Rewriting Conversational Utterances with Instructed Large Language Models

Elnara Galimzhanova
University of Pisa, Pisa, Italy
e.galimzhanova@studenti.unipi.it

Cristina Ioana Muntean, Franco Maria Nardini,
Raffaele Perego, Guido Rocchietti
ISTI-CNR, Pisa, Italy
{name.surname}@isti.cnr.it

***Abstract*—Many recent studies have shown the ability of large language models (LLMs) to achieve state-of-the-art performance on many NLP tasks, such as question answering, text summarization, coding, and translation. In some cases, the results provided by LLMs are on par with those of human experts. These models' most disruptive innovation is their ability to perform tasks via zero-shot or few-shot prompting. This capability has been successfully exploited to train *instructed* LLMs, where reinforcement learning with human feedback is used to guide the model to follow the user's requests directly. In this paper, we investigate the ability of instructed LLMs to improve conversational search effectiveness by rewriting user questions in a conversational setting. We study which prompts provide the most informative rewritten utterances that lead to the best retrieval performance. Reproducible experiments are conducted on publicly-available TREC CAST datasets. The results show that rewriting conversational utterances with instructed LLMs achieves significant improvements of up to 25.2% in MRR, 31.7% in Precision@1, 27% in NDCG@3, and 11.5% in Recall@500 over state-of-the-art techniques.**

***Index Terms*—conversational systems, query rewriting, LLMs, ChatGPT, information retrieval**

I. Introduction

Since their introduction, Large Language Models (LLMs) have impressed with their capabilities in dealing with tasks such as question answering, text summarization, coding, and translation, with performances that are comparable to those of human annotators. Thanks to their ability to perform tasks via few-shot learning, LLMs can learn from just a few examples, considerably expanding the range of applications supported and lowering the effort needed for targeting novel tasks. This feature has been successfully exploited to train *Instructed* LLMs, where methods from reinforcement learning with human feedback (RLHF) are used to directly instruct the model to act following the user's intention [1].

As a result, we assisted a new gold rush for part of the major tech companies to show their new intelligent systems. At first, we witness the introduction of ChatGPT, powered by a GPT-3.5 model. Then, we witness the release of a novel version of Bing search powered by GPT-4. The availability of the GPT-4-powered Bing search engine sets a definitive shift from a search paradigm based on "ten blue links" returned as an answer to a user query to a natural-language answer that is then returned to the user. Such an autonomous system automatically chooses the most relevant documents and extracts and elaborates the relevant information that is then presented to the user in the form of an answer to her/his query. This novel paradigm that exploits the dialogue to interact between the user and the search system can indeed provide a more friendly and natural way of interacting with the search service.

In this paper, we move a step forward in an orthogonal direction by studying the ability of instructed LLMs to improve the retrieval effectiveness of a state-of-the-art search engine in a conversational setting [2]–[4]. We aim to answer two main research questions:

- RQ1 Can an instructed LLM improve conversational search effectiveness by automatically rewriting the users' utterances to allow the search engine to retrieve more precise and relevant results?
- RQ2 Which prompting template performs best in order to generate rewritten queries that enhance retrieval performance?

We investigate the research questions above by adopting the Conversational Assistance Track (CAsT) framework provided by TREC for training and evaluating models in open-domain information-centric conversational dialogues [2].

The characteristics of conversational utterances, i.e., missing context from previous questions, topic shifts [5], [6], and implied concepts from previous answers, pose new challenges to deal with, which are a direct consequence of the paradigm shift introduced by conversational search. They heavily impact the performance of standard information retrieval techniques. Query rewriting techniques applied on a per-utterance level answer these challenges as they help propagate the context throughout the conversation and deal with possible topic shifts.

The novel contributions of this work are thus the following:

- We investigate utterance rewriting in conversational search using an instructed LLM and specifically designed prompting templates. Given an utterance and its context, we prompt the model asking to generate a rewriting of the utterance with the goal of enhancing the retrieval effectiveness of a state-of-the-art information retrieval system. This approach allows us to evaluate the ability of an instructed LLM to deal with the context of a conversation and possible topic shifts that may occur. At the same time, we inspect its ability to rewrite natural language utterances containing ambiguities, coreferences, omissions, acronyms, and colloquial grammar misuses.
- We present five different prompting templates to rewrite the utterances. Each prompt has been evaluated in an end-

to-end retrieval framework to assess its ability to improve the effectiveness of the conversational search system. All of the prompts have been tested in different conditions to establish the best way of prompting an LLM.
- We report the results of a comprehensive and reproducible experimental evaluation conducted using the publicly-available TREC CAsT datasets. Results show that rewriting utterances with the chosen instructed LLM achieves significant improvements of up to 31.7% in Precision@1, 25.2% in MRR, 27% in NDCG@3, and 11.5 % in Recall@500 over state-of-the-art rewriting techniques in conversational search.

The rest of the paper is organized as follows. Section II discusses related work, while Section III introduces our methodology. In Section IV we discuss the details of the prompting templates designed for query rewriting, the datasets, the baselines and competitors, and the two-stage retrieval architecture. The end-to-end performance of the proposed query rewriting pipeline is comprehensively assessed in Section V. Finally, Section VI presents the concluding remarks.

II. Related Work

a) Conversational search: Query rewriting is central in modern web search as it better models the user's information need and enhances retrieval effectiveness [7]. Similar challenges arise in conversational search, since utterances, like queries, may be ambiguous or poorly formulated.

