Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search

Helia Hashemi
University of Massachusetts Amherst
hhashemi@cs.umass.edu

Hamed Zamani
Microsoft
hazamani@microsoft.com

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

ABSTRACT

Asking clarifying questions in response to ambiguous or faceted queries has been recognized as a useful technique for various information retrieval systems, especially conversational search systems with limited bandwidth interfaces. Analyzing and generating clarifying questions have been studied recently but the accurate utilization of user responses to clarifying questions has been relatively less explored. In this paper, we enrich the representations learned by Transformer networks using a novel attention mechanism from external information sources that weights each term in the conversation. We evaluate this Guided Transformer model in a conversational search scenario that includes clarifying questions. In our experiments, we use two separate external sources, including the top retrieved documents and a set of different possible clarifying questions for the query. We implement the proposed representation learning model for two downstream tasks in conversational search; document retrieval and next clarifying question selection. Our experiments use a public dataset for search clarification and demonstrate significant improvements compared to competitive baselines.

ACM Reference Format:
Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2020. Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3397271.3401061

1 INTRODUCTION

Conversational search has recently attracted much attention as an emerging information retrieval (IR) field. The ultimate goal of conversational search systems is to address user information needs through multi-turn natural language conversations. This goal is partially addressed in previous work with several simplifying assumptions. For example, the TREC Conversational Assistance Track (CAsT) in 2019 has focused on multi-turn conversational search, in which users submit multiple related search queries [16]. Similarly, conversational question answering based on a set of related questions about a given passage has been explored in the natural language processing (NLP) literature [7, 40, 44]. However, the existing settings are still far from the ideal *mixed-initiative* scenario, in which both user and system can take any permitted action at any time to perform a natural conversation. In other words, most existing work in conversational search assumes that users always ask a query and the system only responds with an answer or a ranked list of documents.

Recent conversational information seeking platforms, such as Macaw [59], provide support for multi-turn, multi-modal, and mixed-initiative interactions. There have been recent efforts to go beyond the "user asks, system responds" paradigm by asking clarifying questions from the users, including offline evaluation of search clarification [1], clarifying question generation for open-domain search queries [61], and preference elicitation in conversational recommender systems [8, 46, 64]. Past research in the area of search clarification has shown significant promise in asking clarifying questions. However, utilizing user responses to clarifying questions to improve the search performance has been relatively unstudied. In this paper, we propose a model that learns an accurate representation for a given user-system conversation. We focus on the conversations in which the user submits a query, and due to uncertainty about the query intent or the search quality, the system asks one or more clarifying questions to reveal the actual information need of the user. This is one of the many necessary steps that should be taken to achieve an ideal mixed-initiative conversational search system.

Motivated by previous research on improving query representation by employing other information sources, such as the top retrieved documents in pseudo-relevance feedback [2, 14, 27], we propose a neural network architecture that uses multiple information sources for learning accurate representations of user-system conversations. We extend the Transformer architecture [51] by proposing a novel attention mechanism. In fact, the sequence transformation in Transformer networks are guided by multiple external information sources in order to learn more accurate representations. Therefore, we call our network architecture *Guided Transformer* or GT. We train an end to end network based on the proposed architecture for two downstream target tasks: document retrieval and next clarifying question selection. In the first target task, the model takes a user-system conversation and scores documents based on their relevance to the user information need. On the other hand, the second task focuses on selecting the next clarifying question that would lead to higher search quality. For each target task, we also introduce an auxiliary task and train the model using a multi-task loss

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401061

function. The auxiliary task is identifying the actual query intent description for a given user-system conversation. For text representation, our model takes advantage of BERT [18], a state-of-the-art text representation model based on the Transformer architecture, modified by adding a “task embedding” vector to the BERT input to adjust the model for the multi-task setting.

In our experiments, we use two sets of information sources, the top retrieval documents (similar to pseudo-relevance feedback) and the pool of different clarifying questions for the submitted search query. The rational is that these sources may contain some information that helps the system better represent the user information needs. We evaluate our models using the public Qulac dataset and follow the offline evaluation methodology recently proposed by Aliannejadi et al. [1]. Our experiments demonstrate that the proposed model achieves over 29% relative improvement in terms of MRR compared to competitive baselines, including state-of-the-art pseudo-relevance feedback models and BERT, for the document retrieval task. We similarly observe statistically significant improvements in the next clarifying question selection task compared to strong baselines, including learning to rank models that incorporate both hand-crafted and neural features, including BERT scores.

In summary, the major contributions of this work include:

- Proposing a novel attention-based architecture, called Guided Transformer or GT, that learns attention weights from external information sources.
- Proposing a multi-task learning model based on GT for conversational search based on clarification. The multi-task learning model uses query intent description identification as an auxiliary task for training.
- Evaluating the proposed model on two downstream tasks in clarification-based conversations, namely document retrieval and next clarifying question selection.
- Outperforming state-of-the-art baseline models on both tasks with substantial improvements.

2 RELATED WORK

In this section, we review related work on conversational search, search clarification, and neural model enhancement using external resources.

2.1 Conversational Search and QA

Although conversational search has become an emerging topic in the IR community in recent years, it has roots in early work on interactive information retrieval, such as [11, 15, 33]. For instance, Cool et al. [11] extracted how users can have an effective interaction with a merit information seeking system. Later on, Croft and Thompson [15] introduced I^3R, the first IR model with a user modeling component for interactive IR tasks. Conversational system research in the form of natural language interaction started in the form of human-human interactions [11] or human-system interactions with rule-based models [52]. Some early work also focus on spoken conversations in a specific domain, such as travel [3, 22].

More recently, Radlinski and Craswell [42] introduced a theoretical framework and a set of potentially desirable features for a conversational information retrieval system. Trippas et al. [50] studied real user conversations and provided suggestions for building conversational systems based on human conversations. The recent improvements in neural models has made it possible to train conversation models for different applications, such as recommendation [64], user intent prediction [41], next user query prediction [58], and response ranking [57].

There is also a line of research in the NLP community with a focus on conversational question answering [44]. The task is to answer a question from a passage given a conversation history. In this paper, we focus on the conversations in which the system ask a clarifying question from the user, which is fundamentally different from the conversational QA literature.

