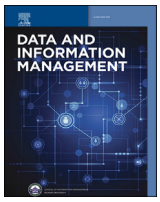
Data and Information Management 7 (2023) 100025

Contents lists available at ScienceDirect

Data and Information Management

journal homepage: www.journals.elsevier.com/data-and-information-management



Improving conversational search with query reformulation using selective contextual history

Haya Al-Thani [a,*], Tamer Elsayed [b], Bernard J. Jansen [a]

[a] Hamad Bin Khalifa University, Doha, Qatar
[b] Qatar University, Doha, Qatar

ARTICLE INFO

Keywords:
Conversational information seeking
Conversational search systems
Multi-stage retrieval systems
Open-domain

ABSTRACT

Automated responses to questions for conversational agents, known as conversation passage retrieval, is challenging due to omissions and implied context in user queries. To help address this challenge, queries can be rewritten using pre-trained sequence-to-sequence models based on contextual clues from the conversation's history to resolve ambiguities. In this research, we use the TREC conversational assistant (CAsT) 2020 dataset, selecting relevant single sentences from conversation history for query reformulation to improve system effectiveness and efficiency by avoiding topic drift. We propose a practical query selection method that measures clarity score to determine whether to use response sentences during reformulation. We further explore query reformulation as a binary term classification problem and the effects of rank fusion using multiple retrieval models. T5 and BERT retrievals are inventively combined to better represent user information need. Using multi-model fusion, our best system outperforms the best CAsT 2020 run, with an NDCG@3 of 0.537. The implication is that a more selective system that varies the use of responses depending on the query produces a more effective conversational reformulation system. Combining different retrieval results also proved effective in improving system recall.

1. Introduction

Conversational Search (CS) is an important research topic in the information retrieval (IR) and natural language processing (NLP) communities, and it was an essential topic in multiple sessions of the Third Strategic Workshop on IR (Culpepper et al., 2018) due to the increasing variety of devices accessible anytime-anywhere, often without a keyboard, as well as the advancements of speech interfaces. In addition to chatbots, improvements in machine learning techniques have led to the rapid popularity of conversational agents, e.g., digital personal assistants such as Apple's Siri and Amazon's Alexa (Mehrotra et al., 2020). These Conversational Agents perform well as task-oriented and social bots; however, they are not yet well suited in handling multi-turn conversational search.

In CS, the user's (often complex) information need is expressed in a sequence of queries or "turns" during a conversation with the system (Liu, 2021). Users often receive single conversation responses from the system and cannot scan multiple results (as in search browsers), modify their queries, or look at previous system results (Sa & Yuan, 2020). The user begins the conversation with a question expressing an initial information need. The system attempts to fulfill that need by retrieving answers to the query. The user can then ask follow-up questions of a related (or not) information need, entering a new turn in the dialogue. Subsequent turns are often ambiguous as stand-alone queries, making context-awareness of individual turns challenging to build conversational agents for CS. For example, looking at turn T2 in Table 1, the turn is difficult to understand without referencing the previous turn T1.

The annual Text REtrieval Conference Conversational Assistance Track (TREC CAsT) (Dalton et al., 2020a, 2020b) is a large-scale benchmark for open-domain CS where answers to turns are retrieved passages from an extensive passage collection. The CAsT 2020 dataset includes multi-turn conversations with a "canonical response" for every turn in the conversations. Canonical responses are passages selected by the organizers to represent relevant responses to turns. TREC organizers selected canonical responses from the top 5 results of the baseline system, which is a standard BM25 initial ranker followed by a BERT re-ranker. Table 1 shows the first three turns and corresponding canonical responses of topic 83, as an example. Even with the canonical responses, the challenge is how to ensure context-awareness throughout the conversation (Mele et al., 2021). Context-awareness can be achieved when ambiguous references are resolved for every turn in the conversation.

Currently, the most effective context-awareness solution for passage

* Corresponding author.
E-mail address: hayaalthani@hbku.edu.qa (H. Al-Thani).

https://doi.org/10.1016/j.dim.2022.100025
Received 5 September 2022; Received in revised form 2 November 2022; Accepted 22 November 2022

Table 1
TREC CAsT sample topic from the 2020 dataset.

T1:		**What are some interesting facts about bees?**
	R1:	Fun facts about bees … Honey never spoils.
T2:		**Why doesn't it spoil?**
	R2:	The water content … support microbial growth.
T3:		**Why are so many dying?**
	R3:	Honeybees are dying … industry in America itself.

retrieval is a multi-stage pipeline using conversational query reformulation. The conversational query reformulation (CQR) process takes the conversation history (i.e., some turns and responses from the conversation so far) concatenated with the current context-dependent *raw* turn and rewrites the raw turn by resolving co-references and omissions. *Context-dependent* queries, indicated by turns that contain omissions and co-references, can only be fully interpreted by providing the conversation history. The goal of the first stage of the pipeline is to convert context-dependent turns into *context-independent* turns that can then be used for more effective retrieval of passages that constitute potential answers to the question. Context-independent turns are self-contained queries that express the user's information need without omissions and co-references. CQR models can be built using pre-trained transformer models. In this research, a T5 transformer model is fine-tuned for this purpose (T5-CQR). After retrieval, passages go through one or more re-ranking stages that re-order passages according to their relevance.

This research aims to improve the query reformulation pipeline. We explore improvements from three perspectives. The first is to incorporate system responses into conversation history by reusing existing trained models originally built for other tasks such as next sentence prediction and question answering. We then compare text-to-text generative models with a more computationally efficient system that uses term-classification. Finally, we investigate how simple traditional IR solutions, such as query performance prediction and rank fusion can be incorporated to further improve performance. We found using Query Clarity Score for query prediction improves performance via selective conversation history. A more fascinating observation is that by leveraging the power of two pre-trained models (T5 and BERT), multi-model fusion was able to outperform state-of-art CAsT 2020. This implies many future possibilities of combining the strengths of different pre-trained models by complementing instead of competing against each other.

The key to effective CQR implementation is determining how responses can best be included in conversation history. Appending responses in their entirety into conversation history can introduce noise that degrades both efficiency and effectiveness (Wicaksono & Moffat, 2021). Therefore, we propose two response selection models to select a single sentence from the response; a BERT model trained for next sentence prediction (Devlin et al., 2019) and a T5 model trained for question answering (Raffel et al., 2020). Our intuition was that these existing models trained for other applications could be incorporated in a novel way for conversational response selection without the need for further fine-tuning, as next sentence prediction and question answering overlap with areas of conversational search. We also noticed in experimentation that responses might not be needed for every turn for CQR, so we propose adding a query clarity score (QCS) to determine whether reformulating a query benefits from the use of a response.

Training a model for CQR requires a large amount of manually-labeled training data due to its text generative nature (Hou et al., 2018). To date, there is no reasonably-sized dataset for conversational search tasks. The CAsT dataset is still relatively small when compared to other IR datasets. Other passage retrieval datasets such as MS Marco and TREC CAR have training sets that contain 530k and 3M queries with relevant passages respectively. Using T5 is also very computationally expensive due to the model's large parameter size. An alternative solution to work around this limitation is to view query reformulation as a binary classification problem. The model labels each term in the conversational history as either relevant or non-relevant. Relevant terms are then appended to context-dependent turns to resolve ambiguity. This method does not generate a grammatically well-formed turn but does add missing context as a set of keywords using a more efficient model. To properly investigate the benefits and costs of T5-CQR, we propose creating a BERT-based conversational term classifier (BERT-TC) and compare the performance of the two models.

Another interesting method that can improve our system is to explore how different reformulations of the same query affect retrieval and whether combining them improves performance. Rank fusion is when multiple system outputs are combined in order to better represent the user's information need (Fox & Shaw, 1994). We explore whether fusing multiple retrieved lists of passages from different query reformulation methods of the same turn improves the system's effectiveness. Combining query reformulation with T5 and BERT, we create a novel multi-model fusion solution that better represents user information needs. To the best of our knowledge, the effects of fusing query reformulations from different pre-trained models have not been explored. We also compare this technique with our proposed QCS query selection method. Re-ranking is an essential stage in the pipeline, but it adds to the complexity of the solution. We explore the effects of two-stage re-ranking and whether QCS and multi-model fusion benefit from "mono" versus "duo" re-ranking. A "mono" re-ranker (monoT5) reorders passages based on pointwise query-passage relevance, while a "duo" re-ranker (duoT5) reorders passages based on pairwise comparisons of two passages' relevance to a query (Pradeep et al., 2021). Table 2 details the different acronyms used in this paper.

In summary, this research aims to answer the following research questions:

- **RQ1:** Can existing trained models for next sentence prediction and question answering effectively select relevant responses in CQR?
- **RQ2:** Is query clarity score effective in determining when to use responses for query reformulation?
- **RQ3:** How does T5-CQR perform compared with smaller query reformulation models such as BERT-TC?
- **RQ4:** Can we apply rank fusion over multiple lists of retrieved passages using query variations from multiple models to improve system effectiveness? How does that compare to the proposed query clarity score selection function?
- **RQ5:** Would applying duoT5 re-ranking further improve the performance over monoT5 re-ranking?

