returns more than one segment from the text as the answer. The statistics on comparison of $answer_{QuAC}$ and $answer_{Sim}$ can be found in Table 3.

Answer-level human evaluation. We report the result of $teacher_{Sim}$ human evaluation (Fleiss' $\kappa = 0.4365$) in Table 4. The result shows that $teacher_{Sim}$ outperforms the human `teacher` of QuAC in terms of all question-based metrics by a large margin. Additionally, we see that the annotators prefer the answers provided by our $teacher_{Sim}$ over the $answer_{QuAC}$ in 87.7% of the topics. $teacher_{Sim}$ answers exhibit enhanced accuracy and naturalness due to a significant number of incomplete answer spans in QuAC. This leads to grammatically incorrect sentences (e.g., "platinum. Thank U, the'," "Tori Amos on the 5 and a Half Weeks"). It is also noteworthy that we allow $teacher_{Sim}$ to select multiple spans from the text to provide more complete answers to the questions, when necessary — something that is missing in QuAC training set, but only available in the test set [11]. Furthermore, as in our task, we limit the LLM to answer questions from the given text, the risk of hallucination is highly decreased and is indeed verifiable, making LLMs more reliable. These findings are in line with Faggioli et al. [15], showing the potential of LLMs in replacing crowdsourcing in annotation and simulation tasks, addressing the concerned research question **RQ3**.

Table 4: Results of the pairwise human evaluation of answers generated Teacher simulation $answer_{Sim}$ compared to original $answer_{QuAC}$ answers. Each cell reports the percentage of cases where the three human annotators agreed that either $answer_{QuAC}$ or $answer_{Sim}$ wins. We also report the percentage of ties, where the annotators disagreed on a winner.

Annot. level	Metric	$answer_{QuAC}$	$answer_{Sim}$	Tie
Question	Correctness	11.31%	**38.6%**	50.0%
	Naturalness	7.1%	**42.1%**	50.7%
	Completeness	5.26%	**53.8%**	40.8%
Conversation	Preference	6.12%	**87.7%**	6.18%

Conversation-level human evaluation. We further compute *Preference* as reported by the annotators, who indicate their preference for interacting with either of the two systems. We see in Table 4 that $answer_{Sim}$ is the winner in terms of conversation-level annotator preference, where 87.7% of them prefer $answer_{Sim}$ over $answer_{QuAC}$. This indicates the promising potential of LLMs in engaging in a conversation, as long as they are sufficiently informed about the task and certain verification steps are employed.

We follow Siro et al. [43] and cluster the open-ended justifications provided by the annotators into different categories to gain more insights into other aspects of quality that can be overlooked in our human evaluation. In our analysis, we find that most of the comments mention the three aspects that we include in our annotation task (i.e., correctness, naturalness, and completeness). We also find comments that can be classified into *seven* new categories, namely, *clarity*, *coherency*, *directness*, *being comfortable*, *trustworthiness*, *factuality*, and *conciseness.* We see that many annotators found $answer_{Sim}$ answers more factual and found the conversations more comfortable. Interestingly, even though $answer_{Sim}$ answers are lengthier on average, some annotator justified their preference because of their conciseness.

4 SIMULATION EVALUATION

To provide a more comprehensive evaluation of **RQ2** and **RQ3**, we assess the performance of the LLM simulation in comparison to human performance. We first introduce an LLM-based simulated dataset SimQuAC. Furthermore, we report the results of state-of-the-art reading comprehension methods on the two datasets to shed light on the quality and difficulty of the simulated dataset.

4.1 SimQuAC Dataset

We first introduce our dataset named SimQuAC for simulation evaluation, using the simulation framework described in Section 2. To collect SimQuAC we used GPT-4 to implement $student_{Sim}$ and $teacher_{Sim}$. We randomly select 342 conversations from the training set of QuAC and simulate 334 conversations using the unique

Table 5: Statistics of the collected dataset by simulating the conversation with $teacher_{Sim}$ and the $student_{Sim}$.