Conversational utterance rewriting aims to reformulate a concise request in a conversational context to a fully specified, context-independent query dealing with anaphoras, ellipses, and other linguistic phenomena [5], [8]. These techniques aim at identifying terms previously mentioned in the conversation to expand the current utterance profitably [6], [9]–[11]. In this line, Aliannejadi *et al.* propose a novel neural utterance relevance model based on BERT that helps identify the utterances relevant to a given turn [9]. Voskarides *et al.* [10] model query rewriting for conversational search as a binary term classification task and introduce QuReTeC, a Bi-LSTM model that selects the valuable terms in context to enrich the query.

Other approaches rewrite the utterances by exploiting a fine-tuned neural model [12]–[15]. Yu *et al.* presents CQR, a few-shot generative approach to solve coreference and omissions in conversational query rewriting [12]. The authors propose two methods to solve coreference and omissions to generate weak supervision data that are then used to fine-tune GPT-2 to rewrite conversational queries. Results show that on the TREC CAsT Track a weakly-supervised finetuning of GPT-2 improves the ranking accuracy by 12%.

Vakulenko *et al.* [14] approach the problem by tackling conversational question answering. The authors propose a question-rewriting technique that translates ambiguous requests into semantically-equivalent unambiguous questions.

In more recent works, several papers exploit pre-trained language models to represent queries and documents in the same dense latent vector space and then use the inner product to compute the relevance score of a document to a given query. In conversational search, the representation of a query can be computed in two different ways. In one case, a stand-alone contextual query understanding module reformulates the user query into a rewritten query, exploiting the context history [16], and then a query embedding is computed, e.g. using embedding models such as ANCE [17] or STAR [18]. Alternatively, the learned representation function is trained to receive as input the query together with its context history and to generate a query embedding that is more similar to the manual query embeddings [19]. In both cases, dense retrieval methods are used to compute the query-document similarity by deploying efficient nearest neighbor techniques over specialized indexes, such as those provided by the FAISS toolkit [20].

b) Large Language Models: LLMs based on transformer architectures such as GPT are trained on large corpora of text data to comprehend and produce natural language [21], [22]. The pre-trained models produced with unsupervised training [23] can be easily fine-tuned for various tasks in a supervised setting. InstructGPT, based on GPT-3, has been fine-tuned using human feedback to make it better at following user intentions [1]. Bidirectional and Auto-Regressive Transformer (BART) integrate the strengths of two established models, i.e., BERT and GPT-2, and are trained using a denoising autoencoder approach to understand text structure and semantics, as well as generate fluent and coherent text [24]. Another instructed LLM model of the GPT family is ChatGPT[1], which is explicitly tailored for conversational applications [25].

Instructed LLMs such as ChatGPT are easily adaptable to new tasks and domains, making them very useful in various tasks. Wei *et al.* [26] propose ChatIE, a framework that employs ChatGPT to perform zero-shot Information Extraction (IE) tasks via multi-turn question-answering and claim that their method can achieve impressive results and surpass some full-shot models across three IE tasks. Sun et al. [27] found that ChatGPT can perform as well as, or better than, supervised methods in information retrieval relevance ranking when guided by domain-specific guidelines. The models mentioned earlier achieve impressive results in many NLP tasks, and their applications are many, from medicine to finance and beyond. With proper instructions, these models can solve a vast variety of tasks, making them valuable tools for researchers and developers alike. ChatGPT is the instructed LLM we use in our experiments.

Lately Mao et al. [28] conducted a work that studies the impact of LLMs. They focus on capturing the contextual conversational search intent through the use of GPT-3. The authors evaluate their findings in an ad-hoc dense retrieval scenario, using ANCE embeddings [17] for computing the similarity scores between documents and queries. We use their best-performing prompt in our experimental setting to see its effectiveness in our framework.

Our Contribution. This work contributes to the line of rewriting conversational utterances with generative models. Differently from previous works, we assess the capabilities of

[1]https://chat.openai.com/

TABLE I
NOTATION.

Symbol	Definition
$\mathcal{U}$	A multi-turn conversation composed of a sequence of utterances asked by a user to a conversational assistant.
Θ	An instructed LLM that we use for utterance rewriting, also referred to as *Assistant*.
u_i	The current original utterance at turn i in $\mathcal{U}$.
$\hat{u}_i$	The current utterance rewritten by Θ.
$u_1, \ldots, u_{i-1}$	The previous original utterances in $\mathcal{U}$.
$\hat{u}_1, \ldots, \hat{u}_{i-1}$	The previous utterances in $\mathcal{U}$ rewritten by Θ.
$\bar{u}_1, \ldots, \bar{u}_{i-1}$	The previous manually-rewritten utterances in $\mathcal{U}$.
$\hat{r}_1, \ldots, \hat{r}_{i-1}$	Responses to the previous utterances generated by Θ.
$\mathcal{C}$	The *Context* which is composed of the alternation between $u_1, \ldots, u_{i-1}$ and $\hat{u}_1, \ldots, \hat{u}_{i-1}$, or even adding $\hat{r}_1, \ldots, \hat{r}_{i-1}$. An example can be seen in Figure 1.
$\mathcal{E}$	The *Example* comprises original utterances $u_1, \ldots, u_{i-1}$ and their corresponding manually rewritten utterances $\bar{u}_1, \ldots, \bar{u}_{i-1}$.
s	The scope that explains our goal to the rewriting LLM Θ, also referred to as *System*.
p	The actual Prompt that, given u_i, specifies the instruction to Θ, namely, to rewrite the query.

an instructed LLM such as ChatGPT in rewriting utterances after few-shot training. We experiment with different prompts and instructions to offer the model different amounts and kinds of information for obtaining utterance rewritings that are competitive with—or better than—the state-of-the-art.

