Query Reformulation using Query History for Passage Retrieval in Conversational Search

Sheng-Chieh Lin[*]
jacklin_64@citi.sinica.edu.tw
Research Center for Information Technology Innovation, Academia Sinica

Jheng-Hong Yang[*]
jhyang@citi.sinica.edu.tw
Research Center for Information Technology Innovation, Academia Sinica

Rodrigo Nogueira
rodrigonogueira4@gmail.com
David R. Cheriton School of Computer Science, University of Waterloo

Ming-Feng Tsai
mftsai@nccu.edu.tw
Department of Computer Science, National Chengchi University

Chuan-Ju Wang
cjwang@citi.sinica.edu.tw
Research Center for Information Technology Innovation, Academia Sinica

Jimmy Lin
jimmylin@uwaterloo.ca
David R. Cheriton School of Computer Science, University of Waterloo

ABSTRACT

Passage retrieval in a conversational context is essential for many downstream applications; it is however extremely challenging due to limited data resources. To address this problem, we present an effective multi-stage pipeline for passage ranking in conversational search that integrates a widely-used IR system with a conversational query reformulation module. Along these lines, we propose two simple yet effective query reformulation approaches: historical query expansion (HQE) and neural transfer reformulation (NTR). Whereas HQE applies query expansion, a traditional IR query reformulation technique, NTR transfers human knowledge of conversational query understanding to a neural query reformulation model. The proposed HQE method was the top-performing submission of automatic systems in CAsT Track at TREC 2019. Building on this, our NTR approach improves an additional 18% over that best entry in terms of NDCG@3. We further analyze the distinct behaviors of the two approaches, and show that fusing their output reduces the performance gap (measured in NDCG@3) between the manually-rewritten and automatically-generated queries to 4 from 22 points when compared with the best CAsT submission.

1 INTRODUCTION

In recent years, the rise of machine learning techniques has accelerated the development of conversational agents such as smart speakers and digital personal assistants [38]. Therefore, conversational information seeking is both a timely and an important research area in which we seek to boost the ability of conversational assistant systems to satisfy users with information needs [15].

Understanding users' conversations is a challenging part of a generic conversational assistant system. The information needs of users in such a scenario—conversational question answering (ConvQA)—are typically colloquially expressed and contextually dependent. To make such a challenging task tractable, environmental settings are generally controlled to answer questions within a *relevant* document, under which several studies [24, 37] have conducted conversational context modeling leading to progress in ConvQA benchmarks such as CoQA and QuAC [10, 41].

Table 1: CAsT Training Topic 1. A conversation consists of several questions. Each question generated by a user continues its previous utterances. The task is to find the relevant passages for each question based on its previous utterances.

Title: career choice for Nursing and Physician's Assistant

	Conversation Utterances
1	What is a physician's assistant?
2	What are the educational requirements required to become one?
3	What does it cost?
4	What's the average starting salary in the UK?
5	What about in the US?
6	What school subjects are needed to become a registered nurse?
7	What is the PA average salary vs an RN?
8	What the difference between a PA and a nurse practitioner?
9	Do NPs or PAs make more?
10	Is a PA above a NP?
11	What is the fastest way to become a NP?
12	How much longer does it take to become a doctor after being an NP?

Another underlying scenario is open-domain ConvQA, in which answers are sought for given open-domain questions from one or more knowledge bases. This scenario makes the problem much more complex than those considered in previous ConvQA studies and significantly deteriorates the performance of QA systems [16]. In particular, for such a scenario, information retrieval (IR) in conversational search is naturally involved [8, 38]. As a result, to facilitate generic open-domain ConvQA systems, conversational passage retrieval (ConvPR) plays a vital role in the whole systems, for which, however, little has been done in the literature.

There are currently two main challenges facing ConvPR: limited labeled data and ambiguous queries. First, even though neural networks have brought fruitful progress in natural language processing (NLP) [17, 29, 57] and IR [3, 33], ConvPR remains challenging due to the limited amount of labeled data. To our best knowledge, at the current time, there is no reasonably-sized training dataset for ConvPR in contrast to other ad-hoc passage retrieval tasks, e.g., MS MARCO, TREC CAR [7, 19]. Specifically, the conversational assistant track (CAsT) of the text retrieval conference (TREC) 2019 [25] only provides a total of 108 conversational user utterances in 13 topics with relevance judgments for model training, whereas MS MARCO and TREC CAR training sets contain 530k and 3M queries with relevant passages, respectively.

[*] Contributed equally.

Second, ConvPR queries are usually ambiguous, due to commonly faced coreference and omission problems; therefore, it also requires tracking and understanding the information needs behind conversational user utterances, as users may ask questions referring to their past dialogues [38]. Table 1 shows an example of conversational user utterances in the CAsT training set. Observe that the second utterance contains *one*, denoting *physician's assistant*, showing the importance of coreference resolution in ConvPR. Also, the fifth utterance omits the contexts (*average starting salary of physician's assistant*) from previous dialogues, demonstrating the necessity to account for omissions in ConvPR. Clearly, without appropriate processing, raw user utterances are ambiguous queries for traditional IR systems, since it is hard to interpret them without context. The resultant need for tracking and understanding further increases the complexity beyond ad-hoc IR problems.

To address these two challenges, we propose a conversational multi-stage retrieval system with a conversational query reformulation (CQR) module. We build on competitive baselines in an existing IR toolkit for ad-hoc retrieval and take advantage of existing work on BERT-based re-ranking. To be clear, our primary focus is on conversational tracking and understanding. Inspired by research on query expansion and conversational query understanding, we propose two simple yet effective CQR approaches to address this problem, both of which inject context information into ambiguous user utterances for downstream IR systems.

Specifically, the first approach—historical query expansion (HQE)—is a non-parametric model that applies query expansion techniques using context information. Neural transfer reformulation (NTR), the other approach, transfers knowledge of conversational query understanding by training a neural model to mimic how humans rewrite questions in a conversational context.

The contributions of this work are summarized as follows:

- We demonstrate the effectiveness of two conversational query reformulation approaches (HQE and NTR) stacked on top of a widely-used multi-stage search architecture.
- We conduct a detailed analyses of the HQE and NTR approaches quantitatively and qualitatively, explaining their pros and cons. One variant of HQE was the best automatic submission to TREC 2019 CAsT, and NTR further improves it by 18% in NDCG@3. Since our work only exploits CAsT training data for hyperparameter tuning, it provides strong baselines for future studies.

In sum, this work demonstrates how to tackle ConvPR with limited training data, based on which we build simple but effective baselines for future IR research in conversational search.

2 RELATED WORK

Open-domain question answering (QA) systems return answers in response to user questions, both posed in natural language, from a broad range of domains [47, 53]. An automatic open-domain QA system is often constructed with a pipeline: an IR model followed by a reading comprehension (RC) model to infer the answer from retrieved documents [8].

Despite the progress of QA system on RC models [17, 29, 45, 57], few studies address retrieval [8, 16, 53]. Most QA research [26, 40, 50, 56, 59], including conversational QA studies [10, 41], focuses on a restricted version of the open-domain QA problem posed in [8, 18, 20]: returning answers from a finite set of relevant documents—a relevant article [40, 56] or multi-hop hyperlinked documents [58]. Our work instead addresses a research problem regarding retrieval, especially in the context of open-domain conversational questions, posed in natural language, in a sequence.