	QuAC	SimQuAC
# conversations	342	334
# questions	2,498	4,005
# questions with answer	2,062	2,517
Avg. length of the answers	15.33	28.23
Avg. # answers per question	1	1.32

Table 6: The questions from QuAC on "Talland House (1882-1894)" section of the "Virginia Woolf" topic with the questions generated by $student_{Sim}$.

SimQuAC
q_1) Where is Talland House located?
q_2) Did Virginia Woolf live in Talland House during the period of 1882-1894?
q_3) Who owned Talland House during this period of 1882-1894?
q_4) What is the architectural style of Talland House?
q_5) Did any notable events take place in Talland House during the period between 1882-1894?
q_6) What impact did living in Talland House have on Woolf's later work?
QuAC
q_1) What is Talland house?
q_2) What happened at this house?
q_3) Did anything tragic happen?
q_4) What else happened at the house?

topics from this sample. SimQuAC consists of 4,005 questions with an average of 1.32 answer spans per question. The statistics of SimQuAC are presented in Table 5, alongside those of the original QuAC conversations.

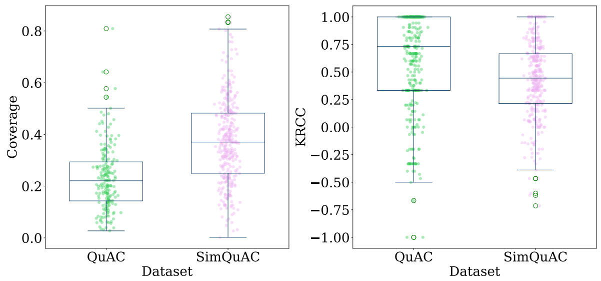
4.2 Student Evaluation

Due to the nature of the `student`'s role in a conversation, which involves asking questions and exploring a topic, it becomes challenging to define an objective metric that determines which model is "better." Therefore, our emphasis lies in highlighting the distinctions between the behavior of two systems by contrasting their linguistic characteristics from various aspects.

Question comparison: QuAC vs. $student_{Sim}$. Table 6 presents a sequential collection of questions in a conversation within both the QuAC and SimQuAC datasets, sharing the same topic. An observation can be made that GPT-4 tends to inquire about more detailed and lengthy questions compared to humans. Additionally, it is worth noting that the human `student` in QuAC ceases asking questions after the fourth one, while the simulated `student` in SimQuAC continues to pose additional queries.

Coverage. We assess the ability of the two `students` to explore a topic by comparing how much of the s is covered by the answers provided to the questions posed. We plot the distribution of coverage in Figure 2a. We observe that SimQuAC questions cover a significantly (two-tailed t-test; p-value < 0.001) larger portion of the text (mean = 0.365; std = 0.163), compared to QuAC (mean = 0.238; std = 0.122), suggesting that careful prompting of LLMs can lead to a diverse and comprehensive set of questions in a conversation.

Conversation flow. Next, we compare the questions posed in terms of how they shape the flow of the conversation. Our objective in this experiment is to evaluate the naturalness of the conversation flow and the smoothness of topic transitions. We hypothesize that a conversation that strictly follows the sequential order of the content in s is less natural. To measure this, we assign an order to the questions based on the positions of their corresponding answers in s. For instance, let us consider questions A, B, and C. To determine their order, we examine the text spans of their respective answers and sort them based on the start position of each answer. In this case, say question B's answer is at the earliest position of s, followed by A and C. Therefore, the question order would be {B, A, C}. To assess the sequential nature of the conversation flow, we compare the order of the questions in the conversation to their order in the document, considering their corresponding answers. In our example, if questions {A, B, C} appear in the conversation in the same order, i.e., A is followed by B and then C, the conversation flow would be considered completely linear.



(a) Topic coverage **(b) Conversation flow**

Figure 2: Comparison between `student` of QuAC and SimQuAC in terms of (a) topic coverage and (b) conversation flow.

