

Asking Clarifying Questions in Open-Domain Information-Seeking Conversations

Mohammad Aliannejadi
Università della Svizzera italiana (USI)
mohammad.alian.nejadi@usi.ch

Hamed Zamani
University of Massachusetts Amherst
zamani@cs.umass.edu

Fabio Crestani
Università della Svizzera italiana (USI)
fabio.crestani@usi.ch

W. Bruce Croft
University of Massachusetts Amherst
croft@cs.umass.edu

ABSTRACT

Users often fail to formulate their complex information needs in a single query. As a consequence, they may need to scan multiple result pages or reformulate their queries, which may be a frustrating experience. Alternatively, systems can improve user satisfaction by proactively asking questions of the users to *clarify* their information needs. Asking clarifying questions is especially important in conversational systems since they can only return a limited number of (often only one) result(s).

In this paper, we formulate the task of asking clarifying questions in open-domain information-seeking conversational systems. To this end, we propose an offline evaluation methodology for the task and collect a dataset, called *Qulac*, through crowdsourcing. Our dataset is built on top of the TREC Web Track 2009-2012 data and consists of over 10K question-answer pairs for 198 TREC topics with 762 facets. Our experiments on an oracle model demonstrate that asking only one good question leads to over 170% retrieval performance improvement in terms of P@1, which clearly demonstrates the potential impact of the task. We further propose a retrieval framework consisting of three components: question retrieval, question selection, and document retrieval. In particular, our question selection model takes into account the original query and previous question-answer interactions while selecting the next question. Our model significantly outperforms competitive baselines. To foster research in this area, we have made Qulac publicly available.

1 INTRODUCTION

While searching on the Web, users often fail to formulate their complex information needs in a single query. As a consequence, they may need to scan multiple result pages or reformulate their queries. Alternatively, systems can decide to proactively ask questions to clarify users' intent before returning the result list [9, 33]. In other words, a system can assess the level of confidence in the results and decide whether to return the results or ask questions from the users to *clarify* their information need. The questions can be aimed to clarify ambiguous, faceted or incomplete queries [44]. Asking clarifying questions is especially important in conversational search systems for two reasons: (i) conversation is the most convenient way for natural language interactions and asking questions [22] and (ii) a conversational system can only return a limited number of results, thus being confident about the retrieval performance becomes even more important. Asking clarifying questions is a possible solution for improving this confidence. Figure 1 shows an example of such a conversation selected from our dataset. We see that both users, Alice and Robin, issue the same query, "dinosaur." Assuming that the system does not have access to any prior personal or contextual information, the conversation starts with the same clarifying question. The rest of the conversation, however, depends on the users' responses. In fact, the users' responses aid the system to get a better understanding of the underlying information need.

A possible workflow for an information system with clarifying questions is shown in Figure 2. As we can see, Alice initiates a conversation by submitting her query to the system. The system then retrieves a list of documents and estimates its confidence on the

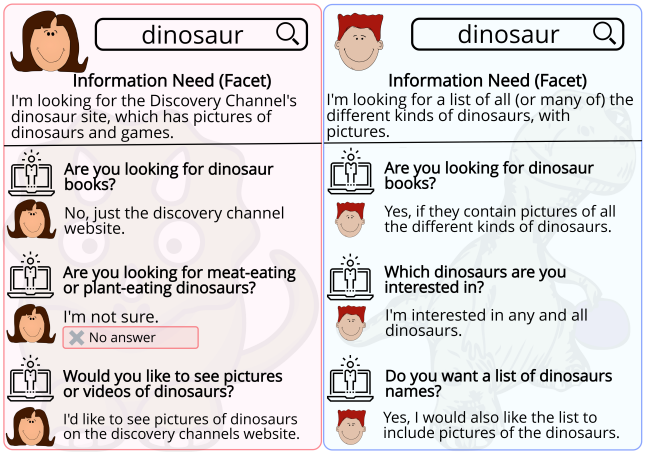


Figure 1: Example conversations with clarifying questions from our dataset, Qulac. As we see, both users, Alice and Robin, issue the same query ("dinosaur"), however, their actual information needs are completely different. With no prior knowledge, the system starts with the same clarifying question. Depending on the user's answers, the system selects the next questions in order to clarify the user's information need. The tag "No answer" shows that the asked question is not related to the information need.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '19, July 21–25, 2019, Paris, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331265

result list (i.e., "Present Results?"). If the system is not sufficiently confident to present the results to the user, it then starts the process of asking clarifying questions. As the first step, it generates a list of candidate questions related to Alice's query. Next, the system selects a question from the candidate question list and asks it from the user. Based on Alice's answer, the system retrieves new documents and repeats the process.

In this paper, we formulate the task of selecting and asking clarifying questions in open-domain information-seeking conversational systems. To this end, we propose an offline evaluation framework based on faceted and ambiguous queries and collect a novel dataset, called *Qulac*,[1] building on top of the TREC Web Track 2009-2012 collections. Qulac consists of over 10K question-answer pairs for 198 TREC topics consisting of 762 facets. Inspired from successful examples of crowdsourced collections [2, 4], we collected clarifying questions and their corresponding answers for every topic-facet pair via crowdsourcing. Our offline evaluation protocol enables further research on the topic of asking clarifying questions in a conversational search session, providing a benchmarking methodology to the community.

Our experiments on an oracle model show that asking only one good question leads to over 100% retrieval performance improvement. Moreover, the analysis of the oracle model provides important intuitions related to this task. For instance, we see that asking clarifying questions can improve the performance of shorter queries more. Also, clarifying questions exhibit a more significant effect on improving the performance of ambiguous queries, compared to faceted queries. We further propose a retrieval framework following the workflow of Figure 2, consisting of three main components as follows: (i) question retrieval; (ii) question selection; and (iii) document retrieval. The question selection model is a simple yet effective neural model that takes into account both users' queries and the conversation context. We compare the question retrieval and selection models with competitive term-matching and learning to rank (LTR) baselines, showing their ability to significantly outperform the baselines. Finally, to foster research in this area, we have made Qulac publicly available.[2]

2 RELATED WORK

While conversational search has roots in early Information Retrieval (IR) research, the recent advances in automatic voice recognition and conversational agents have created increasing interest in this area. One of the first works in conversational IR dates back to 1987 when Croft and Thompson [16] proposed I^3R that acted as an expert intermediary system, communicating with the user in a search session. A few years later Belkin et al. [7] characterized information-seeking strategies for conversational IR, offering users choices in a search session based on case-based reasoning. Since then researchers in the fields of IR and natural language processing (NLP) have studied various aspects of this problem. Early works focused on rule-base conversational systems [45, 47], while another line of research has investigated spoken language understanding approaches [1, 18, 29] for intelligent dialogue agents in the domain of flight [19] and train trip information [5]. The challenge was to understand the user's request and query a database