2.2 Asking Clarifying Questions

Asking clarifying questions has attracted much attention in different domains and applications. To name a few, Braslavski et al. [6] studied user intents, and clarification in community question answering (CQA) websites. Trienes and Balog [49] also focused on detecting ambiguous CQA posts, which need further follow-up and clarification. There is other line of research related to machine reading comprehension (MRC) task, that given a passage, generating questions which point out missing information in the passage. Rao and Daumé III [43] proposed a reinforcement learning solution for clarifying question generation in a closed-domain setting. We highlight that most of the techniques in this area assume that a passage is given, and the model should point out the missing information. Hence, it is completely different form clarifying the user information needs in IR. Clarification has also studied in dialog systems and chat-bots [5, 17, 29], computer vision [31], and speech recognition [47]. However, since non of the above-mentioned systems are information seeking, their challenges are fundamentally different from challenges that the IR community faces in this area.

In the realm of IR, the user study done by Kiesel et al. [24] showed that clarifying questions do not cause user dissatisfaction, and in fact, they sometimes increase the satisfaction. Coden et al. [10] studied the task of clarification for entity disambiguation. However, the clarification format in their work was restricted to a “did you mean A or B?” template, which makes it non-practical for many open-domain search queries. More recently, Aliannejadi et al. [1] introduced an offline evaluation methodology and a benchmark for studying the task of clarification in information seeking conversational systems. They have also introduced a method for selecting the next clarifying question which is used in this paper as a baseline. Zamani et al. [61] proposed an approach based on weak supervision to generate clarifying questions for open-domain search queries. User interaction with clarifying questions has been later analyzed in [62].

A common application of clarification is in conversational recommendation systems, where the system asks about different attributes of the items to reveal the user preferences. For instance, Christakopoulou et al. [8] designed an interactive system for venue recommendation. Sun and Zhang [48] utilized facet-value pairs to represent a conversation history for conversational recommendation, and Zhang et al. [64] extracted facet-value pairs from product reviews automatically, and considered them as questions and answers. In this work, we focus on conversational search with open-domain queries which is different from preference elicitation in recommendation, however, the proposed solution can be potentially used for the preference elicitation tasks as well.

2.3 Enhancing Neural Models using External Information Sources

Improving the representations learned by neural models with the help of external resources has been explored in a wide range of tasks. Wu et al. [54] proposed a text matching model based on recurrent and convolution networks that has a knowledge acquisition gating function that uses a knowledge base for accurate text matching. Yang et al. [57] studied the use of community question answering data as external knowledge base for response ranking in information seeking conversations. They proposed a model based on convolutional neural networks on top the interaction matrix. More recently, Yang and Mitchell [55] exploited knowledge bases to improve LSTM based networks for machine reading comprehension tasks. Our work is different from prior work in multiple dimensions. From the neural network architecture, we extend Transformer by proposing a novel architecture for learning attention weights from external information sources. In addition, we do not use external knowledge bases in our experiments. For example, we use the top retrieved documents as one information source, which is similar to the pseudo-relevance feedback (PRF) methods [2, 14, 27, 60]. We showed that our method significantly outperforms state-of-the-art PRF methods.

3 METHODOLOGY

In this section, we first briefly review the basics of attention and self-attention in neural networks and then motivate the proposed solution. We further provide a detailed problem formulation, followed by the neural network architecture proposed in this work. We finish by describing the end to end multi-task training of the network.

3.1 Background

Attention is a mechanism for weighting representations learned in a neural network. The higher the weight, the higher the attention that the associated representation receives. For example, Guo et al. [19] used IDF as a signal for term importance in a neural ranking model. This can be seen as a form of attention. Later on, Yang et al. [56] used a question attention network to weight the query terms based on their importance. Attention can come from an external source, or can be computed based on the representations learned by the network.

The concept of attention has been around for a while. However, they really flourished when the self-attention mechanism has proven their effectiveness in a variety of NLP tasks [18, 28, 35, 36, 51]. Transformer networks [51] have successfully implemented self-attention and became one of the major breakthroughs in sequence modeling tasks in natural language processing. For instance, BERT [18] is designed based on multiple layers of Transformer encoders and pre-trained for the language modeling task. Self-attention is a specific attention mechanism in which the representations are learned based on the attention weights computed by the sequence tokens themselves. In other words, the sequence tokens decide which part of the sequence is important to emphasize and the representations are learned based on these weights. The original Transformer architecture was proposed for sequence-to-sequence tasks, such as machine translation. The model consists of the typical encoder-decoder architecture. Unlike previous work that often used convolution or recurrent networks for sequence modeling, Transformer networks solely rely on the attention mechanism. Each encoder layer, which is the most relevant to this work, consists of a self-attention layer followed by two point-wise feed forward layers. The self-attention layer in Transformer is computed based on three matrices, the query weight matrix W_Q, the key weight matrix W_K, and the value weight matrix W_V. Multiplying the input token representations to these matrices gives us three matrices Q, K, and V, respectively. Finally the self-attention layer is computed as:

$$Z = \text{softmax}(\frac{Q \times K^T}{\sqrt{d}})V \tag{1}$$

where d is the dimension of each vector. This equation basically represents a probability distribution over the sequence tokens using the $softmax$ operator applied to the query-key similarities. Then the representation of each token is computed based on the linear interpolation of values. To improve the performance of the self-attention layer, Transformer repeats this process multiple times with different key, query, and value weight matrices. This is called multi-headed self-attention mechanism. At the end, all Zs for different attention heads are concatenated as the output of multi-headed self-attention layer, and fed into the point-wise fully connected layer.

3.2 Motivation

In conversational search systems, users pursue their information needs through a natural language conversation with the system. Therefore, in case of uncertainty in query understanding or search quality, the system can ask the user a question to clarify the information need. A major challenge in asking clarifying questions is utilizing user responses to the questions for learning an accurate representation of the user information need.

We believe that the user-system conversation is not always sufficient for understanding the user information need, or even if it is sufficient for a human to understand the user intent, it is often difficult for the system, especially when it comes to reasoning and causation. For example, assume a user submits the query “migraines”, and the system asks the clarifying question “Are you looking for migraine symptoms?” and the user responds “no!”. Although negation has been studied for decades in the Boolean information retrieval and negative feedback literature [13, 37, 45, 53], it is still difficult for a system to learn an effective representation for the user information need.

In the above example, if external information sources cover different intents of query, the system can learn a representation similar to the intents other than “symptoms”. We present a general approach that can utilize multiple different information sources for better conversation representation. In our experiments, we use the top retrieved documents (similar to the pseudo-relevance feedback assumption [2, 14]) and all clarifying questions for the query as two information sources. Future work can employ user interaction data, such as click data, and past user interactions with the system as external sources.

