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:

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].

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.
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’s 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 \( l^3 R \), 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 [1] 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 conversational 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 Daume 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
We finish by describing the end to end multi-task training of the model. Which part of the sequence is important to emphasize and the attention is a mechanism for weighting representations learned in the source. Or can be computed based on the representations learned from the sequence tokens themselves. In other words, the sequence tokens decide the representation of each token is computed based on the linear combination of the representations to these matrices gives us three matrices, the query weight matrix $Q$, the key weight matrix $K$, and the value weight matrix $V$, respectively. Finally the self-attention layer is computed as:

$$Z = \text{softmax}(QK^T)W$$

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 $Z$s 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 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 $Q, K, V$ respectively. Finally the self-attention layer is computed as:

$$Z = \text{softmax}(QK^TW)W$$

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 $Z$s 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.
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 $S_{q_i} = \{s_{1q_i}, s_{2q_i}, \cdots, s_{m_{q_i}}\}$ denote a set of external information sources for the query $q_i$. Each $s_{j_{q_i}}$ 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.

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

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:

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

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:

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

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 $s_k$, 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
which is basically repeating the described process multiple times where $K$ whole sequence \[18\]. Therefore, the multi source cross-attention layers, one per each data.

As depicted in the figure, we use BERT for text representation. The end to end architecture of the model is presented in Figure 2.

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. The multi source cross-attention is followed by $\sigma\left(\frac{Q' \times K_T^{T}}{\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, $K'_T$ is the key matrix for the first token of all sources, $Q'$ is another query matrix for the input sequence (using two separate pairs 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 $\sigma$ 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 \[40\].

### 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].

When training for two tasks (both target tasks and the auxiliary task), but due to GPU memory constraints we limit ourselves to two separate pairs of tasks.

### 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 facet 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.

---

\[\text{This example is only valid for a single clarifying question. The user-system conversation can contain more turns.}\]
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:

$$L = L_{\text{target}} + \alpha L_{\text{aux}}$$  \hspace{1cm} (5)

where \( \alpha \) 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 recog-
nized as either “ambiguous” or “faceted” and has been also used
for evaluating search result diversification. After obtaining the top-
ics 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 \( \mu \) was set to the average document length in the collec-
tion. 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%.

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 selected 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 \(3 \times 10^6\), \(\beta_1 = 0.9, \beta_2 = 0.999, L_2 \) 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 \( \alpha \) (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>
<thead>
<tr>
<th>Table 1: Statistics of the Qulac dataset.</th>
</tr>
</thead>
<tbody>
<tr>
<td># topics</td>
</tr>
<tr>
<td># faceted topics</td>
</tr>
<tr>
<td># Ambiguous topics</td>
</tr>
<tr>
<td># facets</td>
</tr>
<tr>
<td># facet per topic</td>
</tr>
<tr>
<td># informational facets</td>
</tr>
<tr>
<td># navigational facets</td>
</tr>
<tr>
<td># clarifying questions</td>
</tr>
<tr>
<td># question-answer pairs</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Baselines</th>
<th>Method</th>
<th>MRR</th>
<th>nDCG@1</th>
<th>nDCG@5</th>
<th>nDCG@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>QL</td>
<td>STL</td>
<td>0.3187</td>
<td>0.2127</td>
<td>0.2120</td>
<td>0.1866</td>
</tr>
<tr>
<td></td>
<td>RM3</td>
<td>0.3196</td>
<td>0.2189</td>
<td>0.2149</td>
<td>0.2176</td>
</tr>
<tr>
<td></td>
<td>ERM</td>
<td>0.3291</td>
<td>0.2222</td>
<td>0.2191</td>
<td>0.2208</td>
</tr>
<tr>
<td></td>
<td>SDM</td>
<td>0.3235</td>
<td>0.2267</td>
<td>0.2185</td>
<td>0.2182</td>
</tr>
<tr>
<td></td>
<td>BERT</td>
<td>0.3527</td>
<td>0.2360</td>
<td>0.2267</td>
<td>0.2249</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Answer</th>
<th>% MRR improvement</th>
<th>% nDCG@5 improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>24.56%*</td>
<td>19.06%*</td>
</tr>
<tr>
<td>Negative</td>
<td>31.20%*</td>
<td>25.84%*</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Answer</th>
<th>% MRR improvement</th>
<th>% nDCG@5 improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short</td>
<td>33.12%*</td>
<td>26.65%*</td>
</tr>
<tr>
<td>Medium</td>
<td>30.83%*</td>
<td>23.93%*</td>
</tr>
<tr>
<td>Long</td>
<td>23.41%*</td>
<td>20.36%*</td>
</tr>
</tbody>
</table>

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 DNN-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.