Multi-stage retrieval systems are comprised of a candidate generation process followed by one or more re-ranking stages to strike a balance between efficiency and effectiveness [6, 11, 49]. Multi-stage retrieval systems research includes feature extraction efficiency [5], dynamic cutoff depth [14], shard prediction [31], and joint cascade ranking optimization [3, 22, 33, 35, 51]. The foundation of our work is built on a competitive cascade pipeline proposed by [33] and [3]: BM25 candidate generation with BERT re-ranking, the effectiveness of which has been proved in representative IR datasets: Robust04, TREC CAR, and MS MARCO [3, 7, 19].

Query reformulation (QR) has proven effective in IR. For example, Xu and Croft [54] expand a query with terms from retrieved documents; Nogueira and Cho [32] further improve IR systems using reinforcement learning to reformulate the query. Note that although many previous studies focus on improving the performance of ad-hoc queries, we emphasize QR in a conversational context. Among QR studies, the most relevant works to ours are [21, 42], both of which demonstrate the feasibility of deep learning for reformulating conversational queries. However, they only examine one facet of performance in terms of question-in-context rewriting. In this work, though, we practically apply and formally analyze a query-expansion-based method as well as a transfer learning method [23, 36] under a full conversational IR pipeline.

Conversational search [38] covers a broad range of perspectives to facilitate an IR task in a conversational context: natural language interaction, cumulative clarification [4], feedback collection, and information needs profiling during conversations. In the literature, our work is closely related to that based on web search [1, 2]; even so, our study differs from these in the following three ways. First, in our task, the user's information needs are expressed both colloquially and sequentially; thus, utterances include common natural language features beyond keyword queries, e.g., coreference, omission, and sentence semantics. Second, previous web search works involve in-domain model training, whereas this work represents a simple solution—only hyperparameter tuning. Finally, web search studies rely on user responses (e.g., clicks) as positive feedback, which can be viewed as implicit relevance without guidelines for consensus judgements used in our task.[1]

3 METHODOLOGY

3.1 Problem Setup

Conversational passage retrieval (ConvPR) is defined as an IR task in a conversational context. Given a sequence of conversational utterances $u^s = (u_1, \cdots, u_i, u_{i+1}, \cdots)$ for a topic-oriented session $s \in S$, where S is the set of all dialogue sessions and u_i stands for the i-th utterance ($i \in \mathbb{N}^+$) in the session, which is formalized through turn i, the goal of this task is to find a set of relevant passages $\mathcal{P}_i$, for each turn's user utterance u_i that satisfies the information needs in turn i with the context in previous turns $u_{<i} = (u_1, \cdots, u_{i-1})$.

[1] https://static.googleusercontent.com/media/guidelines.raterhub.com/en/ /searchqualityevaluatorguidelines.pdf

Task scope To facilitate the ConvPR task and to provide a reusable, tractable dataset, the organizers of CAsT of TREC 2019 began with a selection of open-domain exploratory information needs I and provided a predefined set of topic-oriented sessions S^I.[2] In addition, a passage collection $C \supseteq \mathcal{P}_i$ was provided to retrieve candidate response passages for each turn in these sessions.

Under the CAsT setting, the utterances in the provided topic-oriented sessions S^I not only control the complexity of the task but also mimic features of "real" dialogues via the following properties:

- Utterance transitions are coherent between turns in a given topic-oriented session.
- Utterances are natural language questions, which are similar to the questions in the widely used Google Natural Questions dataset [27].
- Coreference and omission of natural language features in dialogues are included.
- Turns depend only on previous utterances and not system responses.
- Comparison between subtopics are introduced.

Conversational multi-stage retrieval system To reuse existing IR pipelines and benefit from the fine-tuned performance of relevance prediction models, a typical approach for ConvPR is to reformulate user utterances with their context into suitable queries and feed the reformulated queries into the pipelines.

For an IR system, let $P(R = 1 \mid q, p)$ denote the probability conditioned on a query-passage pair (q, p), where $R = 1$ denotes that passage $p \in C$ is relevant to query q (otherwise, $R = 0$). Currently, a mainstream method to facilitate IR is to further factorize $P(R = 1 \mid q, p)$ into a multi-stage pipeline $f_\theta \circ f_\phi \propto P(R = 1 \mid q, p)$ as a trade-off between efficiency and effectiveness: f_ϕ is a predefined non-parametric model such as Okapi BM25, the vector space model with TF-IDF, or variants of traditional IR models, and f_θ stands for data-driven parametric models such as neural networks or other machine learning methods.

Likewise, for ConvPR, we factorize the probability of retrieving a relevant passage $p \in \mathcal{P}_i$ for each turn i with an information set $\{u_i, u_{<i}\}$ that comprises the utterances by turn i as

$$P(R = 1 \mid \{u_i, u_{<i}\}, p) = P(R = 1 \mid q_i, p)\, P(q_i \mid \{u_i, u_{<i}\})\,. \quad (1)$$

With this formulation, ConvPR can be approximated by separately maximizing the probabilities of (a) a relevance prediction model $P(R = 1 \mid q_i, p)$ and (b) a query reformulation model $P(q_i \mid \{u_i, u_{<i}\})$. Thus the goal of a query reformulation model is to reformulate a raw conversational user utterance u_i in each turn i into a clear and informative query q_i for the relevance prediction model [61].

As the judgments of relevant pairs in the training set from CAsT are sparse and very limited (see Table 2), we here focus on query reformulation methods and leave the burden of tuning a relevance prediction model to a known competitive pipeline—BM25 with BERT—in the large-scale passage ranking task [7, 19, 33].

Conversational query reformulation The goal of conversational query reformulation (CQR) is to obtain an informative query q_i for each turn i for downstream relevance prediction models. Specifically, given an information set $\{u_i, u_{<i}\}$ that includes the utterances by turn i, the tasks of CQR consist of the following two components: (a) filter out unnecessary information in $\{u_i, u_{<i}\}$ and (b) construct informative input q_i from the filtered information. Thus with CQR we seek a function $q_i = g(\{u_i, u_{<i}\})$, the output of which (i.e., q_i) maximizes the probability in Eq. (1).

However, given the limited number of relevance labels, using supervised learning to construct a parametric function to maximize Eq. (1) is difficult. Therefore, we propose two label-free approximations for CQR as intuitive attempts. The first one is a non-parametric predefined model $g_\phi(\cdot)$ (see Section 3.3); the other is an off-the-shelf data-driven parametric model $g_\theta(\cdot)$ pretrained on other datasets under a transfer learning paradigm (Section 3.4). Note that both approaches only approximate $g(\cdot)$ with $g_\phi(\cdot)$ and $g_\theta(\cdot)$, respectively, due to the fact that the objective of $g(\cdot)$ in fact involves optimizing queries for an IR system.

3.2 Observations

In order to develop models for CQR with limited training data, we start with observing the characteristics of conversational user utterances.

Observation #1: Main topic and subtopic A session is centered around a main topic and the turns in the session dive deeper into several subtopics, each of which however only lasts a few turns. For instance, in Table 1, the main topic of the session is "physician's assistant" according to which Turns 2 and 3 discuss the subtopic of "educational requirements" while Turns 4 and 5 are related to the subtopic of "average starting salary."

Observation #2: Degree of ambiguity The degree of ambiguity divides utterances into three categories. The first category includes utterances with clear implications, which can thus be treated as ad-hoc queries, such as Turns 1 and 6 in Table 1. The second category contains those starting a subtopic (e.g., Turns 2 and 4), and the last category is composed of most ambiguous utterances that continue a subtopic (e.g., Turns 3 and 5).