To evaluate the degree of correlation between the question order in the conversation and the corresponding answer order, we calculate the Kendall rank correlation coefficient (KRCC) metric [1] for each conversation. KRCC measures the distance between two ranked lists, where a lower value indicates more distance between the two lists. In our case, the lower the value, the less sequential a conversation flow is. Figure 2b plots the distribution of the two datasets in terms of KRCC. We can see that the average value of KRCC is lower for SimQuAC than QuAC, indicating that the `student` of QuAC poses more questions in a sequential order, compared with $student_{Sim}$. This suggests that $student_{Sim}$ tends to explore the topic by jumping from one part to another part. While there is no indication as to which order is more natural, we can see that there is a clear difference in their behavior. It is noteworthy that exhibiting a more random behavior in posing questions can lead to more challenging datasets, as it prevents model learning such a biased behavior of `student`.

4.3 Reading Comprehension Benchmarking

To gain a deeper understanding of the distinctions between the human-generated (QuAC) and the LLM-simulated data (SimQuAC), we utilize several pre-trained discriminative and generative reading comprehension baselines for evaluating the `teacher` model. These models are pre-trained on the SQuAD dataset [37] and we test them on two datasets directly without further fine-tuning. To ensure a fair comparison with the QuAC subset, we impose a limitation on SimQuAC, restricting it to a maximum of 3 questions within a conversation that do not have an answer. Table 7 reports the results

Table 7: Experimental results of reading comprehension models on QuAC and SimQuAC in terms of precision (Pre.), recall (Rec.), F1-measure (F1), and exact match (EM). '-b' refers to the '-base' variant of the models, while '-l' refers to their '-large' variants. All the numbers are shown in percentages.

	QuAC				SimQuAC			
	Pre.	Rec.	F1	EM	Pre.	Rec.	F1	EM
DistilBERT [40]	10.99	7.36	6.70	1.88	15.28	7.59	7.75	1.36
BERT-b [14]	12.87	10.64	8.31	2.38	17.76	10.25	9.16	1.21
BERT-l [14]	24.93	18.16	16.33	4.17	29.74	16.84	16.75	2.26
T5-b [36]	26.69	21.58	18.93	6.28	31.63	18.02	18.13	2.52
T5-l [36]	29.70	**23.45**	**21.26**	**7.27**	**35.63**	20.90	21.24	3.01

of different models in terms of exact match (EM), precision, recall, and F1-measure when testing on the two datasets.

The results demonstrate that, in comparison to QuAC, most models exhibit superior overall performance when tested on SimQuAC. This suggests that the LLM-simulated data may provide a more favorable context for these models, leading to improved results. Furthermore, it is noteworthy that the EM score in SimQuAC is lower compared with QuAC. This discrepancy can be attributed to the fact that, in SimQuAC, the answers generated by LLM tend to cover a longer span compared to the answers in QuAC, posing more challenges for matching. Moreover, there are more questions with no answer in SimQuAC. The pre-trained models always output a span as an answer, instead of predicting no answer, leading to a lower EM measure. Additionally, our observations reveal the superior performance of generative methods, such as T5, compared to discriminative methods, such as BERT. This finding emphasizes the importance of utilizing generative LLMs for this particular task.

5 RELATED WORK

5.1 Conversational Question Answering

CQA requires the ability to correctly interpret a question in the context of previous conversation turns [53]. Under this context, modern CQA systems can be divided into two types: sequential knowledge-based question-answering (KB-QA) agents [12, 20, 39] and conversational machine reading comprehension (CMRC) systems [19, 25, 35, 38, 52]. In sequential KB-QA systems, agents need to search the database for the appropriate information to generate the answer. In this paper, we focus on the CMRC setting, where the conversation revolves around a given article and the answers are typically a span in the given resource. To this end, several datasets such as CoQA [38], FlowQA [19], have been proposed. Among all the CMRC datasets, QuAC contains over 14K crowdsourced QA dialogues [11]. This dataset allows for a `student` posing a sequence of free-form questions to learn as much as possible about a hidden Wikipedia article and a `teacher` is hired to find the answer to each question in the text. Following this line, extensive tasks such as reading comprehension [19, 32, 35, 55], answer ranking [34], question generation [22, 51] are adopted to measure the performance on both `student` and `teacher` levels.