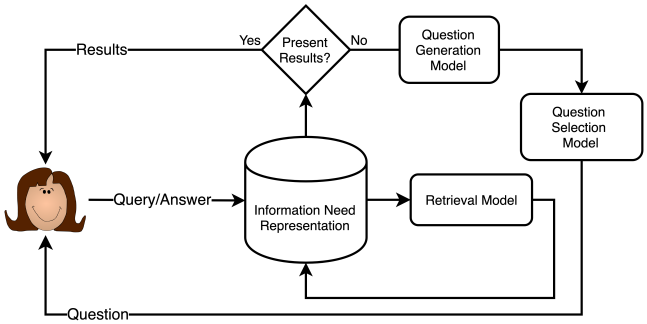


Figure 2: A workflow for asking clarifying questions in an open-domain conversational search system.

of flight or train schedule information accordingly. The recent advances of conversational agents have attracted research in various aspects of conversational information access [3, 6, 40, 49]. One line of research analyzes data to understand how users interact with voice-only systems [39]. Radlinski and Craswell [33] proposed a theoretical framework for conversational search highlighting the need for multi-turn interactions with users for narrowing down their specific information needs. Also, Trippas et al. [42] studied conversations of real users to identify the commonly-used interactions and inform a conversational search system design. Moreover, research on query suggestion is relevant to our work if we consider suggesting queries as a means of clarifying users' intent in a traditional IR setting [33]. Result diversification and personalizing is one of the key components for query suggestion [20], especially when applied to small-screen devices. In particular, Kato and Tanaka [21] found that presenting results for one facet and suggesting queries for other facets is more effective on such devices.

Research on clarifying questions has attracted considerable attention in the fields of NLP and IR. People have studied human-generated dialogues on question answering (QA) websites, analyzing the intent of each utterance [32] and, more specifically, clarifying questions [9]. Kiesel et al. [22] studied the impact of voice query clarification on user satisfaction and found that users like to be prompted for clarification. Much work has been done on interacting with users for recommendation. For instance, Christakopoulou et al. [12] designed a system that can interact with users to collect more detailed information about their preferences in venue recommendation. Also, Sun and Zhang [40] utilized a semi-structured user query with facet-value pairs to represent a conversation history and proposed a deep reinforcement learning framework to build a personalized conversational recommender system. Focusing on clarifying questions, Zhang et al. [53] automatically extracted facet-value pairs from product reviews and considered them as questions and answers. They proposed a multi-memory network to ask questions for improved e-commerce recommendation. Our work is distinguished from these studies by formulating the problem of asking clarifying questions in an open-domain information-seeking conversational setting where several challenges regarding extracting topic facets [23] are different from a recommendation setting.

In the field of NLP, researchers have worked on question ranking [34] and generation [35, 46] for conversation. These studies rely on large amount of data from industrial chatbots [31, 46], query logs [37], and QA websites [34, 35, 41]. For instance, Rao and Daumé

[1] Qulac, pronounced ku:lʌk, means *blizzard* and *wonderful* in Persian.
[2] Code and data are available at https://github.com/aliannejadi/qulac.

[34] proposed a neural model for question selection on a simulated dataset of clarifying questions and answers extracted from QA websites such as StackOverflow. Later, they proposed an adversarial training for generating clarifying questions for a given product description on Amazon [35]. Also, Wang et al. [46] studied the task of question generation for an industrial chatbot. Unlike these works, we study the task of asking clarification question in an IR setting where the user's request is in the form of short queries (vs. a long detailed post on StackOverflow) and the system should return a ranked list of documents.

3 PROBLEM STATEMENT

A key advantage of a conversational search system is its ability to interact with the user in the form question and answer. In particular, a conversational search system can proactively pose questions to the users to understand their actual information needs more accurately and improve its confidence in the search results. We illustrate the workflow of a conversational search system, focusing on asking clarifying questions.

As depicted in Figure 2, once the user submits a query to the system, the *Information Need Representation* module generates and passes their information need to the Retrieval Model, which returns a ranked list of documents. The system should then measure its confidence in the retrieved documents (i.e., *Present Results?* in Figure 2). In cases where the system is not sufficiently confident about the quality of the result list, it passes the query and the context (including the results list) to the *Question Generation Model* to generate a set of clarifying questions, followed by the *Question Selection Model* whose aim is to select one of the generated questions to be presented to the user. Next, the user answers the question and the same procedure repeats until a stopping criterion is met. Note that when the user answers a question, the complete session information is considered for selecting the next question. In some cases, a system can decide to present some results, followed by asking a question. For example, assume a user submits the query "sigir 2019" and the system responds "The deadline of SIGIR 2019 is Jan. 28. Would you like to know where it will be held?" As we can see, while the system is able to return an answer with high confidence, it can still ask further questions [50]. In this work, we do not study this scenario; however, one can investigate it for exploratory search.

3.1 A Facet-Based Offline Evaluation Protocol

The design of an offline evaluation protocol is challenging because conversation requires online interaction between a user and a system. Hence, an offline evaluation strategy requires human-generated answers to all possible questions that a system would ask, something that is impossible to achieve in an offline setting. To circumvent this problem, we substitute the Question Generation Model in Figure 2 with a large bank of questions, assuming that it consists of all possible questions in the collection. Although this assumption is not absolutely realistic, it reduces the complexity of the evaluation significantly as human-generated answers to a limited set of questions can be collected offline, facilitating offline evaluation.

In this work, we build our evaluation protocol on top of the TREC Web track's data. TREC has released 200 search topics, each of which being either "ambiguous" or "faceted."[3] Clarke et al. [13] defined these categories as follows: "... Ambiguous queries are those that have multiple distinct interpretations. ... On the other hand, facets reflect underspecified queries, with different aspects covered by the subtopics..." The TREC collection is originally designed to evaluate search result diversification. In contrast, here we build various conversation scenarios based on topic facets.

Formally, let $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ be the set of topics (queries) that initiates a conversation. Moreover, we define $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$ as the set of facets with $\mathbf{f}_i = \{f_1^i, f_2^i, \ldots, f_{m_i}^i\}$ defining different facets of t_i, where m_i denotes the number of facets for t_i. Further, let $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$ be the set of clarifying questions belonging to every topic, where $\mathbf{q}_i = \{q_1^i, q_2^i, \ldots, q_{z_i}^i\}$ consists of all clarifying questions that belong to t_i; z_i is the number of clarifying questions for t_i. Here, our aim is to provide the users' answers to all clarifying questions considering all topics and their corresponding facets. Therefore, let $\mathcal{A}(t, f, q) \rightarrow a$ define a function that returns answer a for a given topic t, facet f, and question q. Hence, to enable offline evaluation, $\mathcal{A}$ requires to return an answer for all possible values of t, f, and q. In this work, $\mathcal{T}$ and $\mathcal{F}$ are borrowed from the TREC Web track 2009-2012 data. $\mathcal{Q}$ is then collected via crowdsourcing and $\mathcal{A}(t, f, q)$ is also modeled by crowdsourcing (see Section 4). It is worth noting that we also borrow the relevance assessments of the TREC Web track, after breaking them down to the facet level. For instance, suppose the topic "dinosaur" has 10 relevant documents, 6 of which are labeled as relevant to the first facet, and 4 to the second facet. In Qulac, the topic "dinosaur" is broken into two topic-facet pairs together with their respective relevance judgments.