Our contributions in this work are as follows:

- We demonstrate the performance of two pre-trained models as sentence selection methods for improving T5-CQR. This indicates that reusing models can efficiently incorporate response into CS without the need of training or fine-tuning new models specific for CS tasks.
- We establish a query clarity score that is effective at determining whether to use a response for T5-CQR during retrieval.
- We show the benefits of using T5-CQR compared to other models such as BERT-TC.

Table 2
Acronyms used in the paper.

Name	Description
CS	Conversational Search.
TREC	Text REtrieval Conference.
CAsT	TREC Conversational Assistance Track.
CQR	Conversational Query Reformulation.
T5	Text-to-text Transfer Transformer model.
BERT	Bidirectional Encoder Representations from Transformers model.
T5-CQR	T5 model fine-tuned for conversational query reformulation.
BERT-TC	BERT model fine-tuned for conversational term classification.
QCS	Query clarity score to predict query performance at retrieval.

- We combine query representations using both T5 and BERT reformulations using a novel multi-model fusion approach. We find that combining reformulations using both models improve system context-awareness by leveraging the strengths of the two pre-trained models.
- We show that combining multiple retrieved lists of passages can improve not only recall but also overall performance measured in NDCG@3.

The paper is organized in the following sections: first, available literature are reviewed to understand the current state of the field. Section 3 details the methodology of the proposed system and its different components. In section 4, the experiment setup is presented, as well as the metrics used to evaluate the system. Section 5 contains the results and respective discussion, which are followed by the conclusion and future work in section 6.

2. Related work

In this section, related work will be reviewed from two perspectives: multi-stage retrieval systems and conversational search systems.

2.1. Multi-stage retrieval systems

Multi-stage retrieval systems can be a two-step process. First, a list of candidate documents or passages are retrieved from a corpus, and then the list goes through one or more re-ranking phases to strike a balance between efficiency and effectiveness (Asadi & Lin, 2013). Research in this domain includes feature extraction efficiency, dynamic cut-off depth, and joint cascade ranking optimization (Lin et al., 2020a, 2020b). The baseline of this system is BM25 candidate generation followed by a BERT re-ranker.

The final phase of the multi-stage retrieval pipeline uses one or more re-ranking models that adapt pre-trained transformers for query and passage relevance classification (Lin et al., 2021a). The "mono" re-rankers receive as input a query and candidate passage that is scored based on their relevance. "Duo" re-rankers apply a pairwise approach where the re-ranker considers a pair of passages and predicts which passage is more relevant to a query. Similar to monoBERT by Nogueira and Cho (Nogueira & Cho, 2019), Nogueira et al. (Nogueira et al., 2020) applied the Text-to-Text Transfer Transformer (T5) model to re-ranking with monoT5. DuoT5 is a second stage re-ranker that receives the "mono" re-ranked passages and applies the same pairwise approach as duoBERT by Nogueira et al. (Nogueira et al., 2019a) but with the T5 model (Pradeep et al., 2021). The re-ranking models requires a large scale of labeled data and heavy computational load to train. Some research has been conducted to prove the potential of improving passage retrieval using other weak relevance signals when training these neural ranking models (Zheng et al., 2019).

Another retrieval technique used to improve performance is rank fusion. Rank fusion is a technique that combines knowledge from multiple system outputs or query variations to better express user information need (Fox & Shaw, 1994). This is done to optimize the order of a ranked list by investigating one or more features in a supervised or unsupervised approach. Algorithms can generally be categorized as score-based and rank-based fusion (Hsu & Taksa, 2005). Score-based systems depend on the information stored in retrieval scores. Rank-based systems depend on the order of documents in the ranked list.

2.2. Conversational search systems

Conversational search (CS) has many applications, such as e-health systems, recommendation systems, and personality recognition (Alian-nejadi et al., 2020). Rule-based conversational IR systems have given way to more advanced methods based on deep learning (Gao et al., 2018; Onal et al., 2018). One major factor to consider when designing a conversational agent is how to maintain conversation context (Vtyurina et al., 2017). Context also plays an important role in conversational response classification (Cui et al., 2020). Dialogue context can be used to identify malevolent or toxic conversation responses for building safer, more trustworthy chatbots (Almerekhi et al., 2022; Zhang et al., 2021).

One method to ensure context-awareness in multi-turn conversations is rewriting turns using query reformulation (Dehghani et al., 2017). Conversational query reformulation (CQR) uses pre-trained sequence--to-sequence (seq2seq) models to resolve user information need in ambiguous queries. Elgohary et al. (Elgohary et al., 2019) used a T5 model that takes a conversation's entire history, along with the query to be rewritten, and outputs a context-independent query outperforming the best CAsT 2019 baseline. Lin et al. (Lin et al., 2021b) fuse queries reformulated with a T5 model with another query expansion method that estimates query term importance using the BM25 score. They found that fusing the two query variations improved retrieval effectiveness on CAsT 2019 dataset.

Conversational query reformulation can also be expressed as a binary term classification problem. Terms in the conversational history are labeled as relevant or non-relevant to the current turn to resolve missing context. In the work of Voskarides et al. (Voskarides et al., 2020), BERT is trained using the QuAC dataset (Choi et al., 2018) to create a binary term classifier to decide whether to add terms to current turn for retrieval. Kumar and Callan (Kumar & Callan, 2020) train a BERT model using weak supervision to supplement limited available training data in CAsT.

Training data availability for CS is limited. The goal of TREC CAsT is to create a large-scale reusable test collection for open-domain conversational search where answers are retrieved passages from a large text corpus. CAsT 2020 submissions are categorized as: Automatic (using only raw turns), Auto-canonical (using raw turns and canonical response), and Manual (using manually rewritten turns). The H2oloo team achieved the best performance in the automatic and auto-canonical categories (Lin et al., 2020a). Their automatic run used a T5 system trained on CANARD. In their auto-canonical run, a sentence was selected from the canonical response using keyword matching. The second-place team, ASCFDA, also employed a fine-tuned T5 model (Chang et al., 2020). They break canonical responses down into sentences then apply the doc2query model (Nogueira et al., 2019b) to each sentence. Based on the resulting "latent" query, a sentence from the canonical response is added to the conversation history.

Another well-researched problem in CS is conversational response selection. Response selection in conversational search had earlier focused on single-turn response retrieval (Hu et al., 2014; Lu & Li, 2013; Wang et al., 2013). Single-turn systems only use the last utterance for response selection and ignore context from previous utterances. Neural Network models have been used to measure the relevance of context and response pairs in multi-turn conversations (Lowe et al., 2015). More recent work studies the effect of using too much context. Yuan et al. (Yuan et al., 2019) propose a multi-hop selector network that matches filtered context to candidate responses. Another simple yet effective solution is to split long response sentences into simpler components before sentence selection (Finegan-Dollak & Radev, 2016).

2.3. Position of our study

Literature review shows that previous works studied conversational search systems from multiple perspectives. Making sure context is preserved throughout the conversational turns remains essential for retrieval. Some models rewrite queries using historical context, while others include context using classification. This work explores query reformulation using a query rewriting model and compares it to a term classification model. Conversation responses also provide contextual clues as well as previous turns. Exploring response selection methods and how these can be included in CQR is another focus of this research. We finally investigate the benefits of using different variations of the same turn using a simple rank fusion method that combines the reformulated

queries from two different fine-tuned models, and then compare it with our proposed query clarity score selection method. Using our proposed solution achieves best performance for the CAsT 2020 dataset, beating the CAsT 2020 baseline. It is worth noting that several earlier studies use CAsT 2019 dataset for evaluation; however, the 2020 dataset is more challenging due to the increased complexity of the conversations. Turns in CAsT 2020 can reference both previous turns or responses, whereas CAsT 2019 conversations depend only on previous turns.

3. Methodology

The CS problem is defined as follows. A conversation is made up of a series of *N*-turn raw user's utterances $\{u_1, u_2, u_3, \ldots, u_N\}$. The task is to retrieve a list of top-*k* passages p_i for each turn u_i from a collection of passages to satisfy the information need of turn *i*. For each turn u_i, a canonical response c_i, represented as a sequence of sentences $\{s_i^1, s_i^2, s_i^3, \ldots, s_i^M\}$, is available. We also define the *raw* conversation history h_i of turn *i* as the entire sequence of the previous *raw* turns in the conversation, i.e., $h_i = \{u_1, u_2, u_3, \ldots, u_{i-1}\}$. Table 3 lists the various notations used in this paper.

We propose to solve the CS problem defined above using a multi-stage retrieval pipeline, as illustrated in Fig. 1. The pipeline consists of four main stages. The first stage is "response sentence selection", which selects a representative sentence from the canonical responses to include in the conversation history. After that, the "conversational query reformulation" (CQR) stage follows where raw turns are reformulated into context-independent turns. The third stage is "retrieval" where the context-independent turns are used to retrieve top-*k* passages. The fourth and final stage is passage "re-ranking" of the retrieved passages. We next discuss each stage in detail.

3.1. Stage 1: Response sentence selection

We propose two sentence selection models to incorporate a sentence from the canonical response into the conversation history before applying CQR. The goal is to select a single sentence to minimize topic drift and avoid lengthy input to CQR to improve system effectiveness.