Based on the above observations, we propose two CQR methods: (1) Historical Query Expansion (HQE), a heuristic query expansion strategy; (2) Neural Transfer Reformulation (NTR), a data-driven approach transferring human knowledge to neural models from human annotated queries.

3.3 Historical Query Expansion

We first introduce HQE to heuristically capture the observations. Specifically, there are three main steps in HQE. For each utterance in a session, we (1) extract the main topic and subtopic keywords from the utterance; (2) measure the ambiguity of the utterance; (3) perform query expansion for the ambiguous utterances with the main topic and subtopic keywords extracted from previous turns. We propose keyword extractor and query performance predictor modules to realize these three steps for constructing the non-parametric function g_ϕ.

3.3.1 Keyword extractor (KE). Given an utterance u_i consisting of $n(u_i)$ tokens, the utterance is represented as a tuple $\left(t_i^1, \ldots, t_i^{n(u_i)}\right)$, where t_i^k denotes the k-th token in u_i. The aim of the KE is to compute the score of each token in the utterance so that the score indicates the importance of the token in the utterance. For each token, we propose leveraging the retrieval score of its most relevant

[2] http://www.treccast.ai/

document to characterize its importance in the utterance as

$$\mathcal{R}_i^k = \text{KE}\left(t_i^k, C\right) = \max_{p \in C} \left\{\mathcal{F}_{\text{KE}}\left(t_i^k, p\right)\right\}, \tag{2}$$

where $\mathcal{R}_i^k$ denotes the importance score of token t_i^k, and $\mathcal{F}_{\text{KE}}(\cdot)$ is the function to compute the relevance between a token and a passage p. The intuition behind this design is that the importance of a token can be judged from those documents that are highly relevant to it; that is, if a word is representative of its relevant documents, it is with high probability a keyword.

3.3.2 *Query performance predictor (QPP).* Given an utterance u_i and a passage collection C, QPP measures the utterance's ambiguity. The literature demonstrates that the degree of query ambiguity is closely related to its ambiguity with respect to the collection of documents being searched [13, 43, 60]; thus, many metrics evaluate query ambiguity by analyzing retrieval scores. As we here are focused on providing an effective query expansion strategy for CQR rather than calculating the most accurate QPP, we keep the measurement for utterance ambiguity as simple as possible. Following the KE, we measure utterance ambiguity for u_i as

$$\mathcal{A}_i = \text{QPP}\left(u_i, C\right) = \max_{p \in C} \left\{\mathcal{F}_{\text{QPP}}\left(u_i, p\right)\right\}, \tag{3}$$

where $\mathcal{A}_i$ stands for the degree of utterance ambiguity and $\mathcal{F}_{\text{QPP}(\cdot)}$ estimates the relevance score between a passage and an utterance. In our experiments, we set $\mathcal{F}_{\text{QPP}(\cdot)}$ ($\mathcal{F}_{\text{KE}}(\cdot)$) as BM25 function. Note that the higher the $\mathcal{A}_i$ score, the clearer the utterance u_i.

3.3.3 *Putting it all together.* Algorithm 1 details the procedure of the proposed HQE, $g_\phi(\{u_i, u_{<i}\}, C)$: keyword extraction (lines 3–8), query performance prediction (line 10), and query expansion (lines 11–13). Note that $\mathcal{R}_{\text{topic}}$, $\mathcal{R}_{\text{sub}}$ (where $\mathcal{R}_{\text{topic}} > \mathcal{R}_{\text{sub}}$), η, and $\mathcal{M}$ are hyperparameters. Specifically, for each utterance u_i in a session $s \in S$ and a given passage collection C, HQE first extracts topic and subtopic keywords from u_i and collects them in the keyword sets W_{topic} and W_{sub}, respectively. Then, QPP measures the clearness (ambiguity) of all u_i for $i > 1$. Here η is the threshold to judge whether an utterance falls into the most ambiguous category. For all u_i except the first utterance u_1, HQE first rewrites u_i by concatenating u_i with the topic keyword sets W_{topic} collected from u_i and $u_{<i}$. Moreover, if u_i is ambiguous (i.e., $\mathcal{A}_i < \eta$), HQE further adds the subtopic keywords from previous $\mathcal{M}$ turns and turn i. We thus assume that the first utterance in a session is clear enough and that following utterances belong to the second or the most ambiguous category. Note that we concatenate W_{sub} derived from previous $\mathcal{M}$ turns, as subtopic keywords last a few turns (see Observation #1). Also note that W_{sub} includes the topic keywords in W_{topic}, which ensures that topic keywords gain higher term weights than subtopic keywords in rewritten utterances.

3.4 Neural Transfer Reformulation

Following a thought of a series of works regarding data-driven conversational query reformulation using neural networks [21, 28, 42, 52], we also propose reformulating a raw utterance u_i into a coreference-and-omission-free natural language question q_i^{NL} using neural transfer reformulation (NTR), which leverages neural networks to mimic and transfer patterns of how people rewrite questions in a conversational context.

Algorithm 1: Historical Query Expansion

```
   Input: $u_i$, $u_{<i}$, $C$
   Output: $\bar{u}_i$
1  $\bar{u}_i \leftarrow ()$; $W_{\text{topic}} \leftarrow \{\}$; $W_{\text{sub}} \leftarrow \{\}$
2  for $j = 1$ to $i$ do
3      for $k = 1$ to $n(u_j)$ do
4          $\mathcal{R}_j^k = \text{KE}\left(t_j^k, C\right)$
5          if $\mathcal{R}_j^k > \mathcal{R}_{\text{topic}}$ then
6              $W_{\text{topic}}.\text{insert}\left(t_j^k\right)$
7          if $(\mathcal{R}_j^k > \mathcal{R}_{\text{sub}})$ and $(j \geq i - \mathcal{M})$ then
8              $W_{\text{sub}}.\text{insert}\left(t_j^k\right)$
9  if $i > 1$ then
10     $\mathcal{A}_i = \text{QPP}(u_i, C)$
11     $\bar{u}_i.\text{insert}(t)$ for all $t \in W_{\text{topic}}$
12     if $\mathcal{A}_i < \eta$ then
13         $\bar{u}_i.\text{insert}(t)$ for all $t \in W_{\text{sub}}$
14 $\bar{u}_i.\text{append}(u_i)$
15 return $\bar{u}_i$
```

We need these ingredients to use NTR to construct the parametric function g_θ: (a) a large-scale, high-quality dataset of human generated q^{NL} with source utterances and contexts; (b) an architecture to map an utterance and its conversational context into q^{NL}; (c) a dataset with enough diversity to cover open-domain exploratory information needs selected in the session sets of our interest S^I.

Fortunately, open-domain QA research has produced QuAC [10]—a diverse, large-scale dataset that contains conversational natural language questions of exploratory information needs—as well as CANARD [21], a derived conversational question-in-context rewriting dataset with human generated questions for QuAC questions.