4 DATA COLLECTION

In this section, we explain how we collected Qulac (**Qu**estions for **la**ck of **c**larity), that is, to the best of our knowledge, the first dataset of clarifying questions in an IR setting. As we see in Figure 1, each topic is coupled with a facet. Therefore, the same question would receive a different answer based on the user's actual information need. We follow a four-step strategy to build Qulac. In the first step we define the topics and their corresponding facets. In the second step, we collect a number of candidate clarifying questions ($\mathcal{Q}$) for each query through crowdsourcing. Then, in the third step, we assess the relevance of the questions to each facet and collect new questions for those facets that require more specific questions. Finally, in the last step, we collect the answers for every query-facet-question triplet, modeling $\mathcal{A}$. In the following subsections, we elaborate on every step of our data collection procedure.

4.1 Topics and Facets

As we discussed earlier, the problem of asking clarifying questions is particularly interesting in cases where a query can be interpreted in various ways. An example is shown in Figure 1 where two different users issue the same query for different intents. Therefore, any data collection should contain an initial query and description of its facet, describing the user's information need. In other words, we define a target facet for each query. Faceted and ambiguous queries make an ideal case to study the effect of clarifying questions in a conversational search system for the following reasons: (i) the user

[3] In this work, we use the term "facet" to refer to the subtopics of both faceted and ambiguous topics.

information need is not clear from the query; (ii) multiple facets of the same query could satisfy the user's information need; (iii) asking clarifying questions related to any of the facets provide a high information gain. Therefore, we choose the TREC Web track's topics[4] [15] as the basis for Qulac. In other words, we take the topics of TREC Web track 09-12 as initial user queries. We then break each topic into its facets and assume that each facet describes the information need of a different user (i.e., it is a topic). As we see in Table 1, the average facet per topic is 3.85 ± 1.05. Therefore, the initial 198 TREC topics[5] leads to 762 topic-facet pairs in Qulac. Consequently, for each topic-facet pair, we take the relevance judgements associated with the respective facet.

4.2 Clarifying Questions

It is crucial to collect a set of reasonable questions that address multiple facets of every topic[6] while containing sufficient negative samples. This enables us to study the effect of retrieval models under the assumption of having a functional question generation model. Therefore, we asked human annotators to generate questions for a given query based on the results they observed on a commercial search engine as well as query auto-complete suggestions.

To collect clarifying questions, we designed a Human Intelligence Task (HIT) on Amazon Mechanical Turk.[7] We asked the workers to imagine themselves acting as a conversational agent such as Microsoft Cortana where an imaginary user had asked them about a topic. Then, we described the concept of facet to them, supporting it with multiple examples. Finally, we asked them to follow the steps below to figure out the facets of each query and generate questions accordingly:

(1) Enter the same query in a search engine of their choice and scan the results in the first three pages. Reading the title of the results as well as scanning the snippets would give them an idea of different facets of the query on the Web.
(2) For some difficult queries such as "toilet," scanning the results would not help in identifying the facets. Therefore, inspired by [8], we asked the workers to type the query in the search box of the search engine, and press the space key after typing the query. Most commercial search engines provide a list of query auto-complete suggestions. Interestingly, in most cases the suggested queries reflect various aspects of the same query.
(3) Finally, we asked them to generate six questions related to the query, aiming to address the facets that they had figured out.

We assigned two workers to each HIT, resulting in 12 questions per topic in the first round. In order to preserve language diversity of the questions, we limited each worker to a maximum of two HITs. HITs were available to workers residing in the U.S. with an approval rate of over 97%. After collecting the clarifying questions, in the next step, we explain how we verified them for quality assurance.

4.3 Question Verification and Addition

In this step, we aim to address two main concerns: (i) how good are the collected clarifying questions? (ii) are all facets addressed by at least one clarifying question? Given the high complexity of this step, we appointed two expert annotators for this task. We instructed the annotators to read all the collected questions of each topic, marking invalid and duplicate questions. Moreover, we asked them to match a question to a facet if the question was *relevant* to the facet. A question was considered relevant to a facet if its answer would address the facet. Finally, in order to make sure that all facets were covered by at least one question, we asked the annotators to generate an additional question for the facets that needed more specific questions. The outcome of this step is a set of verified clarifying questions, addressing all the facets in the collection.

Table 1: Statistics of Qulac.

# topics	198
# faceted topics	141
# ambiguous topics	57
# facets	762
Average facet per topic	3.85 ± 1.05
Median facet per topic	4
# informational facets	577
# navigational facets	185
# questions	2,639
# question-answer pairs	10,277
Average terms per question	9.49 ± 2.53
Average terms per answer	8.21 ± 4.42

4.4 Answers

After collecting and verifying the questions, we designed another HIT in which we collected answers to the questions for every facet. The HIT started with detailed instructions of the task, followed by several examples. The workers were provided with a topic and a facet description. Then we instructed them to assume that they had submitted the query with their actual information need being the given facet. Then they were required to write the answer to *one* clarifying question that was presented to them. To avoid the bias of other questions for the same facet, we included only one question in each HIT. If a question required information other than what workers were provided with, we instructed the workers to identify it with a "No answer" tag. Each worker was allowed to complete a maximum of 100 HITs to guarantee language diversity. Workers were based in the U.S. with an approval rate of 95% or greater.

Quality check. During the course of data collection, we performed regular quality checks on the collected answers. The checks were done manually on 10% of submissions per worker. In case we observed any invalid submissions among the sampled answers of one user, we then studied all the submissions of the same user. Invalid submissions were then removed from the collection and the worker was banned from the future HITs. Finally, we assigned all invalid answers to other workers to complete. Moreover, we employed basic behavioral check techniques in the design of the HIT. For example, we disabled copy/paste features of text inputs and tracked workers' keystrokes. This enabled us to detect and reject low-quality submissions.

5 SELECTING CLARIFYING QUESTIONS

In this section, we propose a conversational search system that is able to select and ask clarifying questions and rank documents based on the user's responses. The proposed system retrieves a set of questions for a given query from a large pool of questions,

[4] https://trec.nist.gov/data/webmain.html
[5] The official TREC relevance judgements cover 198 of the topics.
[6] Candidate clarifying questions should also address out-of-collection facets.
[7] http://www.mturk.com

containing all the questions in the collection. At the second stage, our proposed model, called NeuQS, aims to select the best question to be posed to the user based on the query and the conversation context. This problem is particularly challenging because the conversational interactions are in natural language, highly depending on the previous interactions between the user and the system (i.e., conversation context).

As mentioned earlier in Section 3, a user initiates the conversation by submitting a query. Then the system should decide whether to ask a clarifying question or present the results. At every stage of the conversation, the previous questions and answers exchanged between the user and the system are known to the model. Finally, the selected question and its corresponding answer should be incorporated in the document retrieval model to enhance the retrieval performance.