Canonical responses represent the system's answer to the user's turn. These responses typically contain an average of 100–150 words. Ideally, within these responses is an answer to the user's information need. Responses are an integral part of the conversation as they often lead to follow-up turns or topic shifts further along. However, including responses in their entirety into conversation history could negatively impact system performance. It can introduce noise and incorrect context into conversation history. We propose re-purposing two existing pre-trained models to select a single sentence from the response. The selected sentence is the one most likely to have generated the following turn in the conversation. By doing so, we have included only relevant information into conversation history without needing to fine-tune a new model specifically for this task. This reduces the computational cost, since the two models are already fine-tuned and available for use.

Table 3
Notation used in the paper.

Name	Description
u_i	Raw conversation utterance at turn *i*. Raw turns are context-dependent.
m_i	Manual conversation utterance at turn *i*. Manual turns are context-independent.
p_i	List of retrieved passages for turn u_i.
c_i	Canonical response for turn u_i.
s_i^m	Single sentence of canonical response c_i.
h_i	Conversation history made up of previous raw turns at turn *i*.
x_i	Conversation context at turn *i* that can be made up of previous turns and responses.
r_i	Reformulated utterance at turn *i* rewritten by the trained model.
t_i	Set of context term for turn u_i. Context terms are terms that are in m_i but not u_i.

3.1.1. Next sentence prediction

The first sentence selection model uses a pre-trained BERT model for next sentence prediction (NSP) (Devlin et al., 2019). BERT pre-training includes next sentence prediction to predict how likely two sentences follow each other or not. The model is used to select a sentence from the canonical response that most likely "triggered" the follow-up question from the user. To select sentence s_{i-1} for turn u_i, previous turn canonical response c_{i-1} is divided into sentences $\{s_{i-1}^1, s_{i-1}^2, s_{i-1}^3, \ldots, s_{i-1}^M\}$. Each sentence is paired with the current turn u_i. The sentence with the highest probability is selected and added to the conversation history at that turn.

$$s_{i-1} = arg\,max_{s \in c_{i-1}} NSP(s, u_i) \quad (1)$$

3.1.2. Question answering

The second method we propose for sentence selection uses the T5 model fine-tuned for question answering (QA) using the SQuAD dataset (Rajpurkar et al., 2018). This model outputs either a sentence or sentence span. We refer to both here as a "sentence" for convenience. Instead of pairing questions and context in the turn, the turn u_i is paired with the previous canonical response c_{i-1}. The premise is that the turn is not necessarily "answered", but it is "conceptually related" to a sentence of the previous response. Our motivation for this second method is to extract the sentence that is most related to the subsequent turn. The selected sentence is then added to the conversation history as the response sentence s_{i-1}.

$$s_{i-1} = QA(c_{i-1}|u_i) \quad (2)$$

3.2. Stage 2: Conversational query reformulation

The second stage aims to take a raw context-dependent turn u_i and produce a context-independent turn. The two investigated models are T5-CQR, a T5 query rewriting model, and BERT-TC, a BERT binary term classification model. The goal is to measure which system reintroduces context back into turns more accurately and examine their efficiency-effectiveness trade-off; T5 is a powerful generative model that is computationally expensive to train and run, while BERT is simply fine-tuned as a binary classifier but might not be as effective.

CQR conceptually takes a context-dependent query and the context to produce a context-independent (i.e., reformulated) query. In our work, the context x_i at turn *i* is represented by the conversation history h_i and the selected sentence s_{i-1} from the canonical response, while the query is the raw turn u_i. CQR then produces a context-independent rewritten turn r_i that can be directly used for passage retrieval.

3.2.1. T5-CQR

This model uses the T5 model fine-tuned using the CANARD dataset (Elgohary et al., 2019), denoted as T5-CQR. T5 is a powerful model that translates NLP tasks into a text-to-text format using an encoder-decoder architecture (Raffel et al., 2019). T5 can be fine-tuned for various downstream tasks, such as translation or summarization. In this work, the model is fine-tuned for conversational query reformulation. The CANARD dataset turns were pre-processed by concatenating the *raw* conversation history at turn *i*, $h_i = \{u_1, u_2, \ldots, u_{i-1}\}$, with the raw turn u_i and using the manual turn m_i as the model target. Manual turns are turns manually rewritten to resolve context.

For this stage, we experimented with two setups. In the first, the context x_i of turn *i* is represented as the raw conversation history of turn *i* concatenated with the selected sentence from the canonical response, i.e., $x_i = \{u_1, u_2, \ldots, u_{i-1}, s_{i-1}\}$. This is paired with the raw turn u_i as input to T5-CQR. We denote this setup as "CQR-Historical Context" or CQR–HC–*S*, where *S* indicates the sentence selection method (i.e., NSP or QA). Fig. 2 illustrates the setup for the first three turns of topic 83 of CAsT 2020 dataset.

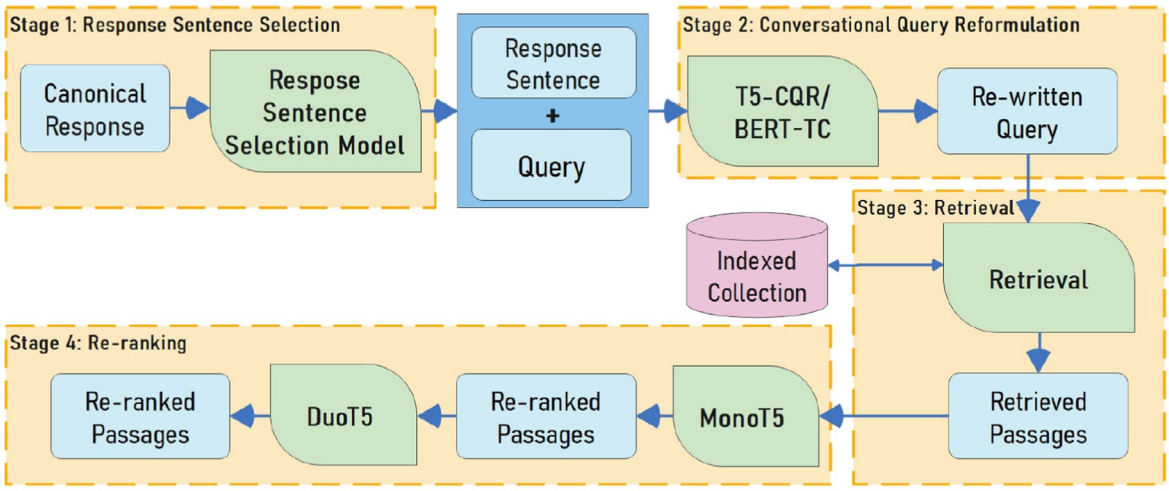


Fig. 1. Multi-stage pipeline.

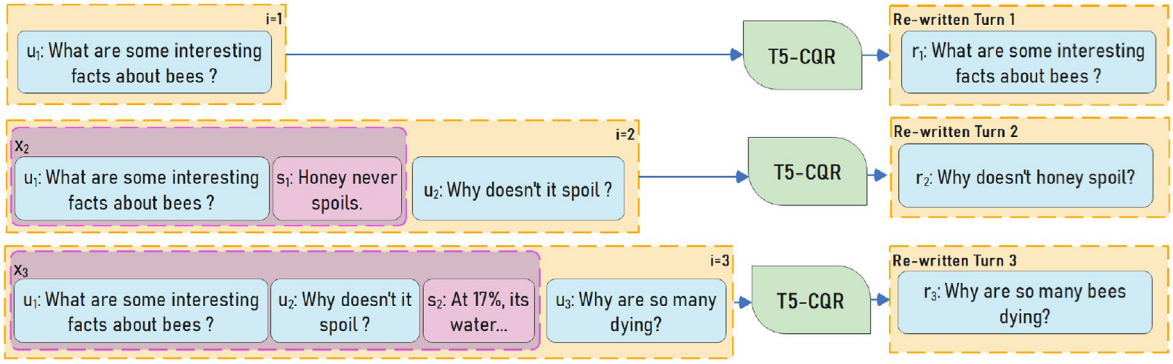


Fig. 2. CQR with historical context (CQR–HC–*S*).

In the second setup, we replace the context-dependent turns in x_i with context-independent rewritten turns r_i, as illustrated in Fig. 3. In this setup, the context is represented here as $x_i = \{r_1, r_2, \ldots, r_{i-1}, s_{i-1}\}$. We denote this setup as "CQR-Rewritten Historical Context" or CQR-RHC-*S*. This allows context retrieved from sentence s_{i-1} to propagate further into the conversation through r_i without adding more than one sentence to the context, reducing input length and tokens. This is because r_i can retain some context from s_{i-1} using CQR. For example, r_2 in Fig. 3 keeps the word "honey" in the context even when s_1 is no longer in the set.

In our experiments reported in the Experimental Evaluation Section, we also compare with variants of those setups in which the context does *not* include the selected sentence, denoted as CQR-HC and CQR-RHC, respectively.

3.2.2. *BERT-TC*

An alternative solution that adds context to turns uses BERT for term classification instead of the previous T5 model. Unlike T5, BERT is only built using encoder blocks and can only output a label classification or a span of the input text (Devlin et al., 2019). To use BERT for query reformulation, we have to take a classification-based approach. Modeling query reformulation as a binary classification problem is more computationally efficient than using the larger seq2seq model and requires less training data.