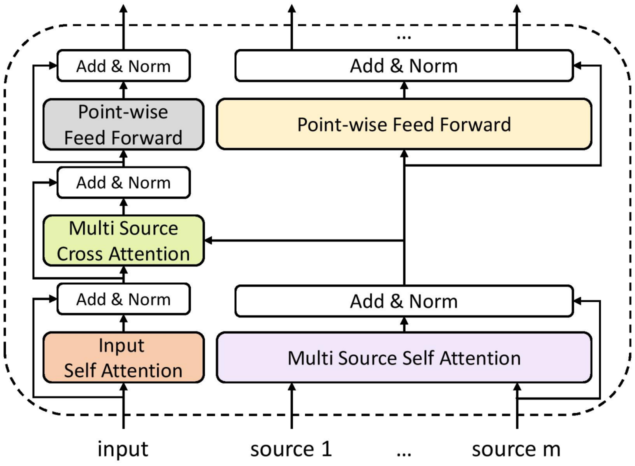

Figure 1: The architecture of each Guided Transformer layer on the input representations.

3.3 Problem Formulation

Let $Q = \{q_1, q_2, \cdots, q_n\}$ be the training query set, and $F_{q_i} = \{f_{1q_i}, f_{2q_i}, \cdots, f_{nq_i}\}$ denote the set of all facets associated with the query q_i.[1] In response to each query submitted by the user, a number of clarifying questions can be asked. Each conversation in this setting is in the form of $< q_i, c_1, a_1, c_2, a_2, \cdots, c_t, a_t >$, where c_i and a_i respectively denote the i^{th} clarifying question asked by the system and its answer responded by the user. The user response depends on the user's information need. The goal is to learn an accurate representation for any given conversation $< q_i, c_1, a_1, c_2, a_2, \cdots, c_t, a_t >$. The learned representations can be used for multiple downstream tasks. In this paper, we focus on (1) document retrieval and (2) next clarifying question selection.

Document Retrieval. In this task, each training instance consists of a user-system conversation with clarification, i.e., $< q_i$, c_1, a_1, $\cdots$, c_t, $a_t >$, a document from a large collection, and a relevance label.

Next Clarifying Question Selection. In this task, each training instance includes a user-system conversation with clarification (similar to above), a candidate clarifying question c and a label associated with c. The label is computed based on the search quality after asking c from the user.

Our model takes advantage of multiple information sources to better represent each user-system conversation. Let $S_{q_i} = \{s_{1q_i}, s_{2q_i}, \cdots, s_{mq_i}\}$ denote a set of external information sources for the query q_i. Each s_{jq_i} is a text. We later explain how we compute these information sources in our experiments. Note that the term "external" here does not necessary mean that the information source should come from an external resource, such as a knowledge graph. The term "external" refers to any useful information for better understanding of the user-system conversation that is not included in the conversation itself.

3.4 Guided Transformer

The architecture of each Guided Transformer (GT) layer is presented in Figure 1. The inputs of each GT layer is an input sequence (e.g., the user-system conversation) and m homogeneous textual information sources. Each source is a set of sequences. We first feed the input sequence to a self-attention layer, called "input self-attention". This is similar to the self-attention layer in Transformer (see Section 3.1). In the self-attention layer, the representation of each token in the input sequence is computed based on the weighted linear interpolation of all token representations, in which the weights are computed based on the similarity of the query and key vectors, normalized using the softmax operator. See Equation 1 for more details. We also apply a self-attention layer to the source representations. In other words, the "multi source self-attention" layer looks at all the sources and based on their similarity increases the attention on the tokens similar to those frequently mentioned in different source sequences. Based on the idea in residual networks [21], we add the input representations of each self-attention layer with its output and apply layer normalization [4]. This is also the standard technique in the Transformer architecture [51].

In the second stage, we apply attentions from multiple external sources to the input representations, i.e., the "multi source cross attention" layer. In this layer, we compute the impact of each token in external sources on each token in the input sequence. Let t_i and $t_j^{(k)}$ respectively denote the i^{th} input token and the j^{th} token in the k^{th} source ($1 \leq k \leq m$). The output encoding of t_i is computed as follows:

$$\vec{t_i} = \sum_{k=1}^{m} \sum_{j=1}^{|s_k|} p_{\text{ca}} \left(t_j^{(k)} | t_i \right) \vec{v}_{t_j^{(k)}} \qquad (2)$$

where s_k denotes the k^{th} external source and the vector $\vec{v}_{t_j^{(k)}}$ denotes the value vector learned for the token $t_j^{(k)}$. The probability p_{ca} indicates the cross-attention weight. Computing this probability is not straightforward. We cannot simply apply a softmax operator on top of key-query similarities, because the tokens come from different sources with different lengths. Therefore, if a token in a source has a very high cross-attention probability, it would dominate the attention weights, which is not desired. To address this issue, we re-calculate the above equation using the law of total probability and the Bayes rule as follows:

$$\vec{t_i} = \sum_{k=1}^{m} \sum_{j=1}^{|s_k|} p \left(t_j^{(k)} | s_k, t_i \right) p \left(s_k | t_i \right) \vec{v}_{t_j^{(k)}} \qquad (3)$$

In Equation 3, $p\left(t_j^{(k)} | s_k, t_i\right)$ denotes the attention weight of each token in source s_k to the token t_i, and $p\left(s_k | t_i\right)$ denotes the attention weight of the whole source s_k to the input token. This resolves the length issue, since $p\left(t_j^{(k)} | s_k, t_i\right)$ is normalized for all the tokens inside the source k, and thus no token in other sources can dominate the weight across all multi source tokens.

The cross-attention probability $p\left(t_j^{(k)} | s_k, t_i\right)$ is computed by multiplying the key vectors for the tokens in the source s_k to the query vector of the token t_i, normalized using the softmax operator. To compute the attention probability $p\left(s_k | t_i\right)$, we take the key

[1] The approaches are also applicable for ambiguous queries. Therefore, assume F_{q_i} contains all aspects of the query.