Like CANARD and text summarization studies [21, 46], we choose a sequence-to-sequence [9, 48] (Seq2Seq) architecture to map variable length conversational contexts $u_{<i}$ and u_i into q_i^{NL}. Without loss of generality, instead of using θ to represent the parametric function g_θ that reformulates conversational queries optimized for an IR system in Eq. (1), we define a function parameterized by $\bar{\theta}$ taking input tokens $(x_1, \ldots, x_n)$ of length n and output tokens $(y_1, \ldots, y_m)$ of length m as

$$\text{Seq2Seq}\left((x_1, \ldots, x_n), \bar{\theta}\right) = (y_1, \ldots, y_m), \tag{4}$$

for this neural historical query reformulation task. A proxy for obtaining a particular set of parameters $\hat{\theta}$ of this task under a configuration of a parametric function Seq2Seq$(\cdot, \hat{\theta})$ and a dataset from CANARD instead of CAsT is then

$$\hat{\theta} = \arg\max_{\bar{\theta}} \prod_i P_i\left(q_i^{\text{NL}} \mid \text{Seq2Seq}([\bar{u}_{<i} \parallel \bar{u}_i], \bar{\theta})\right), \tag{5}$$

where $[\bar{u}_{<i} \| \bar{u}_i]$ stands for the concatenation of a conversational context and an utterance of the i-th turn defined in the CANARD dataset with a separation token "|||" that indicates a boundary of utterances of different conversation turns. Finally, for CQR we adopt parameter and network architecture sharing as a simple strategy in transfer learning. Thus, after training on CANARD, we directly use

the Seq2Seq model with its optimized parameter set $\hat{\theta}$ to form our parametric model $g_{\theta}(\cdot)$ (i.e., $\theta = \hat{\theta}$) and directly use the model $g_{\theta}(\cdot)$ to reformulate q_i from the information set $\{u_i, u_{<i}\}$ of the CAsT dataset.

4 EXPERIMENT SETUP

4.1 Dataset

We conducted experiments on the dataset provided by the TREC 2019 Conversational Assistant Track (CAsT), a new task for conversational search research. The dataset consists of training and evaluation sets with 30 and 50 sessions, respectively, covering a wide range of open-domain topics. Each session contains approximately 10 turns, each of which includes a query and the relevant passages expected to be found. The corpus for the task is from passages in MS MARCO Passage Ranking collection (MARCO), TREC CAR paragraph collection v2.0 (CAR), and TREC Washington Post Corpus version 2 (WAPO). Near-duplicate paragraphs in the corpus are handled with the TREC CAsT tools,[3] yielding a total of 46 million candidate passages.

As shown in Table 2, of the 30 training set sessions, 13 have relevance judgments whereas 20 of the 50 evaluation set sessions have relevance judgments for final evaluation.

Table 2: CAsT judgment statistics

	Training[4]	Evaluation
#sessions (topics)	13	20
#turns	108	173
#assessments	2,399	29,571
#fails to meet (0)	1,759	21,451
#slightly meet (1)	329	2,889
#moderately meet (2)	311	2,157
#highly meet (3)	0	1,456
#fully meet (4)	0	1,618

4.2 Baseline Query Reformulation Methods

Best CAsT entry This baseline is one of our submissions to TREC 2019 CAsT, which uses an earlier version of the proposed HQE method in a two-stage ConvPR system. This submission resulted in the best automatic run of the 41 runs from 21 teams.

Raw query A simple baseline that adopts the original queries without any query reformulation.

Concat Another baseline that concatenates each query with the queries in its previous $\mathcal{M}$ turns, where $\mathcal{M}$ is a hyperparameter. A variant of this method is to filter out certain types of words from the queries in the previous $\mathcal{M}$ turns before concatenation. Here we filter out words with POS tags other than adjective and noun, using spaCy as the POS tagger.[5] This variant is also applied to the proposed HQE.

Manual TREC organizers manually rewrote the originally ambiguous queries according to conversational context.[6] As the rewritten queries contain all of the information required to represent a single query, we considered Manual as an empirical bound of human performance in our experiments.

4.3 Evaluation and Settings

Information retrieval model settings As mentioned in Section 3.1, we implemented a two-stage information retrieval pipeline with BM25 retrieval (first stage) and BERT re-ranking (second stage). The parameters for the BM25 model were $k_1 = 0.82$ and $b = 0.68$, for which the number of retrieved passages was set to 1000. We used the Anserini toolkit [55] for corpus indexing and BM25 retrieval. The fine-tuned BERT model for the second stage re-ranking was provided by [33].

Query reformulation model settings The hyperparameters were selected by grid search on the CAsT training set (see Section 5.2.2 for more detail), whereas the neural models (i.e., LSTM and T5) were directly applied to rewrite the queries with beam search decoding after training on the CANARD dataset. The detailed settings of the neural models are described as follows. (a) **LSTM (+Atten.)**: we adopted the bi-LSTM Seq2Seq model with attention, copy mechanism, and the same hyperparameter settings proposed in [21].[7] (b) **T5** [39]:[8] we used the T5-base model and its pretrained weights as the initialization and then fine-tuned it with the same hyperparameters used in [34].[9]

Evaluation For both stages, the results were evaluated by the overall ranking metric, mean average precision (MAP) at depth 1000, and the top-k ranking metrics NDCG@3 and NDCG@1. Note that NDCG@3 is the main metric used in CAsT. In addition, we report the values of recall at depth 1000 (R@1000) for first-stage retrieval. The evaluation was done using the TREC tool.[10] We also provide Win/Tie/Loss results based on R@1000 and MAP to show the number of queries whose performance was improved/unchanged/ deteriorated compared to manual query reformulation.

5 RESULTS

In this section, we first examine the effectiveness of the proposed HQE and NTR on the TREC-CAsT 2019 dataset; the results and analysis in terms of turn depths are also provided. Second, we study the impact of different query reformulation methods on passage re-ranking and provide the sensitivity analysis of our proposed HQE and NTR.

5.1 Main Results

Full ranking Table 3 ("Full ranking" columns on right) lists the final results with the two-stage ConvPR approach on the TREC-CAsT evaluation set. The listed performance is from the re-ranked results based on the corresponding 1000 retrieved passages obtained in the first stage, the performance of which can be found in the same row. We note that all the query reformulation methods outperform the baseline with raw query; the naive Concat method serves as a competitive baseline. The proposed HQE and neural methods beat the best entry in TREC-CAsT 2019 and are only 4% to 5% below manual queries. In particular, the ad-hoc HQE (+POS) marginally

[3] https://github.com/gla-ial/trec-cast-tools

[4] Note that training judgments are only graded on a three point scale (2 very relevant, 1 relevant, and 0 not relevant)

[5] https://github.com/explosion/spaCy

[6] https://github.com/daltonj/treccastweb

[7] Hyperparameter settings in https://github.com/aagohary/canard

[8] https://console.cloud.google.com/storage/browser/t5-data/pretrained_models/base

[9] https://github.com/castorini/docTTTTTquery

[10] https://github.com/usnistgov/trec_eval

Table 3: Performance on CAsT evaluation set. Win/Tie/Loss denotes the number of queries whose performance is improved/unchanged/deteriorated compared to manual query reformulation. The best results among single models for automatic query reformulation are in bold-faced. RRF denotes the reciprocal rank fusion of HQE (+POS) and NTR (T5).

		BM25						Full ranking (BM25+BERT re-ranking)			
Query reformulation		R@1000	W/T/L	MAP	W/T/L	NDCG@3	NDCG@1	MAP	W/T/L	NDCG@3	NDCG@1
Best CAsT entry		-	-	-	-	-	-	0.267	-	0.436	-
Manual		0.788	-	0.245	-	0.303	0.291	0.370	-	0.558	0.580
Raw query		0.404	4/59/110	0.100	3/56/114	0.127	0.126	0.161	4/55/114	0.243	0.243
Concat	Raw	0.488	11/30/132	0.092	12/22/139	0.175	0.176	0.171	9/21/143	0.325	0.347
	+POS	0.668	29/51/93	0.153	35/32/106	0.224	0.259	0.253	27/21/125	0.412	0.447
HQE	Raw	0.703	42/48/83	0.196	45/24/104	0.250	0.243	0.269	27/21/125	0.430	0.456
	+POS	0.715	**44**/58/71	0.203	**46**/28/99	0.243	0.242	0.285	**30**/22/121	0.455	0.462
NTR	LSTM (+Atten.)	0.516	13/58/102	0.126	11/49/113	0.168	0.145	0.216	24/36/113	0.335	0.339
	T5	**0.728**	9/**127**/37	**0.207**	10/**113**/50	**0.279**	**0.273**	**0.334**	**30**/**91**/52	**0.515**	**0.537**
RRF	HQE (+POS) NTR (T5)	0.794	60/65/48	0.241	76/26/71	0.309	0.323	0.348	78/17/78	0.536	0.548

outperforms the best CAsT entry, which is from an earlier version of the HQE paper, in terms of MAP and NDCG@3 by 6% and 4%, respectively, while NTR (T5) significantly surpasses the best entry by 30% in MAP and 18% in NDCG@3.