In order to train BERT-TC, each raw turn u_i should be associated with

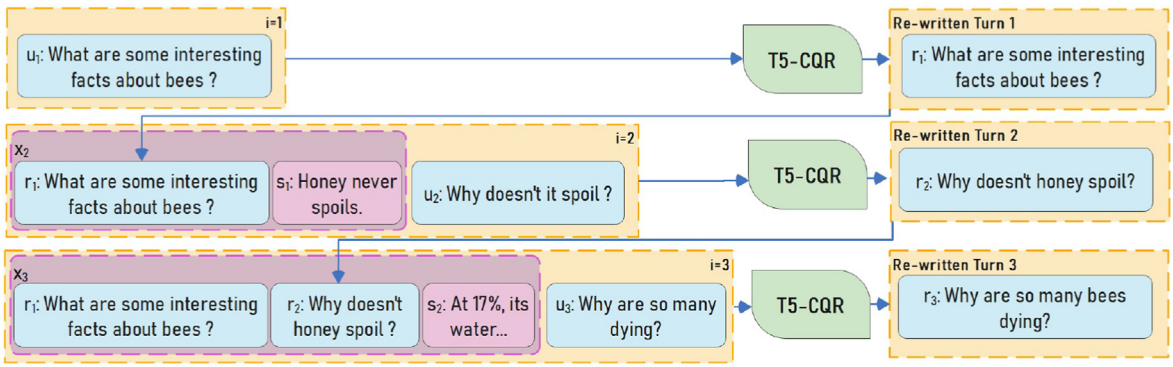


Fig. 3. CQR with rewritten context (CQR-RHC-*S*).

a set of relevant context terms t_i. Context terms can be inferred using manual turns m_i provided in available datasets. The set t_i can be constructed as all terms in the manual turn $m_i = \{m_i^1, m_i^2, m_i^3, \ldots, m_i^J\}$ that are not in $u_i = \{u_i^1, u_i^2, u_i^3, \ldots, u_i^K\}$. The model is provided with a conversation context x_i which is the conversation history usually made up of previous turns, and the current raw turn u_i. Terms in t_i present in conversation context x_i are labeled relevant while other terms are non-relevant. For example, the set of context terms of turn 3 of topic 83, u_3 = "Why are so many dying?", is created using the manual turn m_3 = "Why are so many bees dying?". The set of context terms for this turn will be $t_3 = \{\text{bees}\}$.

The classifier uses a BERT model with an added classification layer that receives the encoded terms from BERT and classifies them in order to determine which terms should be selected (as illustrated in Fig. 4). The model receives as input the conversation context x_i at turn i and the current turn u_i, and outputs a binary label (relevant or non-relevant) for each term in x_i. Relevant terms are concatenated to the current turn u_i to resolve context. The output of the model when resolving turn 3 of topic 83 is: "Why are so many dying? bees".

We experiment with different constructions for x_i. The first uses only historical turns as context, i.e., $x_i = \{u_1, u_2, \ldots, u_{i-1}\}$. This system is denoted as TC–HC–*I*, where *I* refers to how far back history is included into context (i.e., add only last two turns or three turns and so on).

The other set-up is TC-*S*-*I*, where *S* indicates the sentence selection method (i.e., NSP or QA). In this set-up, x_i is composed of both historical turns and response sentence, for example, $x_i = \{u_1, s_1, u_2, \ldots, u_{i-1}, s_{i-1}\}$. With BERT-TC we can experiment with adding multiple response sentences instead of only one like in T5-CQR. This is because the model is smaller and adding these extra tokens will not increase computation time as much as in T5-CQR. However, adding too many response sentences can still introduce topic drifts, which should be considered when constructing the input.

3.3. Stage 3: Retrieval

This stage aims to retrieve a pool of potentially relevant passages using queries reformulated using both T5-CQR and BERT-TC. Reformulated turns are issued as queries to our BM25 retrieval engine to get an initial ranked list of passages. However, not every turn will benefit from CQR that uses the selected sentence from the canonical response. Some turns might only depend on the previous raw turns without reference to the previous response. One example of such a turn would be turn T3 from Table 1. T3 refers to T1, but not any of the previous responses. On the other hand, turn T2 refers to R1. Intuitively, at every turn, we can either use the turn rewritten using only the raw conversation history, denoted as r^h, or the one also using the selected response sentence, denoted as r^s.

3.3.1. Query clarity score

We propose a *query clarity score* (QCS) that uses query prediction measures to decide when to include response for retrieval. Query clarity measures were introduced in the domain of "query performance prediction" to measure the coherence of queries as an indicator of their performance (Cronen-Townsend et al., 2002).

We experimented with three query clarity measures. The first, denoted as BM25-CL, uses the BM25 retrieval score of the top retrieved passage to measure the clarity of queries (Lin et al., 2020a). As this score might not be reasonably comparable across different queries, we also experimented with a normalized version, denoted *n*BM25-CL, in which we first normalized the BM25 scores of the retrieved passages for each query using Z-normalization, then considered the normalized score of the top one. As those two measures require passage retrieval (which is naturally expensive), we experimented with a third measure, the sum of the IDF scores of the query terms, denoted as IDF-CL. That one is a "pre-retrieval" measure that avoids issuing queries while accounting for the informativeness of the query terms. QCS is measured for the two variations of the turn r^h and r^s. The variation with the higher QCS is then used to retrieve top-k passages.

3.3.2. Multi-model fusion

Furthermore, we can combine multiple retrieved lists of passages into a single top-k list to potentially increase the system recall and performance using rank fusion (Hsu & Taksa, 2005). To explore this effect, the retrieved passages from two different query reformulations, r^h and r^s, can be combined into a single list of passages and then passed on to the re-ranker. We experimented with a simple rank-based method, where passages ranked 1 are added first, then passages ranked 2, and so on, while making sure not to add duplicate passages.

We explore combining retrieved passages from queries generated using T5-CQR, r^h and r^s, and queries generated using BERT-TC, r^{tc}. T5 reframes all NLP tasks to a text-to-text format where inputs and outputs are always strings of text. In contrast, BERT only outputs a class label or a span of the input. The two model architectures are also different, since BERT only has encoder blocks while T5 is built using both encoder and decoder. We explore how these two different encoder-only and encoder-decoder architectures perform for query reformulation and whether joining the two systems using rank fusion improves performance. The two models are also trained and fine-tuned using different data. By creating a multi-model fusion of both pre-trained models, we can leverage the strength of the two and improve context-awareness. We are testing the fusion of BERT term classification with T5 conversational query reformulation, with the implication that this method can be used on other pre-trained model combinations in the future.

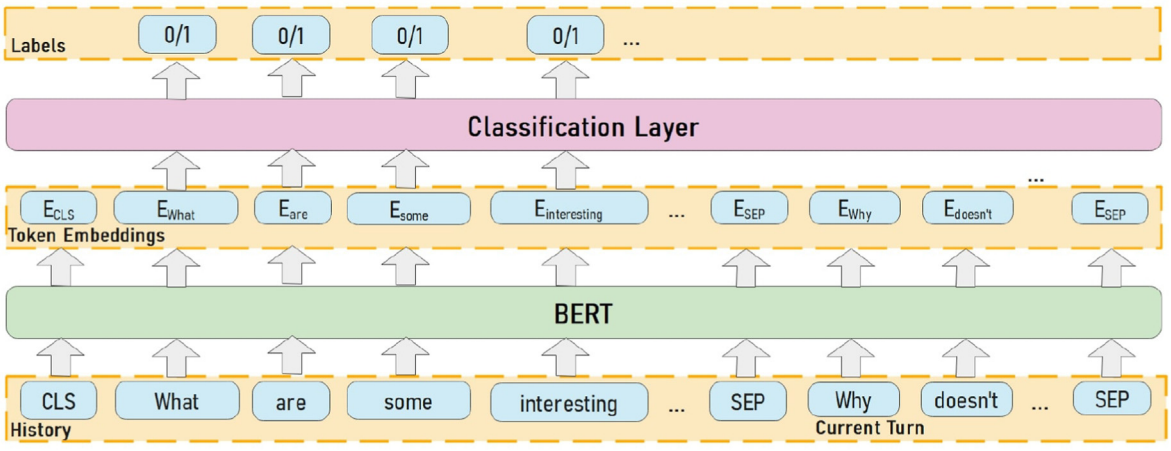


Fig. 4. BERT for term classification.

3.4. Stage 4: Re-ranking

After retrieval, the final stage is re-ranking the retrieved passages to get the top 1000 passages. We use the monoT5 re-ranking model proposed by Nogueira et al. (Nogueira et al., 2020) as an initial re-ranking step. MonoT5 receives query passage pairs and scores them based on their relevance. The passage list is later re-order according to the relevance of the query passage pairs.