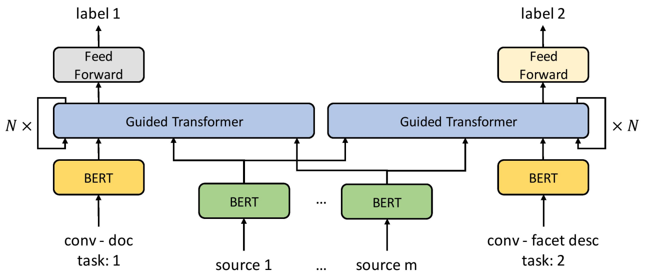

Figure 2: The high-level end to end architecture of the model trained using multi-task learning, where the first task is the target task (e.g., document ranking) and the second one is an auxiliary task that help the model identify the user information need from the user-system conversation. Same colors mean shared weights between the networks.

vector for the first token of the s_k sequence. The rational is that the representation of the start token in a sequence represents the whole sequence [18]. Therefore, the multi source cross-attention layer can be summarized as follows in a matrix form:

$$\sigma\left(\frac{Q' \times K'^{T}_{[\mathtt{CLS}]}}{\sqrt{d}}\right)\sum_{k=1}^{m}\sigma\left(\frac{Q \times K_k^T}{\sqrt{d}}\right)V_k \tag{4}$$

where K_k is the key matrix for the k^{th} external source, Q is the query matrix for the input sequence, $K'_{[\mathtt{CLS}]}$ is the key matrix for the first token of all sources, Q' is another query matrix for the input sequence (using two separate of query weight matrices), V_k is the value matrix for the k^{th} source, and d is the dimension of the vectors. The function σ is the softmax operator to transform real values in the $[-\infty, \infty]$ interval to a probabilistic space. Similar to the Transformer architecture, both self-attention and multi source cross-attention layers are designed based on the multi-headed attention, which is basically repeating the described process multiple times and concatenating the outputs. For the sake of space, we do not present the math for multi-headed attention and refer the reader to [51]. Finally, the multi source cross-attention is followed by residual and layer normalization. Therefore, multiple GT layers can be stacked for learning more complex representations.

As shown in Figure 1, the last step in the multi source attention layer is a point-wise feed forward network. This is similar to the last step of the Transformer architecture, and consists of two point-wise feed forward layers with a ReLU activation in the first one. This feed forward network is applied to all tokens. A final residual and layer normalization produces the output of each multi source attention layer. The input and output dimension of multi source attention layers are the same.

If there are different data with multiple instances as source signals, the model can be simply extended by adding more cross-attention layers, one per each data.

3.5 End to End Modeling and Training

The end to end architecture of the model is presented in Figure 2. As depicted in the figure, we use BERT for text representation. BERT [18] is a large-scale network based on Transformer which is pre-trained for a language modeling task. BERT has recently proven to be effective in a wide range of NLP and IR tasks, including question answering [18], passage re-ranking [32, 34], query performance prediction [20], and conversational QA [40]. The coloring in Figure 2 shows shared parameters. In other words, we use the same initial parameters for all the models, however, the parameters for BERTs with different colors are different and they are fine-tuned for accurate representation of their inputs. The output of external source representations and input representations are then fed to N Guided Transformer layers introduced above. N is a hyper-parameter. The representation for the start token of the input sequence (i.e., `[CLS]`) is then fed to a feed forward network with a single linear layer to predict the label. Note that for reusing BERT parameters in both tasks (the yellow components) we modified the BERT inputs, which is described later in this section.

As shown in the figure, we train our model using multi-task learning. The first task is the downstream target task (either document retrieval or next clarifying question selection) and the second task is an auxiliary task to help the model better learn the representation to identify the user information need. Note that we can train the model for three tasks (both target tasks and the auxiliary task), but due to GPU memory constraints we limit ourselves to two separate pairs of tasks.

Task 1: The Target Task (Document Retrieval or Next Clarifying Question Selection). For the first task, we concatenate all the interactions in the conversation and separate them with a `[SEP]` token. For the document retrieval task, we concatenate the obtained sequence with the document tokens. For the next clarifying question selection task, we concatenate the obtained sequence with the next clarifying question. Therefore, the input of BERT for the document retrieval task is `[CLS] query tokens [SEP] clarifying question tokens [SEP] user response tokens [SEP] document tokens [SEP]`.[2] Note that BERT has a maximum sequence length limitation of 512 tokens. Some documents in the collection are longer than this length limit. There exist a number of techniques to reduce the impact of this issue by feeding passages to the network and aggregating the passage scores. Since the focus of the work is on learning accurate conversation representations, we simply cut the documents that are longer than the sequence limit.

Since we re-use the BERT parameters for the second task, we modify the BERT input by adding a **task embedding** vector. In other words, the model learns a representation for each task (in the multi-task learning setting) and we simply add the token embedding, positional embedding, the segment embedding and the task embedding. The first three embeddings are used in the original BERT model.

Task 2: The Auxiliary Task (Intent Description Identification). The clarifying questions are asked to identify the user information need behind the query. For example, each faceted query has multiple facets and each facet shows a query intent. Therefore, we used the intent description (or facet description) in the data (see Section 3.3

[2] This example is only valid for a single clarifying question. The user-system conversation can contain more turns.

for more details). Similar to the first task we concatenate the user-system conversation with the intent (or facet) descriptions. The label for this task is to identify whether the given facet description describes the user information need or not. In other words, for each user information need, we take some negative samples for training the network and the goal is to distinguish the correct query intent description. This auxiliary task helps the network adjust the parameters by learning attentions that focus on the tokens related to the relevant facet descriptions. Note that the intent description is only available at the training time and at the inference time we put the second part of the model (i.e., task 2) aside.

Loss Function. Our loss function is a linear combination of the target loss function and the auxiliary loss function:

$$L = L_{\text{target}} + \alpha L_{\text{aux}} \tag{5}$$

where α is a hyper-parameter controlling the impact of the auxiliary loss function. Each of these loss functions is defined using cross entropy. For the task of document retrieval, the labels are binary (relevant vs. non-relevant), while, for the next clarifying question selection task the labels are real numbers in the [0, 1] interval (which are computed based on the retrieval performance if the question is selected). Note that this loss function for the document ranking is equivalent to pointwise learning to rank. Previous work that uses BERT for text retrieval shows that point-wise BERT re-rankers are as effective as the pair-wise BERT models [32].

4 EXPERIMENTS

In this section, we evaluate the proposed model and compare against state-of-the-art baselines. First, we introduce the data we use in our experiments and discuss our experimental setup and evaluation metrics. We finally report and discuss the results.

4.1 Data

To evaluate our model, we use the recently proposed dataset for asking clarifying questions in information seeking conversations, called **Qulac** [1]. Qulac was collected through crowdsourcing based on the topics in the TREC Web Track 2009-2012 [9]. Therefore, Qulac contains 200 topics (two of which are omitted, because they have no relevant document in the judgments). Each topic has recognized as either "ambiguous" or "faceted" and has been also used for evaluating search result diversification. After obtaining the topics and their facets from the TREC Web Track data, a number of clarifying questions has been collected through crowdsourcing for each topic. In the next step, the authors ran another crowdsourcing experiment to collect answers to each clarifying question based on a topic-facet pair. The relevance information was borrowed from the TREC Web Track. The statistics of this dataset is reported in Table 1. According to the table, the average facets per topic is 3.85 ± 1.05, and Qulac contains over 10k question-answer pairs.