Comparing the results of Concat with and without the POS filter suggests that using adjectives and nouns accurately extracts keywords from historical queries. Although the POS filter further improves the performance of HQE, the proposed HQE without such filtering still yields competitive performance, indicating the effectiveness of our keyword extraction module in HQE. However, for the neural models, the LSTM trained from scratch performs poorly; in contrast, the fine-tuned T5 delivers state-of-the-art performance, illustrating that the pretrained weights provide a satisfactory initialization for neural query reformulation models.

Also listed in Table 3 is the detailed Win/Tie/Loss performance comparison of each query with its manual counterpart. The "Raw query" results indicate that 55 out of 173 original queries in the dataset are clear enough for the full ranking task whereas the other 114 original queries are ambiguous and effectively rewritten manually; only 4 raw queries yield better performance than the manually rewritten ones. The proposed HQE (+POS) and NTR (T5) methods, in turn, successfully generate 30 better quality queries for full ranking compared to the manual ones. We also note that nearly 50% of NTR (T5) rewritten queries (i.e., 82 of 173) yield the same performance as the manually rewritten ones, demonstrating the effectiveness of transfer learning, where we directly fine-tune the T5 model on the CANARD dataset and conduct inference for queries (query written) in CAsT.

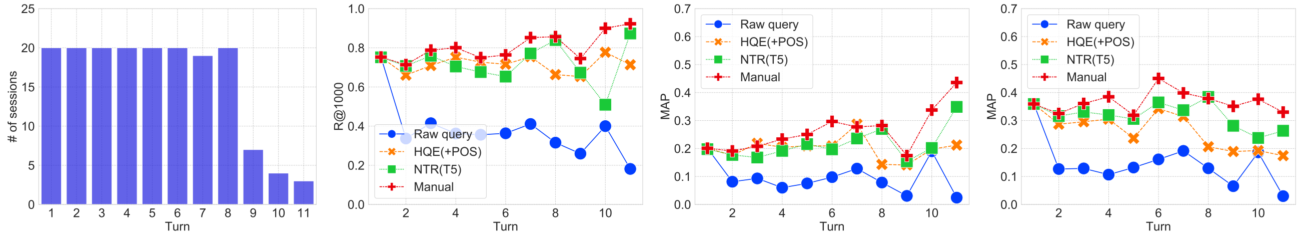
First stage retrieval with BM25 The effectiveness of the proposed query reformulation methods can also be observed from the results simply using the BM25 retriever in the first stage. As shown in Table 3 ("BM25" columns on left), the queries reformulated by HQE (+POS) and NTR (T5) both perform better than other baselines, leading to average performance improvement in terms of R@1000 over 70% and MAP around 20%. However, the Win/Tie/Loss comparison with manual queries shows the two methods improve the retrieval performance in a quite different way. Specifically, only less than 30% of the queries reformulated by NTR (T5) fail to beat their manual counterparts, which is far better than HQE (+POS) as it fails to rewrite 40% and 57% of queries regarding R@1000 and MAP, respectively. On the other hand, HQE (+POS) shows around 45 wins out of 173 queries, while only around 10 NTR (T5) rewritten queries beat the manual queries. It is surprising to find that these two methods achieve similar recall, but in a entirely different way, thus we also conduct detailed analysis in Section 6 to explore this.

Reciprocal rank fusion (RRF) [12] As HQE (+POS) and NTR (T5) improve the performance in a quite different way, we further fused the rank lists generated from HQE (+POS) and NTR (T5) with reciprocal rank fusion using the TREC tool.[11] The result is listed in the last row in Table 3. Observe that the fusion between the two lists in the first stage significantly outperforms the original ones and even yields performance comparable to manual queries, leading to more win than loss queries in terms of both R@1000 and MAP. Furthermore, the fusion from the two full ranking lists generates a better result with 0.348 and 0.536 performance in terms of MAP and NDCG@3, respectively.

Results by the turn depth Figure 1 compares the average recall and ranking performance of different reformulated queries in terms of the conversational turn depth. First, we observe that both the recall and ranking performance of raw queries (blue line) degrade abruptly after the first turn: conversational queries by nature become ambiguous as a dialogue moves forward. In contrast, HQE (+POS) and NTR (T5) yield stable recall performance over the turn depth with only slightly worse performance than the manual case. As for ranking performance, HQE (+POS) sees an obvious performance drop after the 7th turn in both stages, whereas NTR (T5) shows a slight performance decrease after the 8th turn, which suggests an advantage of NTR (T5) over HQE (+POS) in tracing deep conversation.

Summary We provide two strong query reformulation methods for conversational information retrieval: an ad-hoc HQE (+POS) and a neural NTR (T5) model. The experiments demonstrate that the reformulated queries effectively improve the performance of BM25 first-stage retrieval and BERT re-ranking. Furthermore, the two methods significantly outperform the best CAsT entry and achieve state-of-the-art performance for the CAsT full ranking task. Our

[11] https://github.com/joaopalotti/trectools



(a) # of sessions (per turn) **(b) BM25 (R@1000)** **(c) BM25 (MAP)** **(d) Full ranking (MAP)**

Figure 1: Performance comparison by turn depth

analysis also shows that HQE (+POS) and NTR (T5) improve query reformulation from different perspectives and that the reciprocal rank fusion between the ranking lists from the two methods further leads to better performance.

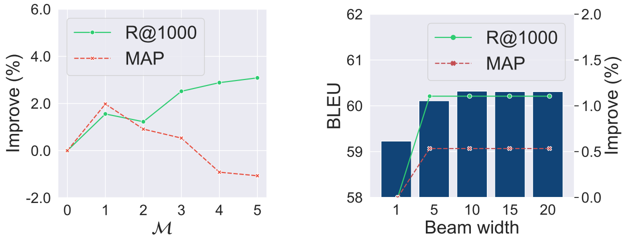
5.2 Component Evaluation

5.2.1 Effects on re-ranking. The performance of full ranking does not fairly reflect the effects of each query reformulation method on BERT re-ranking, as it is also affected by the quality of the retrieved passages in the first stage. Therefore, we conducted another experiment to examine the effects solely of re-ranking. Specifically, we first retrieved the top 1000 passages using manual queries with BM25 and re-ranked the top 1000 passages using the reformulated queries via different query reformulation methods with BERT. In this setting, all the reformulation approaches had the same passage pool for re-ranking, ensuring a fair comparison.