In this study, we evaluate ChatGPT's performance in explicit utterance rewriting. We conduct a comparative analysis with other state-of-the-art models employing explicit rewriting techniques [10], [12]. We acknowledge the potential contribution of dense retrieval approaches for utterance rewriting, as they can be applied after explicitly rewriting utterances. These approaches will be assessed in future research.

III. METHODOLOGY

Our goal is to understand to which extent a state-of-the-art instructed LLM can be used to improve conversational search effectiveness. To this respect, this work assesses with reproducible experiments the rewriting capabilities of ChatGPT (RQ1) and investigates the impact of different prompts and instructions on the effectiveness of a two-stage conversational search pipeline (RQ2). In Table I, we introduce the notation used to describe our task. Our rewriting system Θ, based on an instructed LLM, can take as input many of the elements described in the table in order to perform the rewriting of the current utterance u_i into a rewritten version $\hat{u}_i$. More formally, a typical rewriting request consists of the following:

$$\Theta(s, \mathcal{E}, \mathcal{C}, p, u_i) = \hat{u}_i, \tag{1}$$

where s represents the scope, i.e., the general task instructions of how we want the system to behave, $\mathcal{E}$ is a conversation example different from the current one, $\mathcal{C}$ is the context of u_i, and p is the prompt accompanying u_i, which explicitly instructs Θ detailing the rewriting request by adding specific desired characteristics, e.g., "concise", "verbose", and "self-explanatory".

A. Instructed LLM

We employ ChatGPT as the instantiation of Θ. Specifically, we employ the `gpt-3.5-turbo` model. As indicated in the ChatGPT API description[2], the model takes "a series of messages as input and returns a model-generated message as output". Since the model does not provide memory or session retention, in each interaction, we enclose the interaction history of previous turns of the conversation into the current request. This leads to having a conversational-style request, similar to an actual dialog.

We adapt our utterance rewriting requests to the input structure of the `gpt-3.5-turbo` model. The requests are composed of three main elements: *system*, *user*, and *assistant*. The "system" content is provided at the start of the session to specify the scope of the following interactions, in our case s. The "user" and "assistant", on the other hand, indicate the interactions between the user and ChatGPT, as a series of user instructions/requests consisting of prompt and current original utterance (p, u_i), and the corresponding assistant response containing the rewritten utterance $\hat{u}_i$.

To better understand what the best way of prompting the system is, we experiment with different ways of providing ChatGPT with the prompt and the context.

B. Prompting ChatGPT

We present five different prompts p to ask the instructed LLM to rewrite the utterances of a conversation $\mathcal{U}$.

The typical request submitted through the ChatGPT APIs[3] contains the elements detailed in Eq. 1, namely, scope, example, context, prompt, and current utterance. For all five prompts the example $\mathcal{E}$ consists of an exemplary conversation, chosen randomly from the dataset and not related with $\mathcal{U}$, where the user inputs are the original utterances, and the assistant inputs are instead the same utterances rewritten manually. Moreover, the context $\mathcal{C}$ consists of the previous utterances of $\mathcal{U}$, where the user inputs are the original utterances $u_1, \ldots, u_{i-1}$, and the assistant inputs are instead the same utterances rewritten by the model, $\hat{u}_1, \ldots, \hat{u}_{i-1}$. The only exception to this request template is the prompt P1, where the context consists of the previous utterances $u_1, \ldots, u_{i-1}$, and the assistant inputs consist of the previous utterances rewritten by the model $\hat{u}_1, \ldots, \hat{u}_{i-1}$ together with the generated answers $\hat{r}_1, \ldots, \hat{r}_{i-1}$.

In Figure 1, we report a visual example of a typical rewriting request. We can see how the first block represents the system, the second one is the example and the third is the context of the current conversation, while the last one contains the prompt and current question, followed by the answer (rewritten utterance) provided by the assistant.

The above structure of the requests allows us to assess not only the rewriting capabilities of the model but also its proficiency in retaining and exploiting the context information fed to the system. We now detail the five prompts with their specific characteristics and the intuition behind each of them.

[2]https://platform.openai.com/docs/guides/chat/instructing-chat-models
[3]https://platform.openai.com/docs/api-reference

TABLE II

EXAMPLE OF CONVERSATION TAKEN FROM THE CAST 2019 EVALUATION DATASET. THE "QID" COLUMN IDENTIFIES THE CONVERSATIONAL TURN, THE FIRST NUMBER REFERS TO THE CONVERSATION WHILE THE SECOND TO THE DIALOGICAL TURN. THE "RAW UTTERANCE" COLUMN REPORTS THE ORIGINAL UTTERANCE, THE "MANUALLY-REWRITTEN UTTERANCE" COLUMN REPORTS THE HUMAN ASSESSOR REWRITINGS, AND THE "AUTOMATICALLY-REWRITTEN UTTERANCE" COLUMN REPORTS AN EXAMPLE OF REWRITING GENERATED BY CHATGPT WITH OUR PROPOSED PROMPT P5.

qid	Raw utterance	Manually-rewritten utterance	Automatically-rewritten utterance
31_1	What is throat cancer?	What is throat cancer?	What is throat cancer?
31_2	Is it treatable?	Is throat cancer treatable?	Is throat cancer treatable?
31_3	Tell me about lung cancer.	Tell me about lung cancer.	Tell me about throat cancer.
31_4	What are its symptoms?	What are lung cancer's symptoms?	What are the symptoms of throat cancer?
31_5	Can it spread to the throat?	Can lung cancer spread to the throat?	Can throat cancer spread to other areas of the throat?
31_6	What causes throat cancer?	What causes throat cancer?	What causes throat cancer to spread to other areas of the throat?
31_7	What is the first sign of it?	What is the first sign of throat cancer?	What are the early signs of throat cancer?
31_8	Is it the same as esophageal cancer?	Is throat cancer the same as esophageal cancer?	Is throat cancer the same as esophageal cancer?
31_9	What's the difference in their symptoms?	What's the difference in throat cancer and esophageal cancer's symptoms?	What are the differences in the symptoms of esophageal cancer and throat cancer?