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, · · · , 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.

Table 4: 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.

<table>
<thead>
<tr>
<th>Answer</th>
<th>% MRR improvement</th>
<th>% nDCG@5 improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short</td>
<td>33.12%*</td>
<td>26.65%*</td>
</tr>
<tr>
<td>Medium</td>
<td>30.83%*</td>
<td>23.93%*</td>
</tr>
<tr>
<td>Long</td>
<td>23.41%*</td>
<td>20.36%*</td>
</tr>
</tbody>
</table>

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 DNN-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
The results are reported in Table 4. According to the table, GT achieved higher improvements for shorter user responses. This is true for all metrics, two of which are reported in the table.

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

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 results are presented in Table 3. According to the table, GT achieved higher improvements for positive responses. The reason is that for positive responses, the clarifying question already contains some important terms about the 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.

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.
- $\sigma$-QPP: We use a simple yet effective query performance predictor, $\sigma$ [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 $\sigma$. The clarifying question that leads to the highest $\sigma$ is selected.

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.

Table 5: Results for the next clarifying question selection task, up to 3 conversation turns. $^\dagger$ and $^\ddagger$ 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>MRR</th>
<th>nDCG@1</th>
<th>nDCG@5</th>
<th>nDCG@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>OriginalQuery</td>
<td>0.2715</td>
<td>0.1381</td>
<td>0.1451</td>
<td>0.1470</td>
</tr>
<tr>
<td>$\sigma$-QPP</td>
<td>0.3570</td>
<td>0.1960</td>
<td>0.1938</td>
<td>0.1812</td>
</tr>
<tr>
<td>LambdaMART</td>
<td>0.3558</td>
<td>0.1945</td>
<td>0.1940</td>
<td>0.1796</td>
</tr>
<tr>
<td>RankNet</td>
<td>0.3573</td>
<td>0.1979</td>
<td>0.1943</td>
<td>0.1804</td>
</tr>
<tr>
<td>BERT-NeuQS</td>
<td>0.3625</td>
<td>0.2064</td>
<td>0.2013</td>
<td>0.1862</td>
</tr>
<tr>
<td>GT-Docs-STL</td>
<td>0.3784$^\dagger$</td>
<td>0.2279$^\dagger$</td>
<td>0.2107$^\dagger$</td>
<td>0.1890</td>
</tr>
<tr>
<td>GT-Docs-MTL</td>
<td>0.3928$^\dagger$</td>
<td>0.2410$^{\dagger*}$</td>
<td>0.2257$^{\dagger*}$</td>
<td>0.1946$^{\dagger*}$</td>
</tr>
<tr>
<td>Oracle-Worst Question</td>
<td>0.2479</td>
<td>0.1075</td>
<td>0.1402</td>
<td>0.1483</td>
</tr>
<tr>
<td>Oracle-Best Question</td>
<td>0.4673</td>
<td>0.3031</td>
<td>0.2410</td>
<td>0.2077</td>
</tr>
</tbody>
</table>
Table 6: Some examples with a single clarification turn. ∆MRR is compared relative to the BERT performance.

<table>
<thead>
<tr>
<th>Information need</th>
<th>Find sites for MGB car owners and enthusiasts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>mgb</td>
</tr>
<tr>
<td>Clarifying question</td>
<td>are you looking for what mgb stands for?</td>
</tr>
<tr>
<td>User response</td>
<td>no</td>
</tr>
<tr>
<td>∆MRR</td>
<td>+78%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Information need</th>
<th>What restrictions are there for checked baggage during air travel?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>air travel information</td>
</tr>
<tr>
<td>Clarifying question</td>
<td>where are you looking to travel to?</td>
</tr>
<tr>
<td>User response</td>
<td>doesn’t matter i need information on checked baggage restrictions during air travel</td>
</tr>
<tr>
<td>∆MRR</td>
<td>3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Information need</th>
<th>What states levy a tax against tangible personal property?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>tangible personal property tax</td>
</tr>
<tr>
<td>Clarifying question</td>
<td>would you like to find out how much you owe?</td>
</tr>
<tr>
<td>User response</td>
<td>no i just want to know which states levy a tax against tangible personal property</td>
</tr>
<tr>
<td>∆MRR</td>
<td>-21%</td>
</tr>
</tbody>
</table>

- 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 ∆MRR 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