Table 4: Re-ranking passages retrieved with manually rewritten queries

Query reformulation		MAP	W/T/L	NDCG@3	NDCG@1
Manual		0.370	-	0.558	0.580
Raw query		0.212	5/55/113	0.276	0.266
Concat	Raw	0.281	36/21/116	0.441	0.447
	+POS	0.331	**50**/21/102	0.492	0.501
HQE	Raw	0.319	47/21/105	0.478	0.474
	+POS	0.330	**50**/23/100	0.505	0.529
NTR	LSTM (+Atten.)	0.274	27/40/106	0.385	0.387
	T5	**0.353**	28/**100**/45	**0.554**	**0.530**

Table 4 compares the results of passage re-ranking based on different query reformulation methods. Compared to the full ranking results, all query reformulation methods show better ranking results; NTR (T5)'s re-ranking performance especially closes with the manual case, with 0.353 in MAP and 0.554 in NDCG@3. Second to NTR (T5), both HQE (+POS) and Concat (+POS) obtain a 0.33 in MAP with over 30% of queries (50/173) beating the manual ones. HQE (+POS) and NTR (T5) show similar R@1000 and MAP performance using BM25 retrieval, but NTR (T5) yields significantly better BERT re-ranking results, perhaps because it generates queries more like natural language queries that are thus well-suited for the BERT re-ranker which was trained on natural language queries. Also note that although HQE (+POS) outperforms Concat (+POS) in top-k ranking performance (i.e., NDCG@3, NDCG@1), they have comparable overall ranking (MAP) performance in this task, which suggests that the proposed HQE outperforms Concat in full ranking (see Table 3) mainly due to the gain from first-stage BM25 retrieval.



(a) HQE (+POS) **(b) NTR (T5)**

Figure 2: Sensitivity analysis

5.2.2 Sensitivity analysis. We here conduct a sensitivity analysis on HQE (+POS) and NTR (T5) on the CAsT training set, where the hyperparameters of our CQR models are tuned based on their BM25 retrieval performance in terms of R@1000 and MAP.

HQE (+POS) Figure 2(a) shows the grid search results in R@1000 and MAP. Specifically, we tune $\mathcal{R}_{\text{topic}}, \mathcal{R}_{\text{sub}}, \eta$ and $\mathcal{M}$ for the optimal R@1000 and MAP separately. By fixing $(\eta, \mathcal{R}_{\text{topic}}, \mathcal{R}_{\text{sub}})$ at the best R@1000, $(10, 4.5, 3.5)$, and at the best MAP, $(12, 4.0, 3.0)$, Figure 2(a) shows the grid search results in terms of various $\mathcal{M}$.

We first note from Figure 2(a) that both R@1000 and MAP improve when $\mathcal{M} > 0$, indicating that adding subtopic keywords from previous $\mathcal{M}$ turns is effective for query expansion. In addition, R@1000 and MAP see different trends on the grid search, with the best $\mathcal{M} = 5$ and $\mathcal{M} = 1$ for R@1000 and MAP, respectively, suggesting that the optimal query for BM25 search is different in terms of R@1000 and MAP. Thus, in the previous experiments, we generated HQE (and Concat) queries using the hyperparameters with the best R@1000 for BM25 first-stage retrieval and those with the best MAP for BERT re-ranking.[12]

NTR (T5) We also analyze the sensitivity of beam width w in beam search decoding for NTR (T5) in Figure 2(b), where bars denote the BLEU scores — which is used for evaluating machine translated texts [44] and also served as a performance indicator in the work of Elgohary et al. [21] — (left y-axis) and lines denote the improvements of IR metrics compared to beam width $w = 1$ (right y-axis). Note that w stands for a number of partial sequences with highest probabilities we keep in order to find a single sequence with

[12] For Concat, the best $\mathcal{M}$ is 9. Due to the computational inefficiency of tuning the best hyperparameter for BERT re-ranking, we directly use the best one on BM25 search.

a limited-width bread first search in a context of sequential modeling. To determine the optimal hyperparameters for CAsT query inference, we consider the development (dev) set in CANARD and the training set of CAsT to choose w in the range of $\{1, 5, 10, 15, 20\}$. In specific, Figure 2(b) illustrates the BLEU score versus width in the CANARD dev set and R@1000 and MAP versus width in the CAsT training set. Observe that the best width, $w = 10$, achieves the highest BLEU (**60.32**) in the CANARD dev set,[13] whereas both R@1000 (+1.1 points) and MAP (+0.5 points) compared to $w = 1$ achieve the best performance at $w = 5$ in the training set of CAsT. To maintain query reformulation quality without hurting IR performance, we choose $w = 10$ in all of our experiments.

6 DISCUSSION

To further explore the distinct behaviors of HQE and NTR discovered in Sections 5, we present a study to unearth their differences from the following three perspectives:

(1) query characteristics from embedding space and pure texts;
(2) retrieval characteristics in terms of turn-depth-wise and session-wise aggregations;
(3) a case study that illustrates the models' pros and cons.

Note that our analysis is based on the 20 sessions with relevance judgments in the CAsT evaluation set.

6.1 Query characteristics

An intuitive way to illustrate the characteristics of conversational queries is to visualize their embedding using the BERT encoder. This intuition, which comes from the MS MARCO conversational search task,[14] is based on an assumption that utterances in the same conversation session are similar in the embedding space, as they are topic-oriented. We here leverage the BERT [17] model to project the reformulated queries—Raw query, HQE (+POS), NTR (T5), and Manual—into the embedding space and apply 2-dimensional t-distributed stochastic neighbor embedding (t-SNE) [30] on them altogether to make sure they are in the same embedding space.[15] Panels (a)–(d) in Figure 3 visualize their respective t-SNE embeddings, where color represents different session identifications (IDs) and the embedding size reflects turn depth.

From Figure 3 we note the following. First, the Raw query in panel (a) shows unclear boundaries between sessions, especially in the central region. This could be attributed to the ambiguity from coreferences and omissions in conversational utterances, as it is difficult to differentiate them without context. Second, Manual and NTR (T5) in panel (c) and (d) form more clear clusters between sessions than Raw query, suggesting their queries are more topic-oriented. Furthermore, observe that NTR (T5) and Manual obtain similar embedding distributions, implying that the two models yield similar queries, thereby leading to the many Ties in Tables 3 and 4. Finally, HQE (+POS) in panel (b) forms clear clusters—queries in the same session heavily overlap, suggesting queries reformulated by HQE (+POS) are similar within the same session.

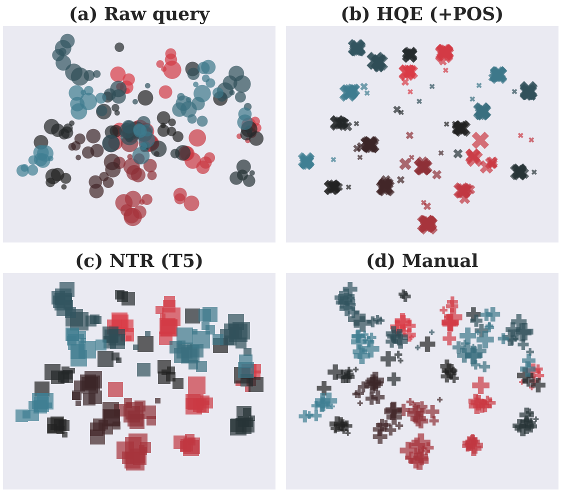
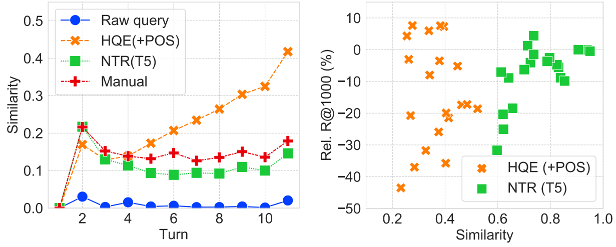


Figure 3: t-SNE plot of 20 sessions in evaluation set

To further attest the high similarity between NTR (T5) reformulated queries and Manual ones, we measure their query similarities quantitatively by pure text. Specifically, we compare query texts from different CQR methods with BLEU [44].[16] Here, we take the Manual queries as the reference sentences and calculate BLEU scores for the other methods. As shown in Table 5, NTR (T5) queries yield the highest score, whereas HQE (+POS) queries have the lowest. Note that raw queries yield the medium score among all. These results not only validate the high query similarity between NTR (T5) and Manual observed in Figure 3 but also show that HQE (+POS) generated queries are markedly different from other methods.