Formally, for a given topic t let $\mathbf{h} = \{(q_1, a_1), (q_2, a_2), \ldots, (q_{|\mathbf{h}|}, a_{|\mathbf{h}|})\}$ be the history of clarifying questions and their corresponding answers exchanged between the user and the system (i.e., context). Here, the ultimate goal is to predict q, that is the next question that the system should ask from the user. Moreover, let a be the user's answer to q. The answer a is unknown to the question selection model, however, the document retrieval model retrieves documents once the system receives the answer a. In the following, we describe the question retrieval model, followed by the question selection and the document retrieval models.

5.1 Question Retrieval Model

We now describe our **BERT**[8] **L**anguage R**e**present**a**tion based **Qu**estion **R**etrieval model, called BERT-LeaQuR. We aim to maximize the recall of the retrieved questions, retrieving all relevant clarifying questions to a given query in the top k questions. Retrieving all relevant questions from a large pool of questions is challenging, because questions are short and context-dependent. In other words, many questions depend on the conversation context and the query. Also, since conversation is in the form of natural language, term-matching models cannot effectively retrieve short questions. For instance, some relevant clarifying questions for the query "dinosaur" are: "Are you looking for a specific web page?" "Would you like to see some pictures?"

Yang et al. [51] showed that neural models outperform term-matching models for question retrieval. Inspired by their work, we learn a high-dimensional language representation for the query and the questions. Formally, BERT-LeaQuR estimates the probability $p(R = 1|t, q)$, where R is a binary random variable indicating whether the question q should be retrieved ($R = 1$) or not ($R = 0$). t and q denote the query (topic) and the candidate clarifying question, respectively. The question relevance probability in the BERT-LeaQuR model is estimated as follows:

$$p(R = 1|t, q) = \psi\Big(\phi_T(t), \phi_Q(q)\Big)\,, \tag{1}$$

where ϕ_T and ϕ_Q denote topic representation and question representation, respectively. ψ is the matching component that takes the aforementioned representations and produces a question retrieval score. There are various ways to implement any of these components.

[8]BERT: Bidirectional Encoder Representations from Transformers

We implement ϕ_T and ϕ_Q similarly using a function that maps a sequence of words to a d-dimensional representation ($V^s \rightarrow \mathbb{R}^d$). We use the BERT [17] model to learn these representation functions. BERT is a deep neural network with 12 layers that uses an attention-based network called Transformers [43]. We initialize the BERT parameters with the model that is pre-trained for the language modeling task on Wikipedia and fine-tune the parameters on Qulac with 3 epochs. BERT has recently outperformed state-of-the-art models in a number of language understanding and retrieval tasks [17, 27]. We particularly use BERT in our model to incorporate the knowledge from the vast amount of unlabeled data while learning the representation of queries and questions. In addition, BERT shows promising results in modeling short texts.

The component ψ is modeled using a fully-connected feed-forward network with the output dimensionality of 2. Rectified linear unit (ReLU) is employed as the activation function in the hidden layers, and a softmax function is applied on the output layer to compute the probability of each label (i.e., relevant or non-relevant). To train BERT-LeaQuR, we use a cross-entropy loss function.

5.2 Question Selection Model

In this section, we introduce a **Neu**ral **Q**uestion **S**election Model (NeuQS) which selects questions with a focus on maximizing the precision at the top of the ranked list. The main challenge in the question selection task is to predict whether a question has diverged from the query and conversation context. In cases where a user has given a negative answer(s) to previous question(s), the model needs to diverge from the history. In contrast, in cases where the answer to the previous question(s) is positive, questions on the same topic that ask for more details are preferred. For example, as we saw in Figure 1, when Robin answers the first question positively (i.e., being interested in dinosaur books), the second question tries to narrow down the information to a specific type of dinosaur.

NeuQS incorporates multiple sources of information. In particular, it learns from the similarity of a query, a question and the context as well as retrieval and performance prediction signals. In particular, NeuQS outputs a relevance score for a given query t, question q, and conversation context h. Formally, NeuQS can be defined as follows:

$$score = \gamma\Big(\phi_T(t), \phi_H(\mathbf{h}), \phi_Q(q), \eta(t, \mathbf{h}, q), \sigma(t, \mathbf{h}, q)\Big)\,, \tag{2}$$

where γ is a scoring function for a given query representation $\phi_T(t)$, context representation $\phi_H(\mathbf{h})$, question representation $\phi_Q(q)$, retrieval representation $\eta(t, \mathbf{h}, q)$, and query performance representation $\sigma(t, \mathbf{h}, q)$. Various strategies can be employed to model each of the components of NeuQS.

We model the components ϕ_T and ϕ_Q similarly to Section 5.1. Further, the context representation component ϕ_H is implemented as follows:

$$\phi_H(\mathbf{h}) = \frac{1}{|\mathbf{h}|}\sum_{i}^{|\mathbf{h}|}\phi_{QA}(q_i, a_i)\,, \tag{3}$$

where $\phi_{QA}(q, a)$ is an embedding function of a question q and answer a. Moreover, the retrieval representation $\eta(t, \mathbf{h}, q) \in \mathbb{R}^k$ is implemented by interpolating the retrieval score of the query, context and question (see Section 5.3) and the score of the top k retrieved documents is used. Finally, the query performance prediction (QPP) representation component $\sigma(t, \mathbf{h}, q) \in \mathbb{R}^k$ consists

of the performance prediction score of the ranked documents at different ranking positions (for a maximum of k ranked documents). We employed the σ QPP model for this component [28]. We take the representations from the [CLS] layer of the pre-trained uncased BERT-Base model (i.e., 12-layer, 768-hidden, 12-heads, 110M parameters). To model the function γ we concatenate and feed $\phi_T(t)$, $\phi_H(\mathbf{h})$, $\phi_Q(q)$, $\eta(t, \mathbf{h}, q)$, and $\sigma(t, \mathbf{h}, q)$ into a fully-connected feed-forward network with two hidden layers. We use ReLU as the activation function in the hidden layers of the network. We use a pointwise learning setting using a cross-entropy loss function.

5.3 Document Retrieval Model

Here, we describe the model that we use to retrieve documents given a query, conversation context, and current clarifying question as well as user's answer. We use the KL-divergence retrieval model [24] based on the language modeling framework [30] with Dirichlet prior smoothing [52] where we linearly interpolate two likelihood models: one based on the original query, and one based on the questions and their respective answers.

For every term w of the original query t, conversation context $\mathbf{h}$, the current question q, and answer a, the interpolated query probability is computed as follows:

$$p(w|t, \mathbf{h}, q, a) = \alpha \times p(w|\theta_t) + (1 - \alpha) \times p(w|\theta_{\mathbf{h},q,a}) \,, \quad (4)$$

where θ_t denotes the language model of the original query, and $\theta_{\mathbf{h},q,a}$ denotes the language model of all questions and answers that have been exchanged in the conversation. α determines the weight of the original query and is tuned on the development set.