The authors are not aware of any other IR dataset that contains clarifying questions.

4.2 Experimental Setup

We use the language modeling retrieval model [39] based on KL-divergence [26] with Dirichlet prior smoothing [63] for the initial retrieval of documents from the ClueWeb collection. The smoothing parameter μ was set to the average document length in the collection. For document indexing and retrieval, we use the open-source Galago search engine.[3] The spam documents were automatically identified and removed from the index using the Waterloo spam scorer[4] [12] with the threshold of 70%.

Table 1: Statistics of the Qulac dataset.

# topics	198
# faceted topics	141
# Ambiguous topics	57
# facets	762
# facet per topic	3.85 ± 1.05
# informational facets	577
# navigational facets	185
# clarifying questions	2,639
# question-answer pairs	10,277

We evaluate the models using 5-fold cross-validation. We split the data based on the topics to make sure that each topic is either in the training, validation, or test set. To improve reproducibility, we split the data based on the remainder (the modulo operation) of the topic ID to 5. Three folds are used for training, one fold for validation (hyper-parameter setting), and one fold for testing. After the hyper-parameters were selected based on the validation set, we evaluate the model with the seleted parameters on the test set. The same procedure was used for the proposed model and all the baselines.

We implemented our model using TensorFlow.[5] We optimize the network parameters using the Adam optimizer [25] with the initial learning rate of 3×10^6, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L_2 weight decay of 0.01, learning rate warm-up over the first 5000 steps, and linear decay of the learning rate. The dropout probability 0.1 is used in all hidden layers. The number of attention heads in multi-headed attentions is set to 8. The maximum sequence length for each document is set to 512 (i.e., the BERT maximum sequence length). We use the pre-trained BERT-base model (i.e., 12 layer, 768 dimensions, 12 heads, 110M parameters) in all the experiments.[6] The batch size was set to 4 due to memory constraints. The other hyper-parameters, including the parameter α (Equation 5) and the parameter N (the number of Guided Transformer layers), were selected based on the performance on the validation set.

4.3 Evaluation Metrics

Due to the nature of conversational search tasks, we focus on precision-oriented metrics to evaluate the models. We use mean reciprocal rank (MRR), and normalized discounted cumulative gain (nDCG) [23] with ranking cut-offs of @1, @5, and @20. We report the average performance across different conversations in the data. We identify statistical significant improvements using the paired t-test with Bonferroni correction at 95% and 99% confidence intervals (i.e., p-value less than 0.05 and 0.01, respectively).

[3] http://lemurproject.org/galago.php
[4] https://plg.uwaterloo.ca/~gvcormac/clueweb09spam/
[5] https://www.tensorflow.org/
[6] The pre-trained BERT models are available at https://github.com/google-research/bert

Table 2: The retrieval performance obtained by the baseline and the proposed models. In this experiment, only one clarifying questions has been asked. † and ‡ indicate statistically significant improvements compared to all the baselines with 95% and 99% confidence intervals, respectively. * indicates statistical significant improvements obtained by MTL compared to the STL training of the same model at 99% confidence interval.

	Method		MRR	nDCG@1	nDCG@5	nDCG@20
Baselines	QL		0.3187	0.2127	0.2120	0.1866
	RM3		0.3196	0.2189	0.2149	0.2176
	ERM		0.3291	0.2222	0.2191	0.2208
	SDM		0.3235	0.2267	0.2185	0.2182
	BERT		0.3527	0.2360	0.2267	0.2249
GT Network	**Source**	**Loss**				
	Docs	STL	$0.3710^{\ddagger}$	0.2388	$0.2309^{\dagger}$	0.2284
		MTL	$0.3826^{\ddagger *}$	$0.2407^{\dagger *}$	$0.2376^{\ddagger *}$	$0.2328^{\dagger}$
	CQs	STL	$0.4028^{\ddagger}$	$0.2638^{\ddagger}$	$0.2521^{\ddagger}$	$0.2491^{\ddagger}$
		MTL	$0.4259^{\ddagger *}$	$0.2742^{\ddagger *}$	$0.2626^{\ddagger *}$	$0.2543^{\ddagger *}$
	Docs +CQs	STL	$0.4338^{\ddagger}$	$0.2792^{\ddagger}$	$0.2714^{\ddagger}$	$0.2610^{\ddagger}$
		MTL	$\mathbf{0.4554}^{\ddagger *}$	$\mathbf{0.2939}^{\ddagger *}$	$\mathbf{0.2803}^{\ddagger *}$	$\mathbf{0.2697}^{\ddagger *}$

Table 3: Relative improvement achieved by GT with Docs+CQs and MTL compared to BERT for positive vs. negative user responses to clarifying question. * indicates statistical significant improvements at 99% confidence interval.

Answer	% MRR improvement	% nDCG@5 improvement
Positive	$24.56\%^{*}$	$19.06\%^{*}$
Negative	$31.20\%^{*}$	$25.84\%^{*}$

Table 4: Relative improvement achieved by GT with Docs+CQs and MTL compared to BERT for user different response length to the clarifying question. * indicates statistical significant improvements at 99% confidence interval.

Answer	% MRR improvement	% nDCG@5 improvement
Short	$33.12\%^{*}$	$26.65\%^{*}$
Medium	$30.83\%^{*}$	$23.93\%^{*}$
Long	$23.41\%^{*}$	$20.36\%^{*}$

4.4 Results and Discussion

4.4.1 Document Retrieval. In the first set of experiments, we focus on conversations with only one clarifying question. The main reason behind this is related to the way the Qulac dataset was created. The questions in Qulac were generated by people operating in a realistic setting. However, the multi-turn setting is not as realistic as the single turn. Therefore, we first focus on single clarification in our main experiment and later extend it to multi-turn as suggested in [1].