Table 5: BLEU with Manual as reference

Model	Raw query	HQE (+POS)	NTR (T5)
BLEU	60.41	33.73	**76.22**



(a) Turn similarity **(b) Session similarity**

Figure 4: Retrieved set analysis

6.2 Retrieval characteristics

Above, we clarified the distinct behaviors of two CQR approaches from a query perspective. To further uncover the reasons behind the Wins and Ties of HQE (+POS) and NTR (T5) versus Manual queries in Table 3, we analyze the similarities of the retrieved sets when different CQR methods are adopted. In Figure 4, the sets retrieved by

[13] T5 achieves better BLEU than 51.37 of LSTM (+Atten.) in the dev set of CANARD [21].
[14] https://github.com/microsoft/MSMARCO-Conversational-Search
[15] Note that we here follow the setup for building artificial conversational sessions from Bing search queries (see footnote 14 for details).
[16] We used multi-bleu-detok.perl from [21, 44].

Table 6: Comparison of queries of manual vs HQE (+POS) and NTR (T5) in session 32

Turn	Raw query	Manual	HQE (+POS)	NTR (T5)
1-5	(*We provide Raw queries here as context*): (1) What are the different types of sharks? (2) Are sharks endangered? If so, which species? (3) Tell me more about tiger sharks. (4) What is the largest ever to have lived on Earth? (5) What's the biggest ever caught?			
6	What about for great whites?	What about for **great whites**?	**sharks** sharks **tiger** sharks largest Earth biggest great whites What about for great whites?	What about for **great whites**?
R@1000	0.177	0.177	0.824	0.177
7	Tell me about makos.	Tell me about **Mako sharks**.	sharks sharks tiger sharks largest Earth biggest makos Tell me about makos.	Tell me about **makos**.
R@1000	0.273	1.000	1.000	0.273
8	What are their adaptations?	What are **Mako shark adaptations**?	**sharks sharks tiger sharks** largest Earth biggest **makos adaptations** What are their **adaptations**?	What are **makos adaptations**?
R@1000	0.000	1.000	0.941	0.765

Table 7: Comparison of queries of manual vs HQE (+POS) and NTR (T5) in session 54

Turn	Raw query	Manual	HQE (+POS)	NTR (T5)
1-4	(*We provide Raw queries here as context*): (1) What is worth seeing in Washington D.C.? (2) Which Smithsonian museums are the most popular? (3) Why is the National Air and Space Museum important? (4) Is the Spy Museum free?			
5	What is there to do in DC after the museums close?	What is there to do in Washington D.C. after the museums close?	worth Washington D.C. Smithsonian museums Space Museum Spy Museum DC museums What is there to do in DC after the museums close?	What is there to do in DC after the **Smithsonian** museums close?
R@1000	0.579	0.368	0.526	0.632
6	What is the best time to visit the reflecting pools?	What is the best time to visit the reflecting pools in Washington D.C.?	worth Washington D.C. **Smithsonian museums Space Museum Spy Museum DC museums** pools What is the best time to visit the reflecting pools?	What is the best time to visit the reflecting pools of Washington D.C.?
R@1000	0.250	1.000	0.000	1.000
7	Are there any famous foods?	Are there any famous foods in **Washington D.C.**?	worth Washington D.C. **Smithsonian museums Space Museum Spy Museum DC museums** pools famous foods Are there any famous foods?	Are there any famous foods in **Washington D.C.**?
R@1000	0.000	0.500	0.000	0.500

BM25 are analyzed in a turn-depth-wise perspective in panel (a) and in a session-wise perspective in panel (b). Specifically, we consider the Jaccard similarity $J(\cdot)$ to quantitatively analyze the retrieved sets. Note that in Figure 4(a), the similarity for turn i is the averaged values of the Jaccard similarities between the $(i-1)$-th and i-th turns over all sessions. Figure 4(b), in turn, takes the retrieved sets from Manual query as the reference sets to calculate relative (rel.) R@1000 and $J(\cdot, \mathcal{P}_{\text{Manual}})$ of NTR (T5) and HQE (+POS) versus Manual; then, a pair of average metrics (rel. R@1000, $J(\cdot)$) over all turns in each session is illustrated as a point on the figure.

We draw three conclusions from Figure 4. First, the Ties of Manual and NTR (T5) in Table 3 could be explained by the observations from panel (a) and (b) in Figure 4. As shown in panel (a), whereas the retrieved sets' similarities of NTR (T5) and Manual stay around 0.15 as the turns proceed, NTR (T5) also mainly centralizes around 0 on the y-axis in panel (b). Second, we conjecture the Wins of HQE (+POS) in Table 3 come along with the upper-left clustering in Figure 4(b); this could be due to the disparate behaviors observed in Figure 4(a)—HQE (+POS) tends to retrieve similar sets as the turns proceed. Finally, Figure 4 illustrates not only a significant gap between HQE (+POS) and NTR (T5) in panel (a) but also a clear boundary at 0.55 of the x-axis in panel (b). These observations suggest that the success of the fusion approach (RRF) could be attributable to the dissimilar behaviors of these two methods, which balance the biases from the two models [12].

6.3 Case Study

Tables 6 and 7 present two examples from sessions 32 and 54 to showcase the pros and cons of HQE (+POS) and NTR (T5). The row under each turn's query texts also shows the BM25 retrieval performance (R@1000) of the four reformulation methods: Raw query, Manual, HQE (+POS), and NTR (T5).[17]

Table 6 compares the reformulated queries about sharks and shows that the queries reformulated by NTR (T5) lose the context word *shark* after turn 6. Furthermore, from turns 7 to 10, NTR (T5) considers the context as *makos* rather than *makos shark*; hence, NTR (T5) is unlikely to retrieve passages with *makos shark* compared to HQE (+POS) and Manual. However, HQE (+POS) performs better in terms of R@1000 and the Win mainly due to the concatenation of the topic keyword *shark*. Especially in turn 6, HQE (+POS) significantly outperforms NTR (T5) and Manual, the main reason being that the words *great white* in NTR (T5) and Manual guide the BM25 model to retrieve documents with both *great* and *white* but not relevant to *shark*. This example also demonstrates that human rewriting queries are not always applicable.

On the other hand, HQE (+POS) can sometimes be too aggressive in injecting context into utterances. As shown in Table 7, HQE (+POS) emphasizes too much about "museum" when the subtopics

[17] Due to space limitation, we only provide raw queries from earlier turns as context, for which HQE (+POS) and NTR (T5) have similar performance.

have changed to *reflecting pool* in turn 6 and *food (D.C. half smoke)* in turn 7. On the contrary, the NTR (T5) mimics human to put adequate contexts in the utterances. For instance, as shown in the table, NTR (T5) puts *Washington D.C.* in turn 7 as sufficient contexts for BM25 model to understand the raw utterance. Moreover, take turn 5 as an example; NTR (T5) can sometimes address the context missing issue (i.e., adding the word *Smithsonian*) introduced by human writers, thereby making NTR (T5) outperform Manual query rewriting in few cases.