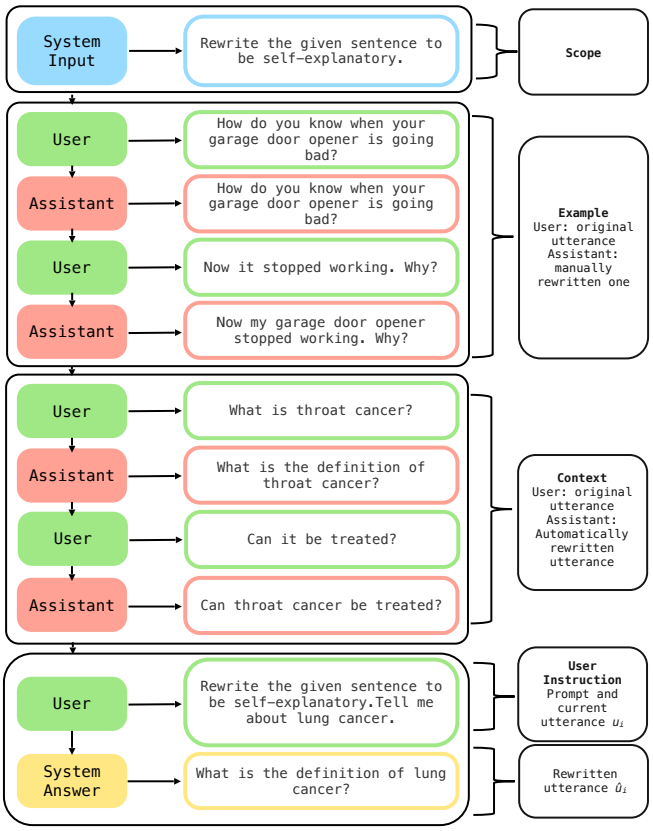

Fig. 1. Main elements of an utterance rewriting request. The *Scope* indicates the task that the model should perform. The *Example* is the artificial part of the interaction where the user part is the query to rewrite and the assistant part is the query rewritten by a human. The *Context* is composed of the previous queries rewritten by our model. The last section represents the current prompt and the output of the system.

P1 *Prompt:* "Rewrite the following question to be clear and complete and then provide an answer. Use the previous questions and answers to rewrite the question."
Rationale: P1 aims to instruct the model to generate a self-explanatory sentence using not only the information provided by the previous utterances but also by the generated answers.

P2 *Prompt:* "Rewrite the following question adding keywords for a retrieval system. Use the information from the previous questions. Return only the rewritten question."
Rationale: P2 aims to specify the final goal of the rewriting while keeping track of the context to see if the model is able to maximize the retrieval results.

P3 *Prompt:* "Rephrase the current question into a more concise and context-free form that is suitable for a multi-turn information search dialog using the context of the previous question. Do not add any extra sentences or notes."
Rationale: P3 aims to specify the final goal of the rewriting in the prompt and to instruct the model to generate a complete and concise rewriting of the given utterance.

P4 *Prompt:* "Reformulate the current question following the examples. [a list of 8 example pairs where each pair has the format "*Question:* raw question. *Rewritten:* manually rewritten question"]."
Rationale: P4 aims at reproducing the pattern given in the prompt to better rewrite the given utterances. Besides providing the example $\mathcal{E}$ within the request, we also repeat it in the prompt.

P5 *Prompt:* "In a multi-turn dialog system, rewrite the given sentence to be self-explanatory following the pattern of the previous interactions."
Rationale: P5 aims at reproducing the pattern given by the previous interactions between the user and the model, assuming that they are proficient in the rewriting task.

Moreover, we experiment in our setting also the best-performing prompt presented in the work of Mao *et al.* [28].

E "Reformulate the current question into a de-contextualized rewrite under the multi-turn information-seeking dialog context. Then generate a correct response. Print also the reformulated question."

We use the prompts above to generate rewritten utterances and test their effectiveness. We rewrite all the utterances of a conversation except the first one, u_1. In fact, several studies have shown that the first utterance of each conversation is already a self-explanatory sentence [6].

Before selecting the five prompts, we tested several other configurations not reported for the sake of brevity but resulting in worse performance. For example, we tried to use the prompt p only as system input and not in every user input u_i. We

also tested prompts not providing rewriting examples or using different textual instructions. As a general consideration, we notice that explaining the input to ChatGPT in a detailed way (e.g., by specifying *"In some cases, I will provide the questions previously made by the user. Use them to better reformulate the question."*) improves the performance and avoids some rewritings errors. Finally, we observe that the output of ChatGPT sometimes contains additional elements (e.g., clarifying questions) or it directly includes an answer to the utterance. For this reason, we post-process the output and keep only the actual rewritten utterance, $\hat{u}_i$.