After that, the output of monoT5 is used as input to duoT5 (Pradeep et al., 2021). DuoT5 considers a pair of passages and predicts which passage is more relevant to the query. Passages are then re-ordered from the most to the least relevant to the query. This process is computationally expensive, so it is best to run duoT5 only on a small number of candidate passages. The goal is to improve the quality of the passages ranked high on the list. DuoT5 is used to re-rank the top 100 passages from the list of 1000 retrieved passages. This means that the first 100 passages will be re-ranked using duoT5, while passages ranked 101 to 1000 will remain as they were ranked by monoT5.

4. Experimental setup

We evaluated our systems using the TREC CAsT 2020 test collection (Dalton et al., 2020b). The CAsT 2020 test set contains a total of 216 turns across 25 conversation topics with an average length of 8.6 turns per topic. Topics in the dataset were developed to ensure conversations are complex, diverse, open-domain, and answerable. The CAsT 2020 test set is used because of its dynamic, open-domain conversations that can shift topics based on either historical turns or responses. This makes it ideal for experimenting the effects of selective history. CAsT 2019 and other datasets are not used, as conversations are either not open-domain, or do not change topics based on system response.

The first response sentence selection model is a pre-trained BERT for next sentence prediction trained on a large corpus of English data (Devlin et al., 2019). This is built using the BERT-base-uncased model with 12 transformer blocks, 768 hidden layers, 12 attention heads, totaling 110 million parameters as part of the HuggingFace Transformer Library.[1] The other response sentence selection model is also available in the HuggingFace Transformer Library,[1] T5 for question answering. This model is built using base T5 fine-tuned using the SQuaD dataset (Rajpurkar et al., 2018). Base T5 has 12 layers, 768 hidden-states, 3072 feed-forward hidden-states, 12 heads with 220 million parameters.

The T5-CQR model was initialized with pre-trained weights, and training hyper-parameters were set as proposed by (Lin et al., 2020a). For fine-tuning the model, the CANARD dataset was pre-processed using the setup proposed by (Elgohary et al., 2019). For the model inputs, all historical utterances along with the query to be rewritten were concatenated with a special separator token between each utterance and appended to the manual context-independent query as the training target. T5 is fine-tuned with a constant learning rate of 0.001 for 4k iterations. The maximum input tokens were set to 512 with 64 output tokens. None of the inputs needed to be truncated. A single Google Cloud Platform TPU v3-8 was used to train both base T5 and large T5. Base T5 has 220 million parameters as described previously, while large T5 is built with 24 layers, 1024 hidden-states, 4096 feed-forward hidden--states, 16 heads with 770 million parameters.

BERT-TC uses the BERT-large-uncased model built using the HuggingFace Pytorch implementation.[1] The model contains 24 transformer blocks, 1024 hidden layers, and 16 attention heads, which add up to 340 million parameters in total. Training hyper-parameters for BERT-TC were set as proposed by (Voskarides et al., 2020). BERT is fine-tuned using the Adam optimizer with an initial learning rate of 5e-5 and a dropout rate of 0.4 on the layers. The model is fine-tuned using the OR-QuAC dataset (Qu et al., 2020) after applying lower-casing, lemmatization, and stopword removal to create the context terms set.

The Anserini toolkit [2] was used for indexing and retrieval. We used BM25 to retrieve the top 1000 passages from the MS MARCO passage collection and TREC Complex Answer corpora. For re-ranking, we used the 3B-parameter monoT5 and duoT5 re-rankers with the setting proposed by (Nogueira et al., 2020) and (Pradeep et al., 2021) respectively. The re-rankers were trained with a constant learning rate of 0.001 for 100k iterations for monoT5 and 50k iterations for duoT5. The re-ranking models are available in PyGaggle,[3] a neural re-ranking library.

We compare the performance of several variants of our system. Systems that do not consider the canonical responses for the context are denoted as "auto" systems, while those that consider the canonical responses are denoted as "auto-canonical" systems. We compare our systems performance with simply using the context-dependent turns without CQR. This run is denoted as "raw turn". Paired sample *t*-test is conducted between internal system variants.

As external baselines, we also compare with the best submitted auto and auto-canonical runs to CAsT 2020 as a strong baseline (Lin et al., 2020a). Auto and auto-canonical CAsT 2020 baselines use base T5 and large T5, respectively, followed by a monoT5 re-ranker. External baseline systems also use a combination of neural and spare retrieval to improve system recall (Lin et al., 2020a, 2020b). The external baselines are current state-of-art for the CAsT 2020 dataset. More detailed performance analysis of the external baselines were unavailable for *t*-testing as only the averaged performance measures are released for all baselines.

We evaluated the system performance using Recall@1000 for retrieval. We used NDCG@3 as the main evaluation metric as well as MAP@1000 for post re-ranking evaluation (Dalton et al., 2020b).

5. Experimental evaluation and results

In this section, we present the results of our experiments to answer the respective research questions. First, we explore the results of T5-CQR without QCS for query selection to address RQ1 where the performance of the two sentence selection models is explored. After that, the performance of T5-CQR with query selection is presented, which addresses RQ2 and whether using QCS for query selection improves system performance. Next we address RQ3 and explore how BERT-TC performs compared to T5-CQR. RQ4 is addressed where rank-based fusions of retrieved lists of passages are investigated. For RQ5, duoT5 is used on the systems to explore the benefits of multi-stage re-ranking. Finally, we analyze the performance of the best system at different turn depths in the conversation.

5.1. T5-CQR without query selection (RQ1)

First, we explore system performance without using clarity measures for query selection. As our internal baselines, we have the following auto systems: CQR-HC and CQR-RHC. We also experimented with NSP or QA for sentence selection for T5-CQR, resulting in four auto-canonical systems: CQR–HC–QA, CQR–HC–NSP, CQR-RHC-QA, and CQR-RHC-NSP. We also experimented with two versions of the T5 model for CQR, Base and Large. Results of all variants are reported in Table 4.

The results reveal that using NSP for sentence selection yielded better performance than QA by an average NDCG@3 of 3.3% for both base and large T5. Also, overall, sentence selection (alone) did not exhibit better performance to the auto systems. At best, using NSP yielded comparable performance with the auto systems only when using large T5. Expectedly, using the large T5 model boosted the performance relative to base T5. However, the computational cost of using large T5 should be considered; base T5 still significantly improves over simply using the raw turns and is more efficient than large T5. Finally, our auto baselines, CQR-HC and

[1] https://github.com/huggingface/transformers.

[2] https://github.com/castorini/anserini.

[3] https://github.com/castorini/pygaggle.

Table 4
Performance of auto and auto-canonical systems without query selection.

CQR	Retrieval	MonoT5 Re-Ranking	
	R@1000	MAP@1000	NDCG@3
Raw Turn	0.271	0.125	0.208
CAsT Best Auto	0.668	0.330	0.452
CAsT Best Auto-Canonical	0.724	0.363	0.494
(with Base T5)			
CQR-HC	0.547	0.286	0.444
CQR-RHC	0.565	0.295	0.442
CQR–HC–QA	0.523	0.273	0.411
CQR-RHC-QA	0.531	0.272	0.418
CQR–HC–NSP	0.546	0.280	0.432
CQR-RHC-NSP	0.546	0.274	0.424
(with Large T5)			
CQR-HC	0.588	0.309	0.480
CQR-RHC	0.578	0.300	0.463
CQR–HC–QA	0.584	0.305	0.463
CQR-RHC-QA	0.597	0.306	0.469
CQR–HC–NSP	0.604	0.307	0.480
CQR-RHC-NSP	0.614	0.314	**0.483**

CQR-RHC outperform the best CAsT auto system in NDCG@3 by margins of 6.2% and 2.4%, respectively.

5.2. T5-CQR with query selection (RQ2)

Table 5 presents the performance of our best auto-canonical systems from the earlier results (i.e., using NSP for sentence selection) but now using the proposed clarity measures for query selection. The results show that the proposed query selection method improved the overall NDCG@3 performance using both BM25-CL and IDF-CL clarity scores relative to their respective systems leveraging no query selection. The results of a paired *t*-test shows that the addition of the BM25-CL clarity score provided a statistically significant improvement over the system without query selection (CQR-HC); t(207) = -2.19, p = 0.029. Moreover, BM25-CL with CQR–HC–NSP yielded an NDCG@3 score of 0.506, which slightly outperforms the best CAst 2020 auto-canonical system, while IDF-CL exhibited comparable performance to that baseline. It is worth noting that CAsT 2020 auto-canonical baselines uses large T5 and a combination of dense and sparse retrieval. Our system achieves comparable results with only large T5 and a traditional sparse BM25 retrieval system with the application of QCS.

This shows that combining both sentence selection from canonical responses and query selection is effective and continuously considering the canonical responses in the context is not optimal. Also, IDF-CL is more computationally efficient than the other measures, as it does not require the issuance of an additional search query. Moreover, using a paired *t*-test, we found no statistically significant difference between the systems using BM25-CL and IDF-CL, giving an advantage to IDF-CL; t(207) = 1.93, p = 0.055. Finally, *n*BM25-CL surprisingly degraded performance compared to its non-normalized version, which indeed needs further investigation. The results of this experiment show the ability of QCS at creating a more selective system to improve performance without additional complexity.

Table 5
Auto-canonical results with QCS using large T5.