7 CONCLUSION

We present HQE and NTR, both conversational query reformulation methods stacked on a successful multi-stage IR pipeline. The effectiveness of our methods are attested by experiments on the CAsT benchmark dataset, the results of which suggest that the two methods have different advantages in fusing context information into conversational user utterances for downstream IR models. Finally, this work elevates the state of the art in CAsT benchmarks and provides simple but effectives baselines for future research.

ACKNOWLEDGMENTS

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Ministry of Science and Technology in Taiwan under the grants MOST MOST 107-2218-E-002-061. Additionally, we would like to thank Google for supporting this work by providing Google Cloud credits via the TensorFlow Research Cloud program.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-Task Learning for Document Ranking and Query Suggestion. In *Proc. ICLR.*

[2] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context Attentive Document Ranking and Query Suggestion. In *Proc. SIGIR.* 385âĂŞ394.

[3] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to Document Retrieval with Birch. In *Proc. EMNLP: System Demonstrations.* 19–24.

[4] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proc. SIGIR.* 475âĂŞ484.

[5] Nima Asadi and Jimmy Lin. 2013. Document Vector Representations for Feature Extraction in Multi-Stage Document Ranking. *J. Inf. Retr.* 16, 6 (2013), 747âĂŞ768.

[6] Nima Asadi and Jimmy Lin. 2013. Effectiveness/Efficiency Tradeoffs for Candidate Generation in Multi-Stage Retrieval Architectures. In *Proc. SIGIR.* 997âĂŞ1000.

[7] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv:1611.09268* (2016).

[8] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proc. ACL*, Vol. 1. 1870–1879.

[9] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proc. EMNLP.* 1724–1734.

[10] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036* (2018).

[11] Charles L. A. Clarke, J. Shane Culpepper, and Alistair Moffat. 2016. Assessing efficiency–effectiveness tradeoffs in multi-stage retrieval systems without using relevance judgments. *J. Inf. Retr.* 19, 4 (2016), 351–377.

[12] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proc. SIGIR.* 758âĂŞ759.

[13] Stephen Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *Proc. SIGIR.* 299–306.

[14] J. Shane Culpepper, Charles L. A. Clarke, and Jimmy Lin. 2016. Dynamic Cutoff Prediction in Multi-Stage Retrieval Systems. In *Proc. ADCS.* 17âĂŞ24.

[15] J. Shane Culpepper, Fernando Diaz, and Mark D. Smucker. 2018. Research Frontiers in Information Retrieval: Report from the Third Strategic Workshop on Information Retrieval in Lorne. *SIGIR Forum* 52, 1 (2018), 34âĂŞ90.

[16] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering. In *Proc. ICLR.*

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805* (2018).

[18] Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904* (2017).

[19] Laura Dietz and Nick Craswell. 2018. TREC Complex Answer Retrieval Overview. *Proc. TREC.*

[20] Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *arxiv:1704.05179* (2017).

[21] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. In *Proc. EMNLP.* 5917–5923.

[22] Luke Gallagher, Ruey-Cheng Chen, Roi Blanco, and J. Shane Culpepper. 2019. Joint Optimization of Cascade Ranking Models. In *Proc. WSDM.* 15âĂŞ23.

[23] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. 2019. Using Pre-Training Can Improve Model Robustness and Uncertainty. In *Proc. ICML*, Vol. 97. 2712–2721.

[24] Hsin-Yuan Huang, Eunsol Choi, and Wen tau Yih. 2019. FlowQA: Grasping Flow in History for Conversational Machine Comprehension. In *Proc. ICLR.*

[25] Dalton Jeffrey, Chenyan Xiong, and Jamie Callan. 2019. CAsT 2019: The Conversational Assistance Track Overview. In *Proc. TREC.*

[26] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA Reading Comprehension Challenge. *Trans.of ACL* 6 (2018), 317–328.

[27] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Trans. of ACL* (2019).

[28] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational Question Reformulation via Sequence-to-Sequence Architectures and Pretrained Language Models. *arXiv:2004.01909* (2020).

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692* (2019).

[30] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9 (2008), 2579–2605.

[31] Hafeezul Rahman Mohammad, Keyang Xu, Jamie Callan, and J. Shane Culpepper. 2018. Dynamic Shard Cutoff Prediction for Selective Search. In *Proc. SIGIR.* 85âĂŞ94.

[32] Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-Oriented Query Reformulation with Reinforcement Learning. In *Proc. EMNLP.* 574–583.

[33] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[34] Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery. (2019).

[35] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[36] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Trans. on Knowl. and Data Eng.* 22, 10 (2010), 1345–1359.

[37] Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with History Answer Embedding for Conversational Question Answering. In *Proc. SIGIR.* 1133âĂŞ1136.

[38] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *Proc. CHIIR.* 117âĂŞ126.

[39] Colin Raffel, Noam 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 preprint arXiv:1910.10683* (2019).

[40] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proc. ACL*, Vol. 2. 784–789.

[41] Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Trans. of ACL* 7 (2019), 249–266.

[42] Gary Ren, Xiaochuan Ni, Manish Malik, and Qifa Ke. 2018. Conversational Query Understanding Using Sequence to Sequence Modeling. In *Proc. WWW.* 1715âĂŞ1724.

[43] Haggai Roitman. 2019. Normalized Query Commitment Revisited. In *Proc. SIGIR.* 1085âĂŞ1088.

[44] Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proc. EACL: Software Demonstrations*. 65–68.

[45] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. In *Proc. ICLR*.

[46] Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. 2019. On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186* (2019).

[47] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text. In *Proc. EMNLP*. 4231–4242.

[48] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proc. NIPS*. 3104âĂŞ3112.

[49] Nicola Tonellotto, Craig Macdonald, and Iadh Ounis. 2013. Efficient and Effective Retrieval Using Selective Pruning. In *Proc. WSDM*. 63âĂŞ72.

[50] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A Machine Comprehension Dataset. In *Proc. Workshop on Representation Learning for NLP*. 191–200.

[51] Zhucheng Tu, Matt Crane, Royal Sequiera, Junchen Zhang, and Jimmy Lin. 2017. An Exploration of Approaches to Integrating Neural Reranking Models in Multi-Stage Ranking Architectures. *arXiv preprint arXiv:1707.08275* (2017).

[52] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. Question Rewriting for Conversational Question Answering. *arXiv:2004.14652* (2020).

[53] Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018. Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering. In *Proc. ICLR*.

[54] J. Xu and W. B. Croft. 1996. Query Expansion Using Local and Global Document Analysis. In *Proc. SIGIR*. 4–11.

[55] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using Lucene. *JDIQ* 10, 4 (2018), 16.

[56] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proc. EMNLP*. 2013–2018.

[57] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv:1906.08237* (2019).

[58] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proc. EMNLP*. 2369–2380.

[59] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer Extraction as Sequence Tagging with Tree Edit Distance. In *Proc. NAACL*. 858–867.

[60] Yun Zhou and W. Bruce Croft. 2007. Query performance prediction in web search environments. In *Proc. SIGIR*. 543âĂŞ550.

[61] Shihao Zou, Guanyu Tao, Jun Wang, Weinan Zhang, and Dell Zhang. 2018. On the Equilibrium of Query Reformulation and Document Retrieval. In *Proc. SIGIR*. 43âĂŞ50.