Then, the score of document d is calculated as follows:

$$p(d|t, \mathbf{h}, q, a) = \sum_{w_k \in \tau} p(w_k|t, \mathbf{h}, q, a) \log(p(w_k|d) \,, \quad (5)$$

where τ is the set of all the terms present in the conversation. We use Dirichlet's smoothing for terms that do not appear in d. We use the document retrieval model for two purposes: (i) ranking documents after the user answers a clarifying question; (ii) ranking documents of a candidate question as part of the NeuQS (see Section 5.2). Hence, the model does not see the answer in the latter case.

6 EXPERIMENTS

6.1 Experimental Setup

Dataset. We evaluate BERT-LeaQuR and NeuQS on Qulac, following a 5-fold cross-validation. We follow two strategies to split the data, (i) **Qulac-T:** we split the train/validation/test sets based on topics. In this case, the model has not seen the test topics in the training data; (ii) **Qulac-F:** here we split the data based on their facets. Thus, the same test topic might appear in the training set, but with a different facet.

In order to study the effect of multi-turn conversations with clarifying questions, we expand Qulac to include multiple artificially generated conversation turns. To do so, for each instance, we consider all possible combinations of questions to be asked as the context of conversation. Take t_1 as an example where we select a new question after asking the user two questions. Assuming that t_1 has four questions, all possible combinations of questions in the conversation context would be: (q_1, q_2), (q_1, q_3), (q_1, q_4), (q_2, q_3), (q_2, q_4), (q_3, q_4). Notice that the set of candidate clarifying questions for each multi-turn example would be the ones that have not appeared in the context. The number of instances grows significantly as we enlarge the length of the conversation, leading to a total of 907,366 instances in the collection. At each turn of the conversation, we select the question from all candidate questions of the same topic and facet, having the same conversation history. In other words, they share the same context. Since the total number of unique conversational contexts is 75,200, a model should select questions for 75,200 contexts from all 907,366 candidate questions.

Table 2: Performance of question retrieval model. The superscript * denotes statistically significant differences compared to all the baselines ($p < 0.001$).

Method	MAP	Recall@10	Recall@20	Recall@30
QL	0.6714	0.5917	0.6946	0.7076
BM25	0.6715	0.5938	0.6848	0.7076
RM3	0.6858	0.5970	0.7091	0.7244
LambdaMART	0.7218	0.6220	0.7234	0.7336
RankNet	0.7304	0.6233	0.7314	0.7500
BERT-LeaQuR	**0.8349***	**0.6775***	**0.8310***	**0.8630***

Question retrieval evaluation metrics. We consider four metrics to evaluate the effectiveness of question retrieval models: mean average precision (MAP) and recall for the top 10, 20, and 30 retrieved questions (Recall@10, Recall@20, Recall@30). Our choice of measures is motivated by the importance of achieving high recall for this task.

Question selection evaluation metrics. Effectiveness is measured considering the performance of retrieval after adding the selected question to the retrieval model as well as the user answer. Five standard evaluation metrics are considered: mean reciprocal rank (MRR), precision of the top 1 retrieved document (P@1), and normalized discounted cumulative gain for the top 1, 5, and 20 retrieved documents (nDCG@1, nDCG@5, nDCG@20). We use the relevance assessments as they were released by TREC. However, we modify them in such a way to evaluate the performance with respect to every facet. For instance, if one topic consists of 4 facets it is then broken into 4 different topics each inheriting its own relevance judgements from the TREC assessments.

The choice of evaluation metrics is motivated by considering three different aspects of the task. We choose MRR to evaluate the effect of asking clarifying questions on ranking the first relevant document. We report P@1 and nDCG@1 to measure the performance for scenarios where the system is able to return only one result. This is often the case with voice-only conversational systems. Moreover, we report nDCG@5 and nDCG@20 as conventional ranking metrics to measure the impact of asking clarifying questions in a traditional Web search setting. Notice that nDCG@20 is the preferred evaluation metric for the ClueWeb collection due to the shallow pooling performed for relevance assessments [14, 26].

Statistical test. We determine statistically significant differences using the two-tailed paired t-test with Bonferroni correction at a 99.9% confidence interval ($p < 0.001$).

Compared methods. We compare the performance of our question retrieval and selection models with the following methods:

Table 3: Performance comparison with baselines. *WorstQuestion* and *BestQuestion* respectively determine the lower and upper bounds. The superscript * denotes statistically significant differences compared to all the baselines ($p < 0.001$).

Method	Qulac-T Dataset					Qulac-F Dataset				
	MRR	P@1	nDCG@1	nDCG@5	nDCG@20	MRR	P@1	nDCG@1	nDCG@5	nDCG@20
OriginalQuery	0.2715	0.1842	0.1381	0.1451	0.1470	0.2715	0.1842	0.1381	0.1451	0.1470
σ-QPP	0.3570	0.2548	0.1960	0.1938	0.1812	0.3570	0.2548	0.1960	0.1938	0.1812
LambdaMART	0.3558	0.2537	0.1945	0.1940	0.1796	0.3501	0.2478	0.1911	0.1896	0.1773
RankNet	0.3573	0.2562	0.1979	0.1943	0.1804	0.3568	0.2559	0.1986	0.1944	0.1809
NeuQS	**0.3625***	**0.2664***	**0.2064***	**0.2013***	**0.1862***	**0.3641***	**0.2682***	**0.2110***	**0.2018***	**0.1867***
WorstQuestion	0.2479	0.1451	0.1075	0.1402	0.1483	0.2479	0.1451	0.1075	0.1402	0.1483
BestQuestion	0.4673	0.3815	0.3031	0.2410	0.2077	0.4673	0.3815	0.3031	0.2410	0.2077

- Question retrieval:
  - *BM25, RM3, QL:* we index all the questions using Galago.[9] Then, for a given query we retrieve the documents using BM25 [38], RM3 [25], and QL [30] models.
  - *LambdaMART, RankNet:* for every query-question pair, we use the scores obtained by BM25, RM3, and QL as features to train LambdaMART [48] and RankNet [10] implemented in RankLib.[10] For every query, we consider all irrelevant questions as negative samples.
- Question selection:
  - *OriginalQuery* reports the performance of the document retrieval model only with the original query (Eq. (4) with $\alpha = 1$).
  - *σ-QPP:* we use a simple yet effective query performance predictor, σ [28] as an estimation of a question's quality. We calculate the σ predictor of the document retrieval model with the following input: original query, the context, and candidate questions. We then select the question with the highest σ value.
  - *LambdaMART, RankNet:* we consider the task of question selection as a ranking problem where a list of candidate questions should be ranked and the one with the highest rank is chosen. Therefore, we use LambdaMART [48] and RankNet [10] as two LTR baselines. The list of features are: (i) a flag determining if a question is open or not; (ii) a flag indicating if the answer to the last question in the context is yes or no; (iii) σ [28] performance predictor of the current question; (iv) the Kendall's τ correlation of the ranked list at 10 and 50 of the original query and the current question; (v) the Kendall's τ correlation of the ranked list at 20 and 50 of the current question and previous question-answer pairs in the context; (vi) Similarity of the current question and the query based on their BERT representations; (vii) Similarity of the current question and previous question-answer pairs in the context based on their BERT representations.
  - *BestQuestion, WorstQuestion:* in addition to all the baselines, we also report the retrieval performance when the worst and the best question is selected for an instance. BestQuestion (WorstQuestion) selects the candidate question for which the MRR value of the retrieval model is the maximum (minimum).