We compare GT with the following baselines:

- QL: The query likelihood retrieval model [39] with Dirichlet prior smoothing [63]. The smoothing parameter was set to the average document length in the collection. This baseline also provides the first retrieval list for the re-ranking models.
- RM3: A state-of-the-art variant of relevance models for pseudo-relevance feedback [27]. We selected the number of feedback documents from $\{5, 10, 15, 20, 30, 50\}$, the feedback term count from $\{10, 20, \cdots, 100\}$, and the feedback coefficient from $[0, 1]$ with the step size of 0.05.
- ERM: Embedding-based relevance models that extends RM3 by considering word embedding similarities [60]. The ERM parameter range is similar to RM3.
- SDM: The sequential dependence model of Metzler and Croft [30] that considers term dependencies in retrieval using Markov random fields. The weight of the unigram query component, the ordered window, and the unordered window were selected from $[0, 1]$ with the step size of 0.05, as the hyper-parameters of the model. We made sure that they sum to 1.
- BERT: A pre-trained BERT model [18] with a final feed forward network for label prediction. This is similar to the method proposed by Nogueira and Cho [32]. BERT-base was used in this experiment, which is similar to the setting in the proposed model. The loss function is cross entropy. The optimizer and the parameter ranges are the same as to the proposed method.

The hyper-parameters and training of all the models were done using 5-fold cross-validation, as described in Section 4.2. The same setting is used for the proposed methods. Note that the DMN-PRF model proposed by Yang et al. [57] was developed to use knowledge bases as external resources for response ranking in conversation. However, their model cannot accept long text, such as document level text, and thus we cannot use the model as a baseline. The machine reading comprehension models are all extracting answers from a passage and they require a passage as input, which is different from the setting in conversational search. Therefore, such models, e.g., [40], cannot be used as a baseline either. The BERT model has recently led to state-of-the-art performance in many IR tasks and is considered as a strong baseline for us.

We ran our model with different settings: single-task learning (STL) in which the only objective is the target task (i.e., document ranking) and multi-task learning (MTL) that optimizes two loss functions simultaneously. We also use the top 10 retrieved documents (i.e., Docs) and the top 10 clarifying questions (i.e., CQs) as external sources.

The results are reported in Table 2. According to the table, the proposed models significantly outperform all the baselines in nearly all cases. Using the clarifying questions as an external information source lead to a higher retrieval performance, compared to using the top retrieved documents. The reasons for this are two fold. First, the documents are often long and can be more than the 512 maximum sequence length. Therefore, the model cuts the documents and does not utilize all document tokens. Second, the top retrieved documents are not always relevant and non-relevant documents may introduce some noise. On the other hand, the clarifying questions are generated for the particular query and are all relevant. Note that although the clarifying questions in Qulac are human generated, recent methods are able to produce high quality clarifying questions for open-domain search queries. We refer the reader to [61] for more details on automatic generation of clarifying questions.

According to the results, the multi-task learning setting outperforms the single-task learning setting, in all cases. This shows the

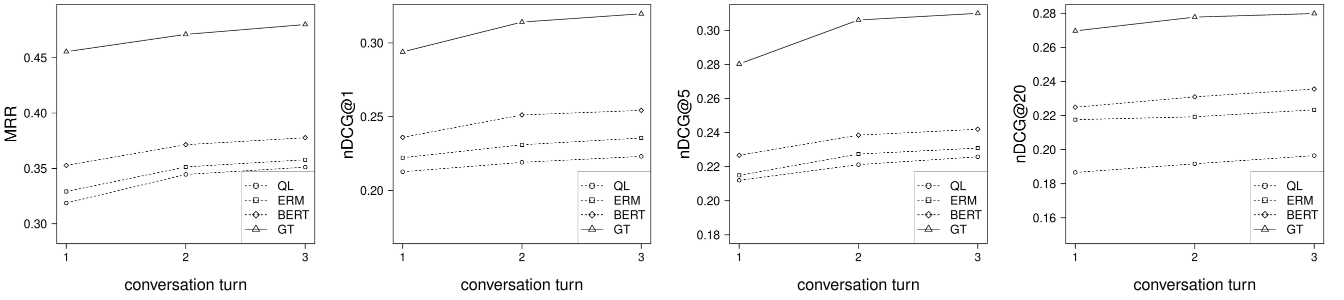

Figure 3: The performance of the GT (with Docs+CQs as source and MTL training) compared to the baselines for different conversation turns.

effectiveness of using an auxiliary task for identifying the correct query intent description. Note that the facet descriptions are only used for training.

Taking both clarifying questions and the top retrieved documents as two information sources leads to the best performance in document retrieval. This shows that these two sources provide complementary information and the model can effectively utilize this information to better represent the user-system conversations.

Figure 3 shows the performance curve as the conversation turn increases. Note that turn i means that i clarifying questions have been asked for each query. For the sake of visualization, we only plot QL, ERM, and BERT as the baselines and the GT with Docs+CQs as external source and MTL as the training setting. According to the plots, the performance of all models increases with the number of turns. However, the relative improvement in turn 3 compared to turn 2 is less than the improvements observed in turn 2 compared to turn 1. It is not practical to ask too many clarifying questions from users to answer their needs. The plots show it is also not helpful for the system. The proposed method substantially outperforms all the models in terms of all metrics, in all conversation turns.

To better understand the performance of the model, we report the results per different properties of user responses to clarifying questions. In this experiment, we only focus on a single clarifying question. We first identify yes/no questions (based on some simple regular expressions and the question response), and report the relative performance compared to BERT (our best baseline). For the sake of space, we only report the results for MRR and nDCG@5. The improvements for the other metrics also follow a similar behavior. For the proposed model, we focus on GT with the Docs+CQs source and MTL training. The results are presented in Table 3. According to the table, GT achieved higher improvements for negative responses. The reason is that for positive responses, the clarifying question already contains some important terms about the user information need, and thus using external information sources leads to smaller improvements. This is true for all metrics, two of which are reported in the table.

We extend our analysis to the response length as well. In this experiment, we divide the test conversations into three equal size buckets based on the user response length to clarifying questions. The results are reported in Table 4. According to the table, GT achieves higher improvements for shorter user responses. This is due to the information that is in the responses. In other words, it is easier for BERT to learn an effective representation of user information need if enough content and context are provided, however, for shorter responses, external information sources are more helpful. In both Tables 3 and 4, the improvements are statistically significant.

Table 5: Results for the next clarifying question selection task, up to 3 conversation turns. † and ‡ indicate statistically significant improvements compared to all the baselines with 95% and 99% confidence intervals, respectively. * indicates statistical significant improvements obtained by MTL compared to the STL training of the same model at 99% confidence interval.