5.2 User Simulation

While user simulators have been studied in the information retrieval (IR) community extensively [8, 10, 26], including applications such as simulating user satisfaction for the evaluation of task-oriented dialogue systems [44] and recommender systems [2, 54], they are often limited to reacting to a system's action. The emergence of LLMs provides the opportunity to improve user simulation, making it more realistic. LLMs are ideal for human simulations due to their remarkable ability to process text in the natural language format. They are also able to generate coherent and contextually appropriate language that is very similar to how humans communicate [29, 30, 56]. One such application is using LLMs as evaluators to mimic human evaluation, which has proven to be highly effective in various contexts [3, 16, 31, 33, 41, 57]. For instance, Guo et al. [16] compare ChatGPT with human experts by collecting tens of thousands of comparison responses from both sources. Tan et al. [45] assess the performance of ChatGPT as a KB-QA system using its own knowledge. Another common application of LLMs in simulation is leveraging them as an annotator-free tool for data augmentation [5, 6, 17, 18, 23, 46, 50]. Sekulic et al. [42] employ GPT-2 and propose an evaluation framework based on mixed-initiative conversations. Owoicho et al. [28] take it one step further and utilize GPT-3.5 to simulate a user that can also provide feedback on the relevance of a returned document in a conversational search setting. In the context of document re-ranking, Askari et al. [5] prompt LLMs to generate synthetic training data for cross-encoder re-rankers. Most recently, Hu et al. [17] adopt LLMs as user simulators in a task-oriented dialogue system. To the best of our knowledge, our work is the first to utilize LLMs as annotator-free `teacher-student` simulators in a CQA system, where the `student` takes a proactive role in exploring a topic.

6 CONCLUSIONS AND FUTURE WORK

We explore simulating human-to-human conversations using zero-shot prompting of LLMs in a CQA setting. Our framework involves two GPT-4s interacting on a topic: one as the student generating questions based on background knowledge, and the other as the teacher seeking answers within a text on the given topic. To assess the system, we initially evaluate the `teacher`'s performance through both automated methods and human assessment. Subsequently, we compare conversations generated by the LLM and those by humans for the student-level evaluation. In summary, our investigation highlights the potential of LLMs in facilitating interactive and informative retrieval experiences.

Despite the superiority of our model, several limitations persist, which point to avenues for future research. Firstly, according to our findings, only GPT-4 consistently follows instructions to generate reasonable conversations, constraining the overall effectiveness of the pipeline. Besides, language models can exhibit various biases, when used for such simulation. It therefore becomes essential to further develop methods to mitigate these biases. Moreover, although we have devised prompting strategies to mimic human interaction, the manual construction of instructions can be time-consuming. Future work should explore more advanced and efficient automatic prompting strategies to enhance the system.

7 ETHICAL CONSIDERATIONS

Our work revolves around utilizing LLMs to simulate users. Given the recent emergence of LLMs and the vast interest in using them for various research directions, we believe that pursuing such a direction is necessary as it unveils the potential of LLMs, while at the same time exhibiting their potential ethical considerations. Below we list some of these concerns that need to be considered and addressed in this research area:

- Bias and discrimination: LLMs are biased towards their training data. The simulated data could in turn carry the same biases and further propagate stereotypes and discrimination.
- Misrepresentation: using an LLM to simulate users would introduce certain biases on the type of users being represented. The biases that exist in the data that the LLM is trained on would be reflected in the simulated user.
- Transparency and accountability: the decision-making process within LLMs can be opaque, making it challenging to understand how or why a particular simulated conversation is generated. This lack of transparency can lead to ethical challenges, particularly in contexts where clear justification for a decision is required.
- Environmental impact: the training and operation of LLMs consume significant computational resources, contributing to energy consumption and potentially having a negative environmental impact.

While simulating users using LLMs has various advantages, it must be approached with careful consideration of the potential ethical implications.

REFERENCES

[1] Hervé Abdi. 2007. The Kendall Rank Correlation Coefficient. *arXiv* abs/1507.01427.

[2] Jafar Afzali, Aleksander Mark Drzewiecki, Krisztian Balog, and Shuo Zhang. 2023. UserSimCRS: A User Simulation Toolkit for Evaluating Conversational Recommender Systems. *arXiv* abs/2301.05544 (2023).