[9] https://sourceforge.net/p/lemur/galago/
[10] https://sourceforge.net/p/lemur/wiki/RankLib/

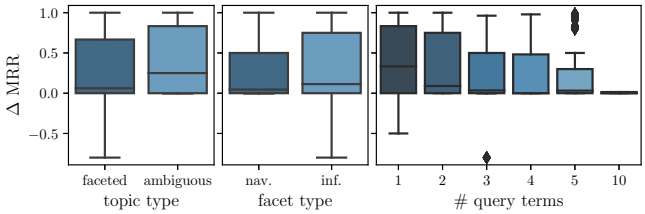


Figure 3: Impact of topic type, facet type, and query length on the performance of BestQuestion oracle model, compared to OriginalQuery.

Note that the retrieval scores are calculated knowing the selected question and its answer (i.e., oracle model). Our goal is to show the upper and lower bounds.

6.2 Results and Discussion

Question retrieval. Table 2 shows the results of question retrieval for all the topics. As we see, BERT-LeaQuR is able to outperform all baselines. It is worth noting that the model's performance gets better as the number of retrieved documents increases. This indicates that BERT-LeaQuR is able to capture the relevance of query and questions when they lack common terms. In fact, we see that all term-matching retrieval models such as BM25 are significantly outperformed in terms of all evaluation metrics.

Oracle question selection: performance. Here we study the performance of an oracle model, i.e. assuming that an oracle model is aware of the answers to the questions. The goal is to show to what extent clarifying questions can improve the performance of a retrieval system. As we see in the lower rows of Table 3 selecting best questions (BestQuestion model) helps the model to achieve substantial improvement, even in the case that the retrieval model is very simple. This shows the high potential gain of asking good clarifying questions on the performance of a conversational system. Particularly, we examine the relative improvement of the system after asking only one question and observe that BestQuestion achieves over 100% relative improvement in terms of different evaluation metrics (MRR: 0.2820 → 0.5677, P@1: 0.1933 → 0.4986, nDCG@1: 0.1460 → 0.3988, nDCG@5: 0.1503 → 0.2793, nDCG@20: 0.1520 → 0.2265). It is worth mentioning that we observe the highest relative improvements in terms of nDCG@1 (=173%) and P@1 (=158%), exhibiting a high potential impact on voice-only conversational systems.

Oracle question selection: impact of topic type and length. We analyze the performance of BestQuestion based on the number

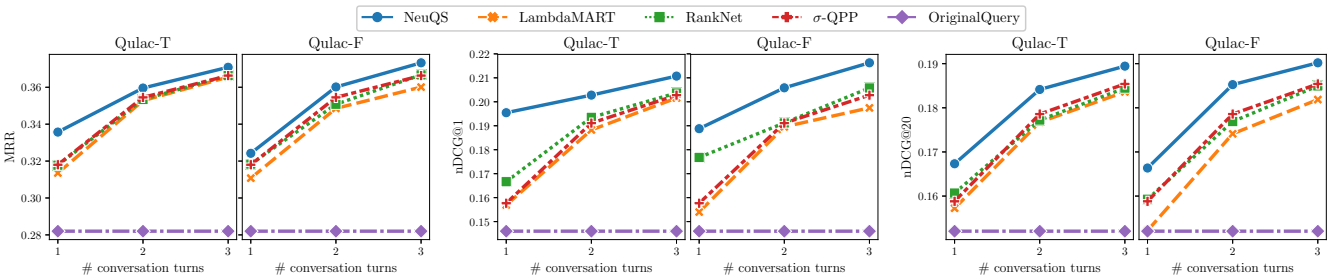


Figure 4: Performance comparison with the baselines for different number of conversation turns ($k \in \{1, 2, 3\}$).

of query terms and topic type. We see that the relative improvement of BestQuestion is negatively correlated with the number of query terms (Pearson's $r = -0.2$, $p \ll 0.001$), suggesting that shorter queries require clarification in more cases. Also, comparing the topic types (ambiguous vs. faceted), we see a significant difference in the relative improvement. The average ΔMRR for ambiguous topics is 0.3858, compared with the faceted topics with average ΔMRR of 0.2898. The difference was statistically significant (2-way ANOVA, $p \ll 0.001$).

Question selection. Table 3 presents the results of the document retrieval model taking into account a selected question together with its answer. We see that all models outperform OriginalQuery, confirming that asking clarifying questions is crucial in a conversation, leading to high performance gain. For instance, compared to OriginalQuery, a model as simple as σ-QPP achieves a 31% relative improvement in terms of MRR. Also, NeuQS consistently outperforms all the baselines in terms of all evaluation metrics on both data splits. All the improvements are statistically significant. Moreover, NeuQS achieves a remarkable improvement in terms of both P@1 and nDCG@1. These two evaluation metrics are particularly important for voice-only conversational systems where the system must return only one result to the user. The obtained improvements highlight the necessity and effectiveness of asking clarifying questions in a conversational search system, where they are perceived as natural means of interactions with users.

Impact of data splits. We compare the performance of models on both Qulac-T and Qulac-F data splits. We see that the LTR baselines perform worse on Qulac-F. Notice that the performance difference of LambdaMART among the splits is statistically significant in terms of all evaluation metrics ($p < 0.001$). RankNet, on the other hand, exhibits a more robust performance, i.e., the difference of its performance on the two splits is not statistically significant. Unlike the baselines, NeuQS exhibits a significant improvement in terms of all evaluation metrics on Qulac-F ($p < 0.05$), except for nDCG@5. This suggests that the baseline models are prone to overfitting on queries and conversations in the training data. As mentioned, Qulac-F's train and test sets may have some queries and questions in common, hurting models that are weak at generalization.

Impact of number of conversation turns. Figure 4 shows the performance of NeuQS as well as the baselines for different conversation turns. We evaluate different models at k turns ($k \in \{1, 2, 3\}$). We see that the performance of all models improves as the conversation advances to multiple turns. Also, we see that all the models consistently outperform the OriginalQuery baseline at different number of turns. Finally, we see that NeuQS exhibits robust performance, outperforming all the baselines at different turns.

Impact of clarifying questions on facets. We study the difference of MRR between NeuQS and OriginalQuery on all facets. Note that for every facet we average the performance of NeuQS at different conversation turns. Our goal is to see how many facets are impacted positively by asking clarifying questions. NeuQS is improves the effectiveness of retrieval by selecting relevant questions for a considerable number of facets on both data splits. In particular, the performance for 45% of the facets is improved by asking clarifying questions, whereas the performance for 19% is worse.