IV. Experimental Setup

To assess the utterance rewriting quality, we submit $\hat{u}_i$ as a query to a two-stage information retrieval pipeline. We evaluate the effectiveness of the different rewriting strategies using the TREC CAsT framework [2]–[4], which allows us to perform an objective evaluation by comparing our results to those obtained by state-of-the-art competitors[4].

A. Conversational Datasets

Our experiments are based on the TREC Conversational Assistant Track (CAsT) 2019 and 2020[5] datasets. The CAsT 2019 [2] dataset consists of 20 human-assessed test conversations, while CAsT 2020 [3] includes 25 conversations, with an average of 10 turns per conversation. The CAsT 2019 and 2020 datasets include relevance judgments at the passage level. Conversations are provided with original and manually-rewritten utterances. The manually-rewritten utterances are the same conversational utterances as the original ones, where human assessors resolve missing keywords or references to previous topics. Relevance judgments have a three-point graded scale and refer to passages of the TREC CAR (TREC Complex Answer Retrieval), the MS-MARCO (MAchine Reading COmprehension) and the WaPo (TREC Washington Post Corpus) collections for CAsT 2019 and 2020 for a total of 38,636,520 passage. In these datasets, questions within a conversation are characterized by anaphora and ellipses. They imply a big part of the context and miss explicit references to the current topic. Table II reports some examples of utterances from the CAsT 2019 dataset. We can see that manually-rewritten utterances are concise and rephrase the original utterance by adding the missing tokens to make it self-explanatory. On the other hand, depending on the prompt, automatically-rewritten utterances tend to be more verbose although well-formed natural language questions.

B. Baselines

We assess the retrieval effectiveness of original, manually-rewritten, and automatically-rewritten utterances. In detail, we consider the following rewriting methods and baselines:

- *Original utterances*: raw utterances provided by TREC CAsT.
- *Manual utterances*: manually-rewritten utterances by human annotators provided by TREC CAsT.
- *QuReTeC* [10]: utterances are rewritten with a BiLSTM sequence to sequence model trained for query resolution.
- *CQR self-learn cv* [12]: utterances are generated in two steps, first with a GPT-2 model trained with self-supervised learning to generate contextual utterances containing few information presented in previous utterances. The second step is performed with a GPT-2 model fine-tuned on manual rewrites via five-fold cross-validation.
- *CQR rule-based cv* [12]: utterances are generated in two steps, first with a rule-base approach that deals with omissions and coreference and successively rewritten with a GPT-2 model fine-tuned on manual rewrites via five-fold cross-validation.
- *Prompt E* [28]: although the results by Mao *et al.* are achieved on a different generative model, i.e., GPT-3, we use their prompt in our experimental framework to compare its retrieval performance against ours.

C. Two-stage Retrieval

To evaluate and compare the different utterance rewritings, we index the TREC CAsT collections by removing stopwords and applying Porter's English stemmer. We use PyTerrier [29] to build the information retrieval pipeline, which is composed of two stages:

- The first stage performs document retrieval on the indexed collection with the DPH weighting model [30], using the raw, manually, and automatically-rewritten utterances;
- The second stage performs reranking of the top-1000 candidates retrieved by the first stage by using the MonoT5 model [31] made available in PyTerrier[6].

We measure the retrieval effectiveness of the first stage and of the second stage using the following metrics: Mean Reciprocal Rank (MRR), Precision@1 (P@1), Normalized Discounted Cumulative Gain@3 (NDCG@3), and Recall@500 (R@500). MRR and NDCG@3 are standard metrics used for evaluation purposes in the TREC CAsT framework while the others are included to provide a more comprehensive evaluation of the retrieval capabilities of the first-stage (R@500) and the reranking capabilities of the second-stage (P@1).

V. Results and Discussion

In this section, we discuss the experimental results on CAsT 2019 and 2020 datasets to assess the various rewriting strategies and compare them with the baselines.

A. First-stage Retrieval

In Table III, we report the results obtained when performing document retrieval using the DPH weighting model [30]. Results refer to the first-stage retrieval pipeline on both the CAsT 2019 and CAsT 2020 datasets. We also experiment with other weighting models, i.e., BM25 [32]. We do not report them as their results are worse than those achieved by DPH.

[4]We will release the code used for the experiments and the full set of rewritten utterances tested to favor the reproducibility of results.

[5]Conversational Assistant Track, https://www.treccast.ai/

[6]https://github.com/terrierteam/pyterrier_t5

TABLE III

FIRST-STAGE RETRIEVAL RESULTS IN TERMS OF MRR, P@1, NDCG@3 AND R@500 ON CAST 2019 AND CAST 2020 DATASETS. IN BOLD, WE REPORT THE BEST RESULTS ACHIEVED FOR EACH METRIC, EXCEPT MANUAL. WE MARK STATISTICALLY-SIGNIFICANT PERFORMANCE GAIN/LOSS, CALCULATED WITH THE TWO-PAIRED t-TEST (p-VALUE < 0.05) WITH BONFERRONI CORRECTION, OF OUR METHODS WITH RESPECT TO THE QURETEC AND CQR SELF-LEARN CV BASELINES WITH THE SYMBOLS ▲ AND ▼ FOR THE FIRST, △ AND ▽ FOR THE LATTER.