[3] Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2023. Can we trust the evaluation on ChatGPT? *arXiv* abs/2303.12767 (2023).

[4] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[5] Arian Askari, Mohammad Aliannejadi, E. Kanoulas, and Suzan Verberne. 2023. Generating Synthetic Documents for Cross-Encoder Re-Rankers: A Comparative Study of ChatGPT and Human Experts. *arXiv* abs/2305.02320 (2023).

[6] Arian Askari, Mohammad Aliannejadi, Evangelos Kanoulas, and Suzan Verberne. 2023. A Test Collection of Synthetic Documents for Training Rankers: ChatGPT vs. Human Experts. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management.*

[7] Krisztian Balog. 2021. Conversational AI from an Information Retrieval Perspective: Remaining Challenges and a Case for User Simulation. In *Proceedings of the Second International Conference on Design of Experimental Search & Information REtrieval Systems*, Vol. 2950.

[8] Krisztian Balog and ChengXiang Zhai. 2023. User Simulation for Evaluating Information Access Systems. *arXiv* abs/2306.08550 (2023).

[9] Nicholas J. Belkin, Colleen Cool, Adelheit Stein, and Ulrich Thiel. 1995. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems With Applications* 9 (1995).

[10] Ben Carterette, Evangelos Kanoulas, and Emine Yilmaz. 2011. Simulating simple user behavior for system effectiveness evaluation. In *Proceedings of the 20th ACM International Conference on Information & Knowledge Management.*

[11] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.*

[12] Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you Hop: Conversational Question Answering over Knowledge Graphs Using Judicious Context Expansion. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019).

[13] W. Bruce Croft and R. H. Thompson. 1987. I3R: A new approach to the design of document retrieval systems. *J. Am. Soc. Inf. Sci.* 38 (1987).

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* abs/1810.04805 (2019).

[15] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval.*

[16] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv* abs/2301.07597 (2023).

[17] Zhiyuan Hu, Yue Feng, Anh Tuan Luu, Bryan Hooi, and Aldo Lipani. 2023. Unlocking the Potential of User Feedback: Leveraging Large Language Model as User Simulator to Enhance Dialogue System. *arXiv* abs/2306.09821 (2023).

[18] Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is ChatGPT better than Human Annotators? Potential and Limitations of ChatGPT in Explaining Implicit Hate Speech. *Companion Proceedings of the ACM Web Conference 2023* (2023).

[19] Hsin-Yuan Huang, Eunsol Choi, and Wen tau Yih. 2018. FlowQA: Grasping Flow in History for Conversational Machine Comprehension. *arXiv* abs/1810.06683 (2018).

[20] Mohit Iyyer, Wen tau Yih, and Ming-Wei Chang. 2017. Search-based Neural Structured Learning for Sequential Question Answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.*

[21] Alexander Kotov and ChengXiang Zhai. 2010. Towards natural question guided search. In *Proceedings of the ACM Web Conference 2010.*

[22] Nischal Ashok Kumar, Nigel Fernandez, Zichao Wang, and Andrew S. Lan. 2023. Improving Reading Comprehension Question Generation with Data Augmentation and Overgenerate-and-rank. *arXiv* abs/2306.08847 (2023).

[23] Taja Kuzman, Igor Mozeti, and Nikola Ljubesic. 2023. ChatGPT: Beginning of an End of Manual Linguistic Data Annotation? Use Case of Automatic Genre Identification. *arXiv* abs/2303.03953 (2023).

[24] Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016. Learning through Dialogue Interactions by Asking Questions. In *Proceedings of the 4th International Conference on Learning Representations.*

[25] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A Unified MRC Framework for Named Entity Recognition. *arXiv* abs/1910.11476 (2019).

[26] Javed Mostafa, Snehasis Mukhopadhyay, and Mathew Palakal. 2003. Simulation studies of different dimensions of users' interests and their impact on user modeling and information filtering. *Information Retrieval* 6 (2003).

[27] OpenAI. 2023. GPT-4 Technical Report. *arXiv* abs/2303.08774 (2023).