Method	MRR	nDCG@1	nDCG@5	nDCG@20
OriginalQuery	0.2715	0.1381	0.1451	0.1470
σ-QPP	0.3570	0.1960	0.1938	0.1812
LambdaMART	0.3558	0.1945	0.1940	0.1796
RankNet	0.3573	0.1979	0.1943	0.1804
BERT-NeuQS	0.3625	0.2064	0.2013	0.1862
GT-Docs-STL	0.3784‡	0.2279‡	0.2107†	0.1890
GT-Docs-MTL	**0.3928**$^{\ddagger *}$	**0.2410**$^{\ddagger *}$	**0.2257**$^{\ddagger *}$	**0.1946**$^{\ddagger *}$
Oracle-Worst Question	0.2479	0.1075	0.1402	0.1483
Oracle-Best Question	0.4673	0.3031	0.2410	0.2077

4.4.2 *Next Clarifying Question Selection.*

In the second downstream task, we focus on next clarifying question selection. The task is to select a clarifying question given a user-system conversation. To be consistent with the results presented in [1], we use up to three conversation turns and report the average performance. The quality of models is computed based on the retrieval performance after asking the selected clarifying question from the user. Since the focus is on the next clarifying question selection, we use query likelihood as the follow up retrieval model, similar to the setting described in [1].

We compare our method against the following baselines:

- OriginalQuery: The retrieval performance for the original query without clarifying question. We use this baseline as a term of comparison to see how much improvement we obtain by asking a clarifying question.
- σ-QPP: We use a simple yet effective query performance predictor, σ [38], as an estimation of the question's quality. In other words, for a candidate clarifying question, we perform retrieval (without the answer) and estimate the performance using σ. The clarifying question that leads to the highest σ is selected.

Table 6: Some examples with a single clarification turn. ΔMRR is compared relative to the BERT performance.

Information need	Find sites for MGB car owners and enthusiasts.
Query	mgb
Clarifying question	are you looking for what mgb stands for?
User response	no
ΔMRR	+78%
Information need	What restrictions are there for checked baggage during air travel?
Query	air travel information
Clarifying question	where are you looking to travel to?
User response	doesn't matter i need information on checked baggage restrictions during air travel
ΔMRR	3%
Information need	What states levy a tax against tangible personal property?
Query	tangible personal property tax
Clarifying question	would you like to find out how much you owe?
User response	no i just want to know which states levy a tax against tangible personal property
ΔMRR	-21%

- LambdaMART: We use LambdaMART to re-rank clarifying questions based on a set of features, ranging from the query performance prediction to question template to BERT similarities. The exact definition of feature descriptions can be found in [1].
- RankNet: Another learning to rank model based on neural networks that uses the same features as LambdaMART.
- BERT-NeuQS: A model based on BERT used for clarifying question re-ranking proposed in [1].

The hyper-parameters of these baselines were set using cross validation, similar to the proposed method. The results are reported in Table 5. For the proposed method we only use the top retrieved documents as the external information source, since this is the most realistic setting for selecting clarifying questions. The proposed model significantly outperforms all baselines, including the recent BERT-NeuQS model. Consistent with the document retrieval experiments, the multi-task learning setting led to higher performance compared to STL. In addition, the achieved performance compared to the original query shows the benefit of asking clarification.

The table also contains the results for two oracle models, one that always selects the best question, which sets the upper-bound performance for this task on the Qulac data, and the one that always chooses the worst question (i.e., the lower-bound). The best possible performance is still much higher than the one achieved by the proposed solution and shows the potential for future improvement.

4.4.3 Case Study. We report three examples with one conversation turn for the document retrieval task in Table 6. The ΔMRR was computed relative to the BERT performance achieved by our best model. We report one win, tie, and loss example. In the first example, the user response is "no" and most baselines, including BERT, cannot learn a good representation from such conversation, however, the proposed model can use the top retrieved documents and the other clarifying questions to observe what are then other intents of the query and learn a better representation for the user information need. In the second example, the performance of the proposed model is similar to BERT, and in the last one, GT loses to BERT. The reason is that although the user response is negative, it contains some useful terms related to the information need, which makes it easier for the models to retrieve relevant documents. The proposed model, however, could not leverage external resources to learn an effective representation.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced Guided Transformer (GT) by extending the Transformer architecture. GT can utilize external information sources for learning more accurate representations of the input sequence. We implemented GT for conversational search tasks with clarifying questions. We introduced an end to end model that uses a modified BERT representations as input to GT and optimizes a multi-task objective, in which the first task is the target downstream task and the second one is an auxiliary task of discriminating the query intent description from other query intents for the input user-system conversation. We evaluated the proposed model using the recently proposed Qulac dataset for two downstream tasks in conversational search with clarification: (1) document retrieval and (2) next clarifying question selection. The experimental results suggested that the models implemented using GT substantially outperform state-of-the-art baselines in both tasks.

In the future, we intend to extend GT to a broad range of text understanding tasks using external sources. In the realm of conversational search, future work can focus on user past history as an external source for personalized conversational search. There are multiple different external sources that can use GT as an underlying framework for accurate representation learning for conversational search, such as query logs, clickthrough data, and knowledge bases. Most importantly, we are interested in extending this study to a fully mixed-initiative conversational search system, in which users can ask follow-up questions and at the same time the system can take different actions.

Acknowledgement. This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF 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 sponsor.

REFERENCES

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *SIGIR '19*. 475âĂŞ484.

[2] R. Attar and A. S. Fraenkel. 1977. Local Feedback in Full-Text Retrieval Systems. *J. ACM* 24, 3 (1977), 397âĂŞ417.

[3] H. Aust, M. Oerder, F. Seide, and Volker Steinbiss. 1994. Experience with the Philips automatic train timetable information system. 67 – 72.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. (2016). arXiv:stat.ML/1607.06450

[5] Marco Boni and Suresh Manandhar. 2005. Implementing clarification dialogue in open-domain question answering. *Natural Language Engineering* (2005), 343–361.

[6] Pavel Braslavski, Denis Savenkov, Eugene Agichtein, and Alina Dubatovka. 2017. What Do You Mean Exactly? Analyzing Clarification Questions in CQA. In *CHIIR '17*.

[7] 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 '18*. Brussels, Belgium, 2174–2184.

[8] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *KDD '16*. 815âĂŞ824.

[9] Charles L.A. Clarke, Nick Craswell, and Ian Soboroff. 2009. Overview of the TREC 2009 Web Track. In *TREC '09*.

[10] Anni Coden, Daniel Gruhl, Neal Lewis, and Pablo N. Mendes. 2015. Did you mean A or B? Supporting Clarification Dialog for Entity Disambiguation. In *SumPre '15*.

[11] Colleen Cool, Adelheit Stein, and Ulrich Thiel. 1970. Cases, Scripts, and Information-Seeking Strategies: On the Design of Interactive Information Retrieval Systems. *Expert Systems with Applications* 9 (02 1970), 379–395.