Query Selection	Retrieval	MonoT5 Re-Ranking	
	R@1000	MAP	NDCG@3
CAsT Best Auto-Canonical	0.724	0.363	0.494
CQR–HC–NSP	0.604	0.307	0.480
CQR-RHC-NSP	0.614	0.314	0.483
CQR–HC–NSP (BM25-CL)	0.636	0.331	**0.506**
CQR-RHC-NSP (BM25-CL)	0.621	0.320	0.491
CQR–HC–NSP (*n*BM25-CL)	0.583	0.306	0.471
CQR-RHC-NSP (*n*BM25-CL)	0.582	0.304	0.465
CQR–HC–NSP (IDF-CL)	0.628	0.325	0.495
CQR-RHC-NSP (IDF-CL)	0.624	0.324	0.493

5.3. Query reformulation (T5-CQR) vs. term classification (BERT-TC) (RQ3)

Performance of the proposed BERT-TC model is shown in Table 6. Again we will consider auto systems, denoted TC–HC–*I*, and auto-canonical systems, TC-*S*-*I*. The auto-canonical systems use both NSP and QA for sentence selection referenced by *S*. *I* refers to the amount of history included during term classification. "1st" uses only the first turn in the conversation as context. "ALL" uses all historical turns in the case of TC–HC–ALL, and all turns and response sentences for TC-*S*-ALL. TC–HC–PREV-*d* and TC-*S*-PREV-*d* refer to history clipped to *d* previous turns, or turns and responses, respectively.

As Table 6 shows, the best auto-system is TC–HC–PREV-4 while the best auto-canonical system is TC-NSP-PREV-5. Generally, NSP performed better than QA for sentence selection. Overall, the performance of the two systems is comparable, and the exclusion or inclusion of responses did not provide a major difference in performance. NDCG@3 of the systems vary depending on how much history is included in context. We can see peak performance at around turn depth 4 and 5 for auto and auto-canonical systems, respectively. Adding too much context introduces noise into the dialogue, as can be seen by looking at "ALL" included history for both system categories. For example, topic 85 turn 4 asks what license is needed to start a food truck business. The manually resolved turn is "What licenses and permits are needed for a food truck?". This turn was better resolved using TC–HC–PREV-4 as "What licenses and permits are needed? pimped food truck". However, the model using all historical turns TC–HC–ALL resolved it as "What licenses and permits are needed? lamborghini". It failed to resolve the main topic; "food truck" and introduced an off-topic keyword "lamborghini".

The model did resolve missing context when compared to the raw turn baseline. However, as expected, the larger T5-CQR model achieved better results. CQR–HC–NSP(BM25-CL) outperformed TC-NSP-PREV-5 by a margin of 27.5%. While BERT-TC is more computationally efficient compared to T5, both in training and execution, the balance between cost and performance should be considered depending on the system's application and available resources.

Table 6
Term classification results.

Included History	Retrival	MonoT5 Re-Ranking	
	R@1000	MAP	NDCG@3
Raw Turn	0.271	0.125	0.208
CAsT Best Auto	0.668	0.330	0.452
(Auto-System)			
TC–HC–1ST	0.518	0.247	0.389
TC–HC–ALL	0.518	0.246	0.391
TC–HC–PREV-3	0.518	0.249	0.391
TC–HC–PREV-4	0.518	0.248	**0.395**
TC–HC–PREV-5	0.518	0.244	0.386
(Auto-Canonical System)			
CAsT Best Auto-Canonical	0.724	0.363	0.494
TC-NSP-ALL	0.563	0.249	0.375
TC-NSP-PREV-3	0.518	0.246	0.390
TC-NSP-PREV-4	0.518	0.250	0.392
TC-NSP-PREV-5	0.518	0.249	**0.397**
TC-QA-ALL	0.518	0.232	0.378
TC-QA-PREV-3	0.518	0.247	0.388
TC-QA-PREV-4	0.518	0.244	0.384
TC-QA-PREV-5	0.518	0.247	0.392

5.4. Rank fusion (RQ4)

In order to further improve the performance and explore the effectiveness of QCS, rank fusion and multi-model fusion are applied. Table 7 starts with CQR–HC–NSP(BM25-CL) as the baseline system to compare with rank fusion systems, in addition to the two best CAsT runs.

To explore the effectiveness of BM25-CL as a query selection method, we compare how the system would perform if the retrieved passages were simply combined instead of using BM25-CL. CQRHC+NSP(Fusion) combines the passages retrieved from both query versions using our best auto and auto-canonical systems; CQR-HC and CQR-RHC-NSP. This system achieved an NDCG@3 of 0.505 after monoT5 re-ranking. Paired *t*-test analysis shows the performance of the two systems are not statistically different; t(207) = -0.29, p = 0.77. However, using BM25-CL is more efficient than CQR-HC+NSP(Fusion) since only one query is issued during retrieval instead of two. Fewer passages will also be passed on to the re-ranker reducing computational time and cost significantly.

The last row in Table 7 fuses three different lists of retrieved passages. It combines the top two CQR systems, CQR-HC and CQR-RHC-NSP, with the best performing BERT-TC system as multi-model fusion. As expected, fusing improved R@1000 to 0.705. This resulted in a higher achieved NDCG@3 score of 0.513. The improvement is due to the higher recall before re-ranking. However, the *t*-test results shows no significant difference between this system and CQR–HC–NSP(BM25-CL); t(207) = -1.64, p = 0.10. This shows again that using the proposed query selection method BM25-CL produced comparable results with a more computationally efficient system. This final result is an 3.8% improvement in NDCG@3 over the 2020 best CAsT submission. CAsT 2020 best auto-canonical uses large T5, dense and spares retrieval, and monoT5 re-ranking. By including QCS and multi-model fusion, we were able to outperform the baseline even with a less sophisticated retrieval system.

5.5. DuoT5 Re-ranking (RQ5)

As a final step to further improve NDCG@3, duoT5 re-ranking is applied to some of our best performing systems. Table 8 displays NDCG@3 score after the final duoT5 re-ranking stage. Since duoT5 is so computationally expensive, it is applied here selectively to re-rank only the top 100 passages.

We observe that the duoT5 re-ranking stage does indeed improve results over just using monoT5. After this final stage, NDCG@3 of CQR–HC–NSP(BM25-CL) is further improved to 0.524 and CQR-HC+NSP+TC(Fusion) to 0.537. This result is an 8.7% improvement over the 2020 best CAsT submission. However, paired *t*-test shows that the differences in CQR-HC+NSP+TC(Fusion) after duoT5 are not statistically significant compared to their monoT5 versions; t(207) = -1.84, p = 0.068. This shows that even though duoT5 did improve final NDCG@3, using monoT5 still achieves good results but with a much more computationally efficient system.

We include two internal baselines to compare the benefits of QCS and multi-model fusion with duoT5 re-ranking. The baselines are the CQR-HC auto system, and the CQR-RHC-NSP auto-canonical system. Both these internal baselines leverage no query selection or fusion. We can observe after duoT5 re-ranking, CQR-HC and CQR-RHC-NSP achieve an NDCG@3 of 0.499 and 0.494, respectively. The two baselines did not outperform the systems using QCS and multi-model fusion. BM25-CL and multi-model fusion score an NDCG@3 of 0.506 and 0.513, respectively, after only monoT5 re-ranking. This emphasizes the benefits of these solutions as compared to the much more computationally expensive duoT5 re-ranker.

Table 7
Rank fusion results compared with baselines.

	Retrival	MonoT5 Re-Ranking	
	R@1000	MAP	NDCG@3
CAsT Best Auto	0.668	0.330	0.452
CAsT Best Auto-Canonical	0.724	0.363	0.494
CQR–HC–NSP(BM25-CL)	0.636	0.331	0.506
CQR-HC + NSP(Fusion)	0.669	0.338	0.505
CQR-HC + NSP + TC(Fusion)	0.705	0.347	**0.513**

Table 8
DuoT5 re-ranked results.

	Retrieval	DuoT5 Re-Ranking	
	R@1000	MAP	NDCG@3
CAsT Best Auto	0.668	0.330	0.452
CAsT Best Auto-Canonical	0.724	0.363	0.494
CQR-HC	0.588	0.314	0.499
CQR-RHC-NSP	0.603	0.312	0.494
CQR–HC–NSP(BM25-CL)	0.636	0.338	0.524
CQR-HC+NSP(Fusion)	0.669	0.347	0.528
CQR-HC+NSP+TC(Fusion)	0.705	0.355	**0.537**

5.6. Depth analysis

Fig. 5 illustrates our systems' average performance at different depths in the conversations used for testing. Fig. 5a presents the distribution of the conversations over different sizes measured by the number of turns. It shows a relatively uniform distribution of conversations of sizes 1 to 8, with fewer sizes 9 and above. This indicates that average performance at turns 1–9 is more indicative and reliable. Examining the behavior at different depths, as depicted in Fig. 5b, shows that our system exhibit relatively stable performance in the middle turns (2nd to 9th), after an initial drop at the first turn, indicating that our method is consistently resolving queries. Better performance at deeper turns indicates that the system is better at interpreting context. Since later turns are relatively few, the shown performance is unstable but not indicative.