	CAsT 2019				CAsT 2020			
Prompt	**MRR**	**P@1**	**NDCG@3**	**R@500**	**MRR**	**P@1**	**NDCG@3**	**R@500**
Manual	0.6753△	0.5491	0.4002△	0.7374△▲	0.6220▲	0.5048▲	0.3277▲	0.6682▲
Original	0.3334▽▼	0.2254▽▼	0.1617▽▼	0.3815▽▼	0.2177▼	0.1587▼	0.0998▼	0.2532▼
P1	0.6327	**0.5260**	**0.3664**	0.6446	**0.5353▲**	**0.4231▲**	**0.2512**	**0.5710**
P2	0.5887	0.4624	0.2921	0.5775▽▼	0.4838	0.3750	0.2406	0.5488
P3	0.6129	0.5087	0.3363	0.6036▼	0.4580	0.3150	0.2153	0.5009
P4	0.6221	0.5116	0.3449	0.6311	0.4302	0.3317	0.2109	0.4963
P5	**0.6359**	0.5145	0.3331	0.6499	0.4775	0.3894	0.2266	0.5133
E	0.5837	0.4798	0.3094	0.5772▼	0.4520	0.3558	0.2181	0.5029
QuReTec [10]	0.6251	0.4913	0.3494	**0.6704**	0.4399	0.3221	0.2145	0.5163
CQR self-learn cv [12]	0.5915	0.4682	0.3336	0.6617	-	-	-	-
CQR rule-based cv [12]	0.5629	0.4162	0.3111	0.6569	-	-	-	-

The performance of our methods and baselines range between the ones obtained for the original and the manually-rewritten utterances. Considering CAsT 2019, P5 is the best-performing prompt when looking at MRR while P1 is the best-performing prompt in terms of Precision@1 and NDCG@3. Regarding R@500, the QuReTec baseline is the best-performing method. When performing the statistical significance evaluation using a two-paired t-test (p-value < 0.05) with the Bonferroni correction [33], the results achieved by our prompts are not statistically different from the state-of-the-art baselines, except for R@500 for P2, P3, and E.

Improved results are achieved when rewriting the utterances of the CAsT 2020 evaluation dataset. The best-performing rewriting method is based on P1, where all metrics show considerable gains over the QuReTec baseline. For P@1 and MRR, the improvement achieved by P1 is statistically significant when compared to the QuReTec baseline, with a 21.6% gain in MRR and 31.7% in P@1. NDCG@3 and R@500 increase by 17.1% and 10.6%, respectively. We remind the reader that P1 also considers the generated answers to the previously rewritten questions to produce the current rewriting. In fact, it is worth noting that, compared to CAsT 2019 where most relevant concepts could be found in the previous utterances, for CAsT 2020, some missing relevant concepts that fill out the context, can be found only in the responses and not in the utterance history. Results show that by generating the answers to the user requests and instructing the model to use them in the rewriting phase, we obtain improved results.

The fact that, independently of the dataset considered, our few-shot rewriting system obtains results as good as—or better than—state-of-the-art techniques should be further exploited in future work.

B. Second-stage Retrieval

In Table IV, we report the end-to-end results obtained with CAsT 2019 and 2020 when performing document re-ranking using the MonoT5 model in the second-stage retrieval pipeline.

Our intuition is that because our rewriting techniques produce verbose and well-formed utterance rewritings, it would be beneficial to use a LLM-based model such as T5, so as to effectively exploit the information added by the `gpt-3.5-turbo` model. We can see that the performance obtained by the generated rewritings achieves higher results than those obtained by the CQR and QuReTec competitors for prompts such as P1, P5 for CAsT 2019, and for all prompts for CAsT 2020.

The winning method for CAsT 2019 is P5, with an MRR of 0.8119 (3.3% increase), P@1 of 0.7283 (5.9% increase), NDCG@3 of 0.5343 that is slightly better than the one provided by QuRETec, i.e., 0.5330. Consistent with the first stage, also in the second-stage retrieval, the results are better with respect to the QuReTec baseline, except for R@500, although not statistically significant.

When considering the CAsT 2020 evaluation dataset, our rewriting methods show significant improvements after reranking. In this case, we have a clear winner, i.e., P1, for which all metrics improve over QuReTec in a statistically-significant way. The MRR increases by 25.2%, the P@1 by 31.7%, the NDCG@3 by 27.0%, and the R@500 by 11.5%. Also, for P2, we have a statistically-significant improvement of 22.17% in terms of NDCG@3.

Even in the second stage of retrieval, we obtain results as good as—or better than—state-of-the-art competitors, confirming that instructed LLMs are effective in rewriting utterances in a multi-turn conversational setting.

C. Answering our Research Questions

RQ1. We affirm that using an instructed LLM to rewrite utterances helps the effectiveness of the retrieval system. In fact, we can observe that for the CAsT 2020 dataset, we obtain significant improvements over the QuReTeC baseline, while for the CAsT 2019 we achieve the same results, and in some cases, we outperform QuReTeC and the two CQR competitors.

The results achieved also show that, although the LLM has not been fine-tuned explicitly for utterance rewriting, it

TABLE IV

SECOND-STAGE RETRIEVAL RESULTS IN TERMS OF MRR, P@1, NDCG@3 AND R@500 ON CAST 2019 AND CAST 2020 DATASETS. IN BOLD, WE REPORT THE BEST RESULTS ACHIEVED FOR EACH METRIC, EXCEPT MANUAL. WE MARK STATISTICALLY-SIGNIFICANT PERFORMANCE GAIN/LOSS, CALCULATED WITH THE PAIRED t-TEST (p-VALUE < 0.05) WITH BONFERRONI CORRECTION, OF OUR CORRESPONDING METHODS WITH RESPECT TO THE QURETEC AND CQR SELF-LEARN CV BASELINES WITH THE SYMBOLS ▲ AND ▼ FOR THE FIRST, △ AND ▽ FOR THE LATTER.