Case study: failure and success analysis. Finally, we analyze representative cases of failure and success of our proposed framework. We list three cases where selecting questions using NeuQS improves the retrieval performance, as well as three other examples in which the selected questions lead to decreased performance. ΔMRR reports the difference of the performance of NeuQS and OriginalQuery in terms of MRR. As we see, the first three examples show the selected questions that hurt the performance (i.e., ΔMRR < 0.0). The first row is an example where the user's response to the question is negative; however, the user provides additional information about their information need (i.e., facet). We see that even though the user has provided additional information, the performance drops. This is perhaps due to existence of no common terms between the additional information (i.e., "dog is too hot") and the facet (i.e., "*excessive heat* on dogs"). This is more evident when we compare this example with a successful answer: "No, I would like to know the effects of *excessive heat* on dogs." The second row of the table shows a case where the answer to the question is positive, but there is no common terms between the question and the facet. Again, the intuition here is that the retrieval model is not able to take advantage of additional information when it has no terms in common with relevant documents. The third row of the table shows another failure example where the selected question is not relevant to the facet and the user provides no additional information. This is a typical failure example where the system does not get any positive feedback, but could still use the negative feedback to improve the ranking. This can be done by diverging from the documents that are similar to the negative feedback.

As for the success examples, we have listed three types. The first example ("east ridge high school") is where the system is able to ask an open question. Open questions are very hard to formulate for open-domain information-seeking scenarios; however, it is more likely to get useful feedback from users in response to such questions. The fifth row shows an example of a positive feedback. The performance gain, in this case, is perhaps due to the existence of term "biography" in the question which would match with relevant documents. It is worth noting that the question and the query in this example have no common terms. This highlights the importance

Table 4: Failure and success examples of NeuQS. Failure and success are measured by the difference in performance of NeuQS and OriginalQuery in terms of MRR (ΔMRR).

Query	Facet Description	Selected Question	User's Answer	ΔMRR
dog heat	What is the effect of excessive heat on dogs?	Would you like to know how to care for your dog during heat?	No, I want to know what happens when a dog is too hot.	−0.86
sit and reach test	How is the sit and reach test properly done?	Do you want to know how to perform this test?	Yes, I do.	−0.75
alexian brothers hospital	Find Alexian Brothers hospitals.	Are you looking for our schedule of classes or events?	No, I don't need that.	−0.54
east ridge high school	Information about the sports program at East Ridge High School in Clermont, Florida	What information about East Ridge High School are you looking for?	I'm looking for information about their sports program.	+0.96
euclid	Find information on the Greek mathematician Euclid.	Do you want a biography?	Yes.	+0.93
rocky mountain news	Who are the sports reporters for the Rocky Mountain News?	Would you like to read recent news about the Rocky Mountain News?	No, I just want a list of the reporters who write the sports for the Rocky Mountain News.	+0.88

of employing a language-representation-based question retrieval model (e.g., BERT-LeaQuR) as opposed to term-matching IR models. The last example shows a case where the answer is negative, but the user is engaged in the conversation and provides additional information about the facet. We see that the answer contains keywords of the facet description (i.e., "reporters," "sports"), improving the score of relevant documents that contain those terms.

7 LIMITATIONS AND FUTURE WORK

Every data collection comes with some limitations. The same is valid for Qulac. First, the dataset was not collected from actual conversations. This decision was mainly due to the unbalanced workload of the two conversation participants. In our crowdsourcing HITs, the task of question generation required nearly 10 times more effort compared to the task of question answering. This makes it challenging and more expensive to pair two workers as participants of the same conversation. There are some examples of this approach in the literature [11, 36]; however, they address the task of reading comprehension, a task that is considerably simpler than identifying topic facets. A possible future direction is to provide a limited number of pre-generated questions (say 10) to the workers to select from, so that the complexity of the task would be significantly reduced.

Furthermore, Qulac is built for single-turn conversations (i.e., one question; one answer). Even though there are questions that can be asked after one another to form a multi-turn conversation, our data collection approach does not guarantee the existence of multi-turn conversations that involve the same participants. Also, we believe that the quality of generated clarifying questions highly depends on how well the selected commercial search engine is able to diversify the result list. We aimed to minimize this bias by asking workers to scan at least three pages of the result list. Also, the questions added by expert annotators guarantees the coverage of all facets (see Section 4.3). Finally, as we mentioned, faceted and ambiguous queries are good examples of topics that a conversational system needs to clarify; however, this task cannot be limited only to such queries. One can collect a similar data for exploratory search scenarios, where asking questions can potentially lead to more user engagement while doing exploratory search.

In this work, our main focus was on question selection. There are various directions that can be explored in the future. One interesting problem is to explore various strategies of improving the performance of the document retrieval model as new information is added to the model. Moreover, we assumed the number of conversation turns to be fixed. Another interesting future direction is to model the system's confidence at every stage of the conversation so that the model is able to decide when to stop asking questions and present the result(s).

8 CONCLUSIONS

In this work, we introduced the task of asking clarifying questions in open-domain information-seeking conversations. We proposed an evaluation methodology which enables offline evaluation of conversational systems with clarifying questions. Also, we constructed and released a new data collection called Qulac, consisting of 762 topic-facet pairs with over 10K question-answer pairs. We further presented a neural question selection model called NeuQS along with models on question and document retrieval. NeuQS was able to outperform the LTR baselines significantly. The experimental analysis provided many insights of the task. In particular, experiments on the oracle model demonstrated that asking only one good clarifying question leads to over 150% relative improvement in terms of P@1 and nDCG@1. Moreover, we observed that asking clarifying questions improves the model's performance for a substantial percentage of the facets. In some failure cases, we saw that a more effective document retrieval model can potentially improve the performance. Finally, we showed that, asking more clarifying questions leads to better results, once again confirming the effectiveness of asking clarifying questions in a conversational search system.

ACKNOWLEDGMENTS

This work was supported in part by the RelMobIR project of the Swiss National Science Foundation (SNSF), in part by the Center for Intelligent Information Retrieval, and in part by NSF grant IIS-1715095. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