6. Discussion and implications

After analyzing the experimental results, it can be observed that for both T5-CQR and BERT-TC, always excluding or including responses did not yield a great improvement in performance. For RQ1, we explored two response sentence selection models. Of the two, BERT pre-trained for next sentence prediction proved to be better than T5 pre-trained for question answering. Existing trained models can efficiently be used to select response sentences without additional training costs. To answer RQ2, query clarity score was proposed to determine when to use responses for query reformulation. Using the BM25 score of the top passage proved to be an effective query selection method. QCS improved system performance by creating a more selective system with BM25-CL clarity score or the more efficient pre-retrieval score IDF-CL.

Answering RQ3 shows that T5-CQR does outperform BERT-TC for all system variants, but it is much more computationally expensive. Deciding which system to use can depend on training data and computational power availability. BERT-TC remains a valid solution, as it still restores context compared to the raw turn baseline. For RQ4, we compared rank fusion to the proposed query clarity score. Using query clarity score, turn responses might be included or excluded, yielding a more selective model. This improved overall NDCG@3 and produced a system comparable to the proposed rank fusion method while still being the more efficient approach. However, the multi-model fusion method performed better than all system variants. Finally, while duoT5 re-ranking exhibited a better performance, it was not significantly different from monoT5, which is more efficient. We found that even without duoT5, both QCS and multi-model fusion outperformed baselines, even with external baseline having a more powerful retrieval system. Internal baselines without these two new approaches could not beat their performance even with added duoT5 re-ranking.

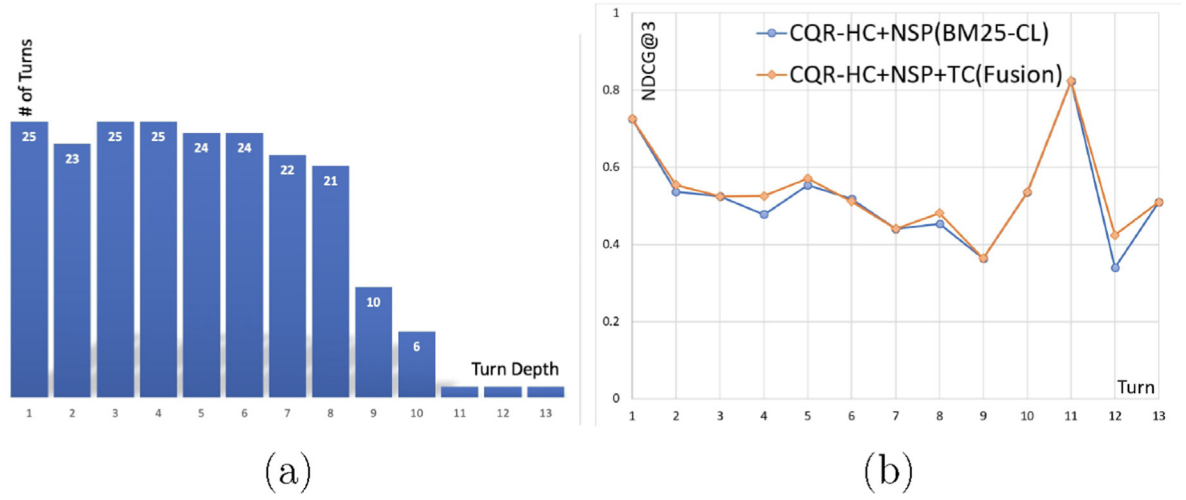


Fig. 5. Comparison of system at different turn depths.

To better investigate the effects of response sentence inclusion in T5-CQR, we explore NDCG@3 score of each turn. Out of the total 216 turns in CAsT 2020, 151 (69.9%) turns are reformulated into the same exact turn regardless of whether response sentences were included into T5-CQR or not. When comparing each reformulated turn with its counterpart reformulated with response sentence, we observe that 32 (14.8%) turns performed better with response sentence inclusion. Inversely, 31 (14.6%) turns performed better without the response sentence included into conversation history.

Table 9 displays some examples of turns that performed better with the response sentence included into T5-CQR input. The turns are labeled with "Turn ID" composed of their topic number and turn number in the CAsT 2020 dataset. As can be seen, these turns reformulated with response (sampled from the CQR-RHC-NSP system) more precisely name the turn subject when compared to the turns reformulated without response (sampled from the CQR-HC system). For example, turn 103_6 specifically names "Jerry Garcia" instead of the more vague "The Dead", which refers to the band as a whole not the person the user is asking about. This is because in these cases, the subjects being referred to in the turns were mentioned directly in the responses and not in the previous turns of the conversations.

On the other hand, sometimes the inclusion of the response sentence resulted in a degraded performance for some turns. In Table 10 we show some examples of such turns. The system added off-topic words extracted from the responses into the reformulated turns. For example, turn 101_4 included "Donald Trump", because he was named in the selected response sentence. The original raw turn "How old is he?" was referring to Melania's son, not Donald Trump. Similarly, in the other examples, T5-CQR model injected very specific keywords from the response sentences instead of simply considering the broader conversation topic.

To resolve this issue, QCS was employed to predict the better performing turn. Using BM25-CL, we were able to dynamically employ the two methods to achieve better average performance. For the above examples, QCS was able to select the better performing turn in all of them except for turn 102_8. For that turn, the QCS function incorrectly predicted "Can Social Security checks stopping coming via mail be fixed?" as the better performing query. Overall, QCS incorrectly predicted the better performing turn for a total of 19 queries (9.1%). This is still better than the 14.8% and 14.6% for retrievals without response and with response respectively. However, there is still room for further improvement. Table 11 shows some example turns where QCS using BM25-CL failed to predict the better performing query.

Implementing multi-model fusion to merge retrievals from trained T5 and BERT models proved to be the most effective system. This is fascinating behavior given that on its own BERT-TC was under-performing compared to T5-CQR. However, combining the two increased system's context-awareness and outperformed CAsT 2020 state-of-art. This introduces opportunities of exploring more combinations of pre-trained

Table 9
Turns with improved NDCG@3 after response inclusion.

Turn ID	T5-CQR without response (CQR-HC)	T5-CQR with response (CQR-RHC-NSP)
89_10	What are examples of **plants** that are predators?	What are examples of **apex** predators?
97_3	What are notable **games** between the Ravens and Steelers?	And notable moments in the **rivalry** of the Ravens and Steelers?
103_6	What was **The Dead's** relationship to the Airplane?	What was **Jerry Garcia's** relationship to the Airplane?
104_6	How did the Information Retrieval **researchers'** studies influence modern initiatives?	How did **Cyril Cleverdon's** experiments influence modern initiatives?

Table 10
Turns with improved NDCG@3 without response inclusion.

Turn ID	T5-CQR without response (CQR-HC)	T5-CQR with response (CQR-RHC-NSP)
93_3	Is there any financial support for the fee to open **a Burger King franchise**?	Is there any financial support for **the 4.5 percent royalty fee**?
98_8	How do you make the flour from **almonds**?	How do you make the flour in the **recipe**?
101_4	How old is **Melania Trump's son**?	How old is **Donald Trump**?
102_8	Can **social security** be fixed?	Can **Social Security checks stopping coming via mail** be fixed?

Table 11
Turns incorrectly selected by QCS.

Turn ID	T5-CQR without response (CQR-HC)	T5-CQR with response (CQR-RHC-NSP)
86_4	What was the impact of the **2002 games** on Salt Lake City?	What was the impact of the **2002 Olympics**?
87_9	How do **navels** compare with blood oranges?	How do **Hamlin variety of orange trees** compare with blood oranges?
91_8	What are the symptoms of **social network privacy addiction**?	What are the symptoms of **social network addiction**?
96_6	Tell me other fun things to do in Tokyo besides **Yakiniku**.	Tell me other fun things to do in Tokyo besides **eating at three star Michelin sushi restaurants**.

models with different architectures and training data in a novel way. Instead of merely having models compete against each other, we can create systems that leverage the strengths of different models to better represent user information need. This opens many possibilities for new model fusions to test in the future.

There are many other implications that can be concluded concerning CQR such as:

- Available pre-trained models such as BERT for NSP can be used as response sentence selection methods for CQR.
- Using seq2seq models such as T5-CQR generally outperforms smaller models such as BERT for term classification. However, appropriate training data is not always available for such models, so finding creative solution using smaller models is still valuable.
- Not all turns benefit form including responses. Using a simple query clarity score function is effective at determining whether to use a response for CQR during retrieval to produce a more selective system.
- Combining multiple lists of retrieved passages from two different fine-tuned models using multi-model fusion improves the system performance. It indicates that query variations from the two models better represent the user's information need.

7. Conclusion and future work

Ensuring context-awareness throughout a conversation is essential for any conversational search system. This paper proposed solving this problem using a T5-CQR model that incorporates previous responses into historical context selectively using a query clarity score function. Selecting what response to include is also important since long responses can introduce topic shifts and degrade performance. Two response selection models are explored, and BERT for next sentence prediction was the better performing one. We also explored restoring context as a term classification problem using BERT-TC. This model is more computationally efficient and requires less training data, however, T5-CQR outperformed it. Multi-model fusion was also used to combine multiple lists of retrieved passages from different pre-trained models and was found to improve recall thus improving NDCG@3; however, using query clarity score produced a comparable system with a more efficient method.