	CAsT 2019				CAsT 2020			
Prompt	**MRR**	**P@1**	**NDCG@3**	**R@500**	**MRR**	**P@1**	**NDCG@3**	**R@500**
Manual	0.8849△▲	0.8266△▲	0.6053△▲	0.7705△▲	0.8161▲	0.7308▲	0.5381▲	0.7361▲
Original	0.4643▽▼	0.3989▽▼	0.2791▽▼	0.4060▽▼	0.3301▼	0.2212▼	0.1813▼	0.2834▼
P1	0.7909	0.6936	0.5193	0.6974	**0.7249▲**	**0.6394▲**	**0.4386▲**	**0.6287▲**
P2	0.7440	0.6358	0.4829▼	0.6347▼	0.6758	0.5962	0.4220▲	0.6091
P3	0.7377	0.6647	0.4867	0.6419▼	0.6022	0.5144	0.3542	0.5597
P4	0.7575	0.6532	0.5155	0.6710	0.6086	0.5240	0.3601	0.5469
P5	**0.8119**	**0.7283**	**0.5343**	0.7059	0.6536	0.5721	0.4046	0.5650
E	0.6863	0.5954	0.4507	0.6157▼	0.6163	0.5481	0.3855	0.5572
QuReTec [10]	0.7858	0.6879	0.5330	**0.7111**	0.5788	0.4856	0.3454	0.5639
CQR self-learn cv [12]	0.7780	0.7052	0.5286	0.6938	-	-	-	-
CQR rule-based cv [12]	0.7630	0.6821	0.5109	0.6853	-	-	-	-

provides competitive results compared to the state of the art. This confirms the ability of these models to perform a variety of tasks via few-shot learning, thus lowering the effort needed for targeting novel tasks. In fact, custom-made models for utterance rewriting in conversational search, i.e., QuReTec, reach worse results on CAsT 2020 than an instructed LLM with well-designed prompts. We explain these results as a consequence of the capability of an LLM to deal with different datasets and domains, keeping a rewriting quality higher than other systems trained on limited data and thus characterized by a lower generalization power.

RQ2. For what concerns the best way of prompting the LLM, the best results are obtained with P1 for CAsT 2020, while with P1 and P5 for CAsT 2019. While for some of the prompts discussed we clearly explicit the scope of the rewriting (e.g. "[...]for a retrieval system[...]" in P2), in both P1 and P5 this information is not explicit, suggesting that this kind of instruction is not useful to obtain better rewritings.

Moreover, in both cases, there is a clear indication of how to exploit examples and context from the previous interactions. The difference is that P1 explicitly asks the model to also add previously generated answers to the context and use all the information for generating the rewriting $\hat{u}_i$. This proved particularly effective in the case of CAsT 2020. This could also be the reason why QuReTec underperforms as, by design, it only focuses on the previous utterance and does not integrate the content of the answers for generating the rewriting. Therefore, after establishing the best-performing prompts and observing that they both make use of the context, we can conclude that providing examples can have a significant impact on the model's capabilities in performing the chosen task.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed several methods for using an instructed LLM for the conversational utterance rewriting task.

We focused on assessing if such type of model is suitable for this task and if it is competitive with the current state-of-the-art rewriting techniques, which use models specifically fine-tuned for the task. We also studied different prompting techniques to assess the most effective ways to instruct the model using 5 prompt formulations.

We evaluate our proposals on the publicly-available TREC CAsT 2019 and CAsT 2020 datasets. We provide a comprehensive experimental evaluation of our proposed five ways of prompting the instructed LLM and state-of-the-art conversational rewriting baselines by assessing their retrieval effectiveness in a two-stage retrieval pipeline.

Experiments show that, in most cases, our proposed rewriting methods outperform the baselines. The largest gain is achieved for CAsT 2020 with increases in MRR by 25.2%, in P@1 by 31.7%, in NDCG@3 by 27.0%, and in R@500 by 11.5%. These results are obtained using prompt P1, in which the system is also required to consider previous answers when rewriting the current utterance. We can conclude that using an instructed LLM is beneficial for the utterance rewriting task in conversational search. These models can become a useful tool to further expand rewriting approaches and set new state-of-the-art standards.

Future Work. As future work, we are interested in studying how instructed LLMs can be used to generate synthetic data that can be exploited in other tasks of conversational search or even for enriching conversational datasets with weak supervision labels. The limited number of assessed conversations is in fact one of the main limitations in the conversational search domain. Moreover, we are interested in assessing the sensibility of prompting, i.e., how the utterance rewriting changes with respect to variations in the prompt and how it influences the retrieval performance, in a systematic and comprehensive way.

Acknowledgements. Funding for this research has been provided by: PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI" funded by the European Union (EU) under the NextGeneration EU programme; the EU's Horizon Europe research and innovation programme EFRA

(Grant Agreement Number 101093026). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the EU or European Commission-EU. Neither the EU nor the granting authority can be held responsible for them.

REFERENCES

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022.

[2] J. Dalton, C. Xiong, V. Kumar, and J. Callan, "CAsT-19: A dataset for conversational information seeking," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Jul. 2020. [Online]. Available: https://doi.org/10.1145/3397271.3401206

[3] J. Dalton, C. Xiong, and J. Callan, "CAsT 2020: The conversational assistance track overview." TREC'20, Virtual, 2020. [Online]. Available: https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.C.pdf

[4] ——, "TREC CAsT 2021: The conversational assistance track overview." TREC'21, Virtual, 2021. [Online]. Available: https://trec.nist.gov/pubs/trec30/papers/Overview-CAsT.pdf

[5] I. Mele, C. I. Muntean, F. M. Nardini, R. Perego, N. Tonellotto, and O. Frieder, "Topic propagation in conversational search," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, J. X. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, and Y. Liu, Eds. ACM, 2020, pp. 2057–2060. [Online]. Available: https://doi.org/10.1145/3397271.3401268

[6] ——, "Adaptive utterance rewriting for conversational search," *Inf. Process. Manag.*, vol. 58, no. 6, p. 102682, 2021. [Online]. Available: https://doi.org/10.1016/j.ipm.2021.102682

[7] Y. He, J. Tang, H. Ouyang, C. Kang, D. Yin, and Y. Chang, "Learning to rewrite queries," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1443–1452.