[12] Gordon V. Cormack, Mark D. Smucker, and Charles L. Clarke. 2011. Efficient and Effective Spam Filtering and Re-Ranking for Large Web Datasets. *Inf. Retr.* 14, 5 (2011), 441âĂŞ465.

[13] Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice* (1st ed.). Addison-Wesley Publishing Company, USA.

[14] W. B. Croft and D. J. Harper. 1988. *Using Probabilistic Models of Document Retrieval without Relevance Information.* 161âĂŞ171.

[15] W. B. Croft and R. H. Thompson. 1987. I3R: A New Approach to the Design of Document Retrieval Systems. *JASIS* (1987), 389âĂŞ404.

[16] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2019. TREC CAsT 2019: The Conversational Assistance Track Overview. In *TREC '19*.

[17] Marco De Boni and Suresh Manandhar. 2003. An Analysis of Clarification Dialogue for Question Answering. In *NAACL '03*. 48âĂŞ55.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL '19*.

[19] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-Hoc Retrieval. In *CIKM '16*. 55âĂŞ64.

[20] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2019. Performance Prediction for Non-Factoid Question Answering. In *ICTIR '19*. 55âĂŞ58.

[21] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR '16*. 770–778.

[22] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS Spoken Language Systems Pilot Corpus. In *Speech and Natural Language*.

[23] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422âĂŞ446.

[24] Johannes Kiesel, Arefeh Bahrami, Benno Stein, Avishek Anand, and Matthias Hagen. 2018. Toward Voice Query Clarification. In *SIGIR '18*. 1257âĂŞ1260.

[25] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. (2014). http://arxiv.org/abs/1412.6980

[26] John Lafferty and Chengxiang Zhai. 2001. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *SIGIR âĂŹ01*. 111âĂŞ119.

[27] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *SIGIR '01*. Association for Computing Machinery.

[28] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. *CoRR* (2017).

[29] Pont Lurcock, Peter Vlugter, and Alistair Knott. 2004. A framework for utterance disambiguation in dialogue. In *ALTA '04*. 101–108.

[30] Donald Metzler and W. Bruce Croft. 2005. A Markov Random Field Model for Term Dependencies. In *SIGIR âĂŹ05*. 472âĂŞ479.

[31] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating Natural Questions About an Image. In *ACL '16*. 1802–1813.

[32] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* (2019).

[33] Robert N. Oddy. 1977. Information Retrieval Through Man-Machine Dialogue. *Journal of Documentation* (1977), 1–14.

[34] Harshith Padigela, Hamed Zamani, and W. Bruce Croft. 2019. Investigating the Successes and Failures of BERT for Passage Re-Ranking. (2019). arXiv:cs.IR/1905.01758

[35] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. *CoRR* (2016).

[36] Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. *CoRR* (2017).

[37] Jaakko Peltonen, Jonathan Strahl, and Patrik Floréen. 2017. Negative Relevance Feedback for Exploratory Search with Visual Interactive Intent Modeling. In *IUI âĂŹ17*. 149âĂŞ159.

[38] Joaquín Pérez-Iglesias and Lourdes Araujo. 2010. Standard Deviation as a Query Hardness Estimator. In *SPIRE âĂŹ10*.

[39] J.M. Ponte and W.B. Croft. 1998. A language modeling approach to information retrieval. In *SIGIR '98*. 275–281.

[40] Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W. Bruce Croft, and Mohit Iyyer. 2019. Attentive History Selection for Conversational Question Answering. In *CIKM âĂŹ19*. 1391âĂŞ1400.

[41] Chen Qu, Yongfeng Zhang, Liu Yang, Johanne R. Trippas, W. Bruce Croft, and Minghui Qiu. 2019. In *CHIIR '19*.

[42] Filip Radlinski and Nick Craswell. A Theoretical Framework for Conversational Search. In *CHIIR '17*.

[43] Sudha Rao and Hal Daumé III. 2019. Answer-based Adversarial Training for Generating Clarification Questions. In *NAACL '19*.

[44] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* 7 (2019), 249–266.

[45] Gerard Salton, Edward A. Fox, and Harry Wu. 1982. *Extended Boolean Information Retrieval.* Technical Report. USA.

[46] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference Elicitation as an Optimization Problem. In *RecSys '18*. 172âĂŞ180.

[47] Svetlana Stoyanchev, Alex Liu, and Julia Hirschberg. 2014. Towards Natural Clarification Questions in Dialogue Systems. In *AISB '14*.

[48] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *SIGIR '18*. 235âĂŞ244.

[49] Jan Trienes and Krisztian Balog. 2019. Identifying Unclear Questions in Community Question Answering Websites. In *ECIR '19*.

[50] Johanne R. Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the Design of Spoken Conversational Search: Perspective Paper. In *CHIIR '18*. 32âĂŞ41.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *NeurIPS '17*.

[52] Marilyn A. Walker, Rebecca Passonneau, and Julie E. Boland. 2001. Quantitative and Qualitative Evaluation of Darpa Communicator Spoken Dialogue Systems. In *ACL '01*. Toulouse, France, 515–522.

[53] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. 2008. A Study of Methods for Negative Relevance Feedback. In *SIGIR '08*. 219âĂŞ226.

[54] Wei Wu, Yu Wu, Can Xu, and Zhoujun Li. Knowledge Enhanced Hybrid Neural Network for Text Matching. In *AAAI '18*.

[55] Bishan Yang and Tom Mitchell. 2019. Leveraging Knowledge Bases in LSTMs for Improving Machine Reading. (2019). arXiv:cs.CL/1902.09091

[56] Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. 2016. ANMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *CIKM âĂŹ16*. 287âĂŞ296.

[57] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-Seeking Conversation Systems. In *SIGIR âĂŹ18*.

[58] 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. In *NeuIR '17*.

[59] Hamed Zamani and Nick Craswell. 2020. Macaw: An Extensible Conversational Information Seeking Platform. In *SIGIR '20*.

[60] Hamed Zamani and W. Bruce Croft. 2016. Embedding-Based Query Language Models. In *ICTIR âĂŹ16*. 147âĂŞ156.

[61] Hamed Zamani, Susan T. Dumais, Nick Craswell, Paul N. Bennett, and Gord Lueck. 2020. Generating Clarifying Questions for Information Retrieval. In *WWW '20*. 418–428.

[62] Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N. Bennett, Nick Craswell, and Susan T. Dumais. 2020. Analyzing and Learning from User Interactions with Search Clarification. In *SIGIR '20*.

[63] Chengxiang Zhai and John Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *SIGIR âĂŹ01*. 334âĂŞ342.

[64] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *CIKM '18*. 177âĂŞ186.