[28] Paul Owoicho, Ivan Sekulic, Mohammad Aliannejadi, Jeffrey Dalton, and Fabio Crestani. 2023. Exploiting Simulated User Feedback for Conversational Search: Ranking, Rewriting, and Beyond. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[29] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. *arXiv* abs/2304.03442 (2023).

[30] Joon Sung Park, Lindsay Popowski, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (2022).

[31] Alessandro Pegoraro, Kavita Kumari, Hossein Fereidooni, and Ahmad-Reza Sadeghi. 2023. To ChatGPT, or not to ChatGPT: That is the question! *arXiv* abs/2304.01487 (2023).

[32] Jin Qian, Bowei Zou, Mengxing Dong, Xiao Li, Aiti Aw, and Yu Hong. 2022. Capturing Conversational Interaction for Question Answering via Global History Reasoning. In *Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2022.*

[33] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? *arXiv* abs/2302.06476 (2023).

[34] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-Retrieval Conversational Question Answering. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).

[35] Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with History Answer Embedding for Conversational Question Answering. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019).

[36] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv* abs/1910.10683 (2019).

[37] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

[38] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* 7 (2018).

[39] Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and A. P. Sarath Chandar. 2018. Complex Sequential Question Answering: Towards Learning to Converse Over Linked Question Answer Pairs with a Knowledge Graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[40] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv* abs/1910.01108 (2019).

[41] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv* abs/2302.04761 (2023).

[42] Ivan Sekulic, Mohammad Aliannejadi, and Fabio Crestani. 2022. Evaluating Mixed-initiative Conversational Search Systems via User Simulation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*.

[43] Clemencia Siro, Mohammad Aliannejadi, and Maarten de Rijke. 2022. Understanding User Satisfaction with Task-oriented Dialogue Systems. In *Proceedings of the 2022 International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[44] Weiwei Sun, Shuo Zhang, Krisztian Balog, Zhaochun Ren, Pengjie Ren, Zhumin Chen, and Maarten de Rijke. 2021. Simulating user satisfaction for the evaluation of task-oriented dialogue systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[45] Yiming Tan, Dehai Min, Y. Li, Wenbo Li, Na Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of ChatGPT as a Question Answering System for Answering Complex Questions. *arXiv* abs/2303.07992 (2023).

[46] Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does Synthetic Data Generation of LLMs Help Clinical Text Mining? *arXiv* abs/2303.04360 (2023).

[47] Silvia Terragni, Modestas Filipavicius, Nghia Khau, Bruna Guedes, André Manso, and Roland Mathis. 2023. In-Context Learning User Simulators for Task-Oriented Dialog Systems. *arXiv* abs/2306.00774 (2023).

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv* abs/2302.13971 (2023).

[49] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 1st Workshop on Representation Learning for NLP*.

[50] Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. 2023. LLM-powered Data Augmentation for Enhanced Crosslingual Performance. *arXiv* abs/2305.14288 (2023).

[51] Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Sherry Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, Tran Bao Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic Questions and Where to Find Them: FairytaleQA – An Authentic Dataset for Narrative Comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

[52] Yi-Ting Yeh and Yun-Nung (Vivian) Chen. 2019. FlowDelta: Modeling Flow Information Gain in Reasoning for Conversational Machine Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

[53] Munazza Zaib, Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood, and Yang Zhang. 2021. Conversational question answering: a survey. *Proceedings of Knowledge and Information Systems* 64 (2021).

[54] Shuo Zhang and Krisztian Balog. 2020. Evaluating Conversational Recommender Systems via User Simulation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[55] Jing Zhao, Junwei Bao, Yifan Wang, Yongwei Zhou, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. RoR: Read-over-Read for Long Document Machine Reading Comprehension. *arXiv* abs/2109.04780 (2021).

[56] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A Survey of Large Language Models. *arXiv* abs/2303.18223 (2023).

[57] Yiming Zhu, Peixian Zhang, Ehsan ul Haq, Pan Hui, and Gareth Tyson. 2023. Can ChatGPT Reproduce Human-Generated Labels? A Study of Social Computing Tasks. *arXiv* abs/2304.10145 (2023).