[8] L. Yang, H. Zamani, Y. Zhang, J. Guo, and W. B. Croft, "Neural matching models for question retrieval and next question prediction in conversation," ArXiv Preprint 1707.05409, 2017.

[9] M. Aliannejadi, M. Chakraborty, E. A. Ríssola, and F. Crestani, "Harnessing evolution of multi-turn conversations for effective answer retrieval," in *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, ser. CHIIR '20. Association for Computing Machinery, 2020, pp. 33–42. [Online]. Available: https://doi.org/10.1145/3343413.3377968

[10] N. Voskarides, D. Li, P. Ren, E. Kanoulas, and M. de Rijke, "Query resolution for conversational search with limited supervision," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 921–930. [Online]. Available: https://doi.org/10.1145/3397271.3401130

[11] G. Rocchietti, O. Frieder, C. I. Muntean, F. M. Nardini, and R. Perego, "Commonsense injection in conversational systems: An adaptable framework for query expansion." in *IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology*, 2023.

[12] S. Yu, J. Liu, J. Yang, C. Xiong, P. Bennett, J. Gao, and Z. Liu, "Few-shot generative conversational query rewriting," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1933–1936. [Online]. Available: https://doi.org/10.1145/3397271.3401323

[13] J. Hao, Y. Liu, X. Fan, S. Gupta, S. Soltan, R. CHADA, P. Natarajan, E. Guo, and G. Tur, "Cgf: Constrained generation framework for query rewriting in conversational ai," in *EMNLP 2022*, 2022.

[14] S. Vakulenko, S. Longpre, Z. Tu, and R. Anantha, "Question rewriting for conversational question answering," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM, 2021, pp. 355–363. [Online]. Available: https://dl.acm.org/doi/10.1145/3437963.3441748

[15] H. Su, X. Shen, R. Zhang, F. Sun, P. Hu, C. Niu, and J. Zhou, "Improving multi-turn dialogue modelling with utterance ReWriter," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 22–31. [Online]. Available: https://aclanthology.org/P19-1003

[16] J. Gao, C. Xiong, P. Bennett, and N. Craswell, "Neural approaches to conversational information retrieval," *CoRR*, vol. abs/2201.05176, 2022. [Online]. Available: https://arxiv.org/abs/2201.05176

[17] L. Xiong, C. Xiong, Y. Li, K. Tang, J. Liu, P. N. Bennett, J. Ahmed, and A. Overwijk, "Approximate nearest neighbor negative contrastive learning for dense text retrieval," *CoRR*, vol. abs/2007.00808, 2020. [Online]. Available: https://arxiv.org/abs/2007.00808

[18] J. Zhan, J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma, "Optimizing dense retrieval model training with hard negatives," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1503–1512. [Online]. Available: https://doi.org/10.1145/3404835.3462880

[19] S. Yu, Z. Liu, C. Xiong, T. Feng, and Z. Liu, "Few-shot conversational dense retrieval," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 829–838. [Online]. Available: https://doi.org/10.1145/3404835.3462856

[20] J. Johnson, M. Douze, and H. Jegou, "Billion-scale similarity search with gpus," *IEEE Trans. Big Data*, vol. 7, no. 03, pp. 535–547, 2021.

[21] R. Alec, N. Karthik, S. Tim, and S. Ilya, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv*, 2017.

[23] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[24] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.

[25] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, and B. Ge, "Summary of chatgpt/gpt-4 research and perspective towards the future of large language models," 2023.

[26] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang, Y. Jiang, and W. Han, "Zero-shot information extraction via chatting with chatgpt," 2023.

[27] W. Sun, L. Yan, X. Ma, P. Ren, D. Yin, and Z. Ren, "Is chatgpt good at search? investigating large language models as re-ranking agent," 2023.

[28] K. Mao, Z. Dou, H. Chen, F. Mo, and H. Qian, "Large language models know your contextual search intent: A prompting framework for conversational search," 2023.

[29] C. Macdonald, N. Tonellotto, S. MacAvaney, and I. Ounis, "PyTerrier: Declarative experimentation in python from BM25 to dense retrieval," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM '21. Association for Computing Machinery, 2021, pp. 4526–4533. [Online]. Available: https://doi.org/10.1145/3459637.3482013

[30] G. Amati and C. J. Van Rijsbergen, "Probabilistic models of information retrieval based on measuring the divergence from randomness," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, p. 357–389, oct 2002. [Online]. Available: https://doi.org/10.1145/582415.582416

[31] R. Pradeep, R. Nogueira, and J. J. Lin, "The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models," *ArXiv*, vol. abs/2101.05667, 2021.

[32] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, p. 333–389, apr 2009. [Online]. Available: https://doi.org/10.1561/1500000019

[33] P. Sedgwick, "Multiple significance tests: the bonferroni correction," *BMJ (online)*, vol. 344, pp. e509–e509, 01 2012.