Some limitations and future work can be addressed as well. Some more exciting training solutions can be introduced to our BERT-TC model. Weak supervision solutions can be explored to incorporate both turns and responses into the training step. We leave as future work a more detailed examination of multi-model fusion techniques and model combinations. Different models can be tested using multi-model retrieval to explore how different models with varying architectures and training data add to user information need representation. We also aim to improve passage retrieval using more sophisticated techniques to increase recall@1000 and MAP, such as dense retrieval. Detecting topic shifts and applying more advanced turn classification techniques can also be explored by training a model on manually-labeled turns. Finally, response sentences can be extracted from retrieved passages to possibly create a more realistic scenario than using the canonical response provided by CAsT.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aliannejadi, M., Chakraborty, M., Ríssola, E. A., & Crestani, F. (2020). Harnessing evolution of multi-turn conversations for effective answer retrieval. In *Proceedings of the 2020 conference on human information interaction and retrieval* (pp. 33–42).

Almerekhi, H., Kwak, H., Salminen, J., Jansen, B. J., & Provoke. (2022). *Toxicity trigger detection in conversations from the top 100 subreddits, Data and Information Management,* Article 100019. https://doi.org/10.1016/j.dim.2022.100019. URL: https://www.sciencedirect.com/science/article/pii/S2543925122001176.

Asadi, N., & Lin, J. (2013). Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13* (pp. 997–1000). ACM Press. https://doi.org/10.1145/2484028.2484132. URL http://dl.acm.org/citation.cfm?doid=2484028.2484132.

Chang, C.-Y., Chen, H.-H., Chen, N., Chiang, W.-T., Lee, C.-H., Tseng, Y.-H., Tsai, M.-F., & Wang, C.-J. (2020). *Query expansion with semantic-based ellipsis reduction for conversational IR*. National Institute of Standards and Technology (NIST).

Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., Liang, P., & Zettlemoyer, L. (2018). Quac: Question answering in context. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2174–2184).

Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 299–306).

Cui, L., Wu, Y., Liu, S., Zhang, Y., & Zhou, M. (2020). Mutual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1406–1416).

Culpepper, J. S., Diaz, F., & Smucker, M. D. (2018). Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in lorne (swirl 2018). In *ACM SIGIR forum* (Vol. 52, pp. 34–90). New York, NY, USA: ACM.

Dalton, J., Xiong, C., & Callan, J. (2020a). *Trec cast 2019: The conversational assistance track overview* (p. 2019). National Institute of Standards and Technology (NIST).

Dalton, J., Xiong, C., & Callan, J. (2020b). *Cast 2020: The conversational assistance track overview* (p. 2020). National Institute of Standards and Technology (NIST).

Dehghani, M., Rothe, S., Alfonseca, E., & Fleury, P. (2017). Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 1747–1756).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. URL: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1*, 4171–4186 (Long and Short Papers) http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

Elgohary, A., Peskov, D., & Boyd-Graber, J. (2019). Can you unpack that? Learning to rewrite questions-in-context. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 5918–5924). https://doi.org/10.18653/v1/D19-1605. URL: https://www.aclweb.org/anthology/D19-1605.

Finegan-Dollak, C., & Radev, D. R. (2016). Sentence simplification, compression, and disaggregation for summarization of sophisticated documents. *Journal of the Association for Information Science and Technology, 67*, 2437–2453.

Fox, E. A., & Shaw, J. A. (1994). *Combination of multiple searches* (Vol. 243). NIST special publication SP.

Gao, J., Galley, M., & Li, L. (2018). Neural approaches to conversational ai. In *The 41st international ACM SIGIR conference on research and development in information retrieval, SIGIR '18* (pp. 1371–1374). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3209978.3210183.

Hou, Y., Liu, Y., Che, W., & Liu, T. (2018). Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1234–1245).

Hsu, D. F., & Taksa, I. (2005). Comparing rank and score combination methods for data fusion in information retrieval. *Information Retrieval, 8*, 449–480.

Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems* (pp. 2042–2050).

Kumar, V., & Callan, J. (2020). Making information seeking easier: An improved pipeline for conversational search. In *Proceedings of the 2020 conference on empirical methods in natural language processing: Findings* (pp. 3971–3980).

Lin, J., Nogueira, R., & Yates, A. (2021a). Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies, 14*, 1–325.

Lin, S.-C., Yang, J.-H., & Lin, J. (2020a). *Trec 2020 notebook: Cast track* (p. 2020). National Institute of Standards and Technology (NIST).

Lin, S.-C., Yang, J.-H., & Lin, J. (2020b). *Distilling dense representations for ranking using tightly-coupled teachers*, Article 11386. arXiv preprint arXiv:2010.

Lin, S.-C., Yang, J.-H., Nogueira, R., Tsai, M.-F., Wang, C.-J., & Lin, J. (2021b). Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting. *ACM Transactions on Information Systems, 39*, 1–29.

Liu, J. (2021). *Deconstructing search tasks in interactive information retrieval: A systematic review of task dimensions and predictors* (Vol. 58). Information Processing and Management, Article 102522. https://doi.org/10.1016/j.ipm.2021.102522. URL: https://www.sciencedirect.com/science/article/pii/S0306457321000315.

Lowe, R., Pow, N., Serban, I. V., & Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th annual meeting of the special interest group on discourse and dialogue* (pp. 285–294).

Lu, Z., & Li, H. (2013). A deep architecture for matching short texts. *Advances in Neural Information Processing Systems, 26*, 1367–1375.

Mehrotra, R., Awadallah, A. H., & Yilmaz, E. (2020). Special issue on learning from user interactions. *Information Retrieval Journal, 23*, 525–527.

Mele, I., Muntean, C. I., Nardini, F. M., Perego, R., Tonellotto, N., & Frieder, O. (2021). Adaptive utterance rewriting for conversational search, Information Processing and Management 58. URL: https://www.sciencedirect.com/science/article/pii/S0306457321001679 https://doi.org/10.1016/j.ipm.2021.102682.

Nogueira, R., & Cho, K. (2019). *Passage re-ranking with bert.* arXiv preprint arXiv:1901.04085.

Nogueira, R., Jiang, Z., Pradeep, R., & Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 708–718). Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.63. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.63.

Nogueira, R., Lin, J., & Epistemic, A. (2019b). *From doc2query to docTTTTTquery*. Online preprint.

Nogueira, R., Yang, W., Cho, K., & Lin, J. (2019a). *Multi-stage document ranking with bert*. arXiv preprint arXiv:1910.14424.

Onal, K. D., Zhang, Y., Altingovde, I. S., Rahman, M. M., Karagoz, P., Braylan, A., Dang, B., Chang, H.-L., Kim, H., McNamara, Q., et al. (2018). Neural information retrieval: At the end of the early years. *Information Retrieval Journal, 21*, 111–182.

Pradeep, R., Nogueira, R., & Lin, J. (2021). *The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models*. arXiv preprint arXiv:2101.05667.

Qu, C., Yang, L., Chen, C., Qiu, M., Croft, W. B., & Iyyer, M. (2020). Open-retrieval conversational question answering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 539–548).

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). *Exploring the limits of transfer learning with a unified text-to-text transformer*, Article 10683. arXiv preprint arXiv:1910.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*, 1–67.

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. URL:. In *Proceedings of the 56th annual meeting of the association for computational linguistics* (Vol. 2, pp. 784–789). Short Papers) http://arxiv.org/abs/1806.03822. arXiv:1806.03822.

Sa, N., & Yuan, X. (2020). Examining users' partial query modification patterns in voice search. *Journal of the Association for Information Science and Technology, 71*, 251–263.

Voskarides, N., Li, D., Ren, P., Kanoulas, E., & de Rijke, M. (2020). Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 921–930).

Vtyurina, A., Savenkov, D., Agichtein, E., & Clarke, C. L. A. (2017). Exploring conversational search with humans, assistants, and wizards. In *Proceedings of the 2017 CHI conference extended abstracts on human factors in computing systems - CHI EA '17* (pp. 2187–2193). ACM Press. https://doi.org/10.1145/3027063.3053175. URL: http://dl.acm.org/citation.cfm?doid=3027063.3053175.

Wang, H., Lu, Z., Li, H., & Chen, E. (2013). A dataset for research on short-text conversations. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 935–945).

Wicaksono, A. F., & Moffat, A. (2021). *Modeling search and session effectiveness, information processing and management 58*, Article 102601 https://doi.org/10.1016/j.ipm.2021.102601. URL: https://www.sciencedirect.com/science/article/pii/S0306457321000996.

Yuan, C., Zhou, W., Li, M., Lv, S., Zhu, F., Han, J., & Hu, S. (2019). Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 111–120). EMNLP-IJCNLP).

Zhang, Y., Ren, P., & de Rijke, M. (2021). *A taxonomy, data set, and benchmark for detecting and classifying malevolent dialogue responses*. Journal of the Association for Information Science and Technology.

Zheng, Y., Liu, Y., Fan, Z., Luo, C., Ai, Q., Zhang, M., & Ma, S. (2019). *Investigating weak supervision in deep ranking, Data and Information Management 3*. https://doi.org/10.2478/dim-2019-0010. URL: https://www.sciencedirect.com/science/article/pii/S2543925122000638.