REFERENCES

[1] Mohammad Aliannejadi, Masoud Kiaeeha, Shahram Khadivi, and Saeed Shiry Ghidary. 2014. Graph-Based Semi-Supervised Conditional Random Fields For Spoken Language Understanding Using Unaligned Data. In *ALTA*. 98–103.
[2] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2018. In Situ and Context-Aware Target Apps Selection for Unified Mobile Search. In *CIKM*. 1383–1392.
[3] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2018. Target Apps Selection: Towards a Unified Search Framework for Mobile Devices. In *SIGIR*. 215–224.
[4] Omar Alonso and Maria Stone. 2014. Building a Query Log via Crowdsourcing. In *SIGIR*. 939–942.
[5] Harald Aust, Martin Oerder, Frank Seide, and Volker Steinbiss. 1995. The Philips automatic train timetable information system. *Speech Communication* 17, 3-4 (1995), 249–262.
[6] Seyed Ali Bahrainian and Fabio Crestani. 2018. Augmentation of Human Memory: Anticipating Topics that Continue in the Next Meeting. In *CHIIR*. 150–159.
[7] Nicholas J Belkin, Colleen Cool, Adelheit Stein, and Ulrich Thiel. 1995. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert systems with applications* 9, 3 (1995), 379–395.
[8] Jan R. Benetka, Krisztian Balog, and Kjetil Nørvåg. 2017. Anticipating Information Needs Based on Check-in Activity. In *WSDM*. 41–50.
[9] Pavel Braslavski, Denis Savenkov, Eugene Agichtein, and Alina Dubatovka. 2017. What Do You Mean Exactly?: Analyzing Clarification Questions in CQA. In *CHIIR*. 345–348.
[10] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *ICML*. 89–96.
[11] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *EMNLP*. 2174–2184.
[12] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *KDD*. 815–824.
[13] Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. 2009. Overview of the TREC 2009 Web Track. In *TREC*.
[14] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Ellen M. Voorhees. 2011. Overview of the TREC 2011 Web Track. In *TREC*.
[15] Charles L. A. Clarke, Nick Craswell, and Ellen M. Voorhees. 2012. Overview of the TREC 2012 Web Track. In *TREC*.
[16] W. Bruce Croft and R. H. Thompson. 1987. I^3R: A new approach to the design of document retrieval systems. *JASIS* 38, 6 (1987), 389–404.
[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] Yulan He and Steve J. Young. 2005. Semantic processing using the Hidden Vector State model. *Computer Speech & Language* 19, 1 (2005), 85–106.
[19] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS Spoken Language Systems Pilot Corpus. In *HLT*. 96–101.
[20] Di Jiang, Kenneth Wai-Ting Leung, Lingxiao Yang, and Wilfred Ng. 2015. Query suggestion with diversification and personalization. *Knowl.-Based Syst.* 89 (2015), 553–568.
[21] Makoto P. Kato and Katsumi Tanaka. 2016. To Suggest, or Not to Suggest for Queries with Diverse Intents: Optimizing Search Result Presentation. In *WSDM*. 133–142.
[22] Johannes Kiesel, Arefeh Bahrami, Benno Stein, Avishek Anand, and Matthias Hagen. 2018. Toward Voice Query Clarification. In *SIGIR*. 1257–1260.
[23] Weize Kong and James Allan. 2013. Extracting query facets from search results. In *SIGIR*. 93–102.
[24] John Lafferty and Chengxiang Zhai. 2001. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *SIGIR*. 111–119.
[25] Victor Lavrenko and W. Bruce Croft. 2001. Relevance-Based Language Models. In *SIGIR*. 120–127.
[26] Xiaolu Lu, Alistair Moffat, and J. Shane Culpepper. 2016. The effect of pooling and evaluation depth on IR metrics. *Inf. Retr. Journal* 19, 4 (2016), 416–445.
[27] Harshith Padigela, Hamed Zamani, and W. Bruce Croft. 2019. Investigating the Successes and Failures of BERT for Passage Re-Ranking. *arXiv:1903.06902* (2019).
[28] Joaquín Pérez-Iglesias and Lourdes Araujo. 2010. Standard Deviation as a Query Hardness Estimator. In *SPIRE*. 207–212.
[29] Roberto Pieraccini, Evelyne Tzoukermann, Z. Gorelov, Jean-Luc Gauvain, Esther Levin, Chin-Hui Lee, and Jay Wilpon. 1992. A speech understanding system based on statistical representation of semantics. In *ICASSP*. 193–196.
[30] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *SIGIR*. 275–281.
[31] Minghui Qiu, Liu Yang, Feng Ji, Wei Zhou, Jun Huang, Haiqing Chen, W. Bruce Croft, and Wei Lin. 2018. Transfer Learning for Context-Aware Question Matching in Information-seeking Conversations in E-commerce. In *ACL (2)*. 208–213.
[32] Chen Qu, Liu Yang, W. Bruce Croft, Johanne R. Trippas, Yongfeng Zhang, and Minghui Qiu. 2018. Analyzing and Characterizing User Intent in Information-seeking Conversations. In *SIGIR*. 989–992.
[33] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *CHIIR*. 117–126.
[34] Sudha Rao and Hal Daumé. 2018. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information. In *ACL (1)*. 2736–2745.
[35] Sudha Rao and Hal Daumé III. 2019. Answer-based Adversarial Training for Generating Clarification Questions. *arXiv:1904.02281* (2019).
[36] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. CoQA: A Conversational Question Answering Challenge. *arXiv:1808.07042* (2018).
[37] Gary Ren, Xiaochuan Ni, Manish Malik, and Qifa Ke. 2018. Conversational Query Understanding Using Sequence to Sequence Modeling. In *WWW*. 1715–1724.
[38] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *TREC*. 109–126.
[39] Damiano Spina, Johanne R. Trippas, Lawrence Cavedon, and Mark Sanderson. 2017. Extracting audio summaries to support effective spoken document search. *JASIST* 68, 9 (2017), 2101–2115.
[40] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *SIGIR*. 235–244.
[41] Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. 2017. How to Make Context More Useful? An Empirical Study on Context-Aware Neural Conversational Models. In *ACL (2)*. 231–236.
[42] Johanne R. Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the Design of Spoken Conversational Search: Perspective Paper. In *CHIIR*. 32–41.
[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv:1706.03762* (2017).
[44] Alexandra Vtyurina, Denis Savenkov, Eugene Agichtein, and Charles L. A. Clarke. 2017. Exploring Conversational Search With Humans, Assistants, and Wizards. In *CHI Extended Abstracts*. 2187–2193.
[45] Marilyn A. Walker, Rebecca J. Passonneau, and Julie E. Boland. 2001. Quantitative and Qualitative Evaluation of Darpa Communicator Spoken Dialogue Systems. In *ACL*. 515–522.
[46] Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to Ask Questions in Open-domain Conversational Systems with Typed Decoders. In *ACL (1)*. 2193–2203.
[47] Jason D. Williams, Antoine Raux, Deepak Ramachandran, and Alan W. Black. 2013. The Dialog State Tracking Challenge. In *SIGDIAL*. 404–413.
[48] Qiang Wu, Christopher J. C. Burges, Krysta Marie Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Inf. Retr.* 13, 3 (2010), 254–270.
[49] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *SIGIR*. 55–64.
[50] Rui Yan, Dongyan Zhao, and Weinan E. 2017. Joint Learning of Response Ranking and Next Utterance Suggestion in Human-Computer Conversation System. In *SIGIR*. 685–694.
[51] Liu Yang, Hamed Zamani, Yongfeng Zhang, Jiafeng Guo, and W. Bruce Croft. 2017. Neural Matching Models for Question Retrieval and Next Question Prediction in Conversation. *arXiv:1707.05409* (2017).
[52] Chengxiang Zhai and John Lafferty. 2017. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. *SIGIR Forum* 51, 2 (2017), 268–276.
[53] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *CIKM*. 177–186.