Leading Conversational Search by Suggesting Useful Questions

Corby Rosset, Chenyan Xiong, Xia Song, Daniel Campos,
Nick Craswell, Saurabh Tiwary, Paul Bennett
Microsoft AI & Research
{corbin.rosset, chenyan.xiong, xiaso, campos.daniel,
nickcr, satiwary, paul.n.bennett}@microsoft.com

ABSTRACT

This paper studies a new scenario in conversational search, *conversational question suggestion*, which leads search engine users to more engaging experiences by suggesting interesting, informative, and useful follow-up questions. We first establish a novel evaluation metric, *usefulness*, which goes beyond relevance and measures whether the suggestions provide valuable information for the *next step* of a user's journey, and construct a public benchmark for useful question suggestion. Then we develop two suggestion systems, a BERT based ranker and a GPT-2 based generator, both trained with novel weak supervision signals that convey past users' search behaviors in search sessions. The weak supervision signals help ground the suggestions to users' information-seeking trajectories: we identify more coherent and informative sessions using encodings, and then weakly supervise our models to imitate how users transition to the next state of search. Our offline experiments demonstrate the crucial role our "next-turn" inductive training plays in improving *usefulness* over a strong online system. Our online A/B test in Bing shows that our more useful question suggestions receive 8% more user clicks than the previous system.

KEYWORDS

Conversational Search, Question Suggestion, Usefulness

ACM Reference Format:
Corby Rosset, Chenyan Xiong, Xia Song, Daniel Campos, Nick Craswell, Saurabh Tiwary, Paul Bennett. 2020. Leading Conversational Search by Suggesting Useful Questions. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, San Francisco, USA, 11 pages. https://doi.org/10.1145/3366423.3380193

1 INTRODUCTION

Commercial search engines have evolved beyond the "ten blue links" paradigm and now provide more direct natural language answers, summaries of web content, and knowledge graph semantics on the search engine results page (SERP). These features are crucial to move toward a more natural, engaging, and conversational search experience that better satisfies users' information needs. The trends in search logs have shown that users prefer preliminary forms of conversational functionality: the average web query length has significantly grown in the past decade, and in August 2019, the majority of Google search sessions (50.3%) did not end in a click [9]: users prefer crisp answers with precise and easy to access information, rather than reading long documents.

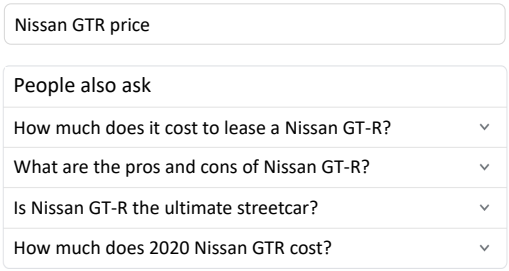

Figure 1: A Conversational Question Suggestion Example.

Moving toward conversational search faces two key challenges. First, while queries have become longer, many users still issue keyword queries after years of experiencing search engine failures on natural language questions – we must subtly encourage users to ask natural language questions now that they can be handled more effectively. Second, to be truly conversational, a search engine must go beyond answering the current query and *lead the conversation* forward for a more interactive experience.

This paper presents significant progress toward conversational search, *conversational question suggestion* – which aims to proactively engage the user in conversation-like experiences by suggesting natural language questions. Instead of only addressing a user's *current information needs*, we aim to suggest questions users would be interested in for the *next step of their inquiry*. While the current information need is often addressed in organic results (e.g. main body SERP) or extracted answers (e.g. top of SERP), one way of further integrating a conversational experience is to provide interesting question suggestions – assuming a user will finish their current query through a search result and then potentially continue the inquiry. For example, for the query "Nissan GTR Price" in Figure 1, the targeted question suggestions include those that help user complete a task ("leasing deals"), weigh options ("pros and cons about GTR"), explore an interesting related topic ("ultimate streetcar"), or learn more details ("2020 GTR"). These question suggestions are more "conversation leading" in that, after (assuming) the information needs of the current query is satisfied, they *lead* the user to an immersive search experience with diverse and fruitful future outcomes.

The next generation of conversational search experiences also require new evaluation metrics beyond the classic notion of relevance. By default, all suggestions should be relevant to the query; but a relevant question is not necessarily "conversation leading". For example, a user who searched "GTR Price" likely already knows the answer to "What is Nissan GTR", which does not provide *forward*-looking information. Thus we propose a novel evaluation metric

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380193

for conversational question suggestion, *usefulness*, which measures whether a suggestion leads the user to valuable and engaging information. We provide guidelines for this metric, create the judgment pools, and construct the first usefulness benchmark[1].

We propose two conversational question suggestion systems: `DeepSuggest`, a BERT based ranking system [8], and `DeepSuggest-NLG`, a GPT-2 based generation system [25]. Both are trained in a multi-task learning framework on a combination of relevance and user feedback signals. In addition, to make question suggestions more useful, this paper presents a new inductive weak supervision method that imitates users' information seeking behaviors in search sessions. Using intent-oriented query encodings [36], we develop a novel search log mining technique that effectively identifies coherent and informative search sessions. These search sessions are injected as an auxiliary Next Query Prediction task to train our suggestion models. They induce the models to prefer questions that are more aligned with users' behavior while they naturally transition to the next state of their search journey.

We apply our investigation to the "People Also Ask" (PAA) function, a search feature provided by commercial search engines (e.g. Google and Bing). Our experiments include intrinsic evaluation with historical user clicks and relevance labels, human evaluation on *usefulness*, and online A/B tests on Bing PAA.

The intrinsic evaluation shows that all the evaluated models perform well in finding *relevant* suggestions or in predicting *offline* users' clicks. However, a relevant suggestion is not necessarily useful. Our online A/B test in Bing verifies that the new *usefulness* metric better reflects *online* users' preference: all the studied methods perform similarly on relevance, but actual search engine users strongly prefer the more useful suggestions.

Our experiments further demonstrate the advantage of our inductively trained `DeepSuggest` in providing more useful suggestions. It improves a strong online production baseline by 36% on *usefulness* in our head-to-head human evaluation. This gain mainly comes from the inductive weak supervision from carefully mined conversation-like search sessions. The gains on *usefulness* also lead to strong online performance: `DeepSuggest` improves the click-through rate of Bing PAA by more than 8%, a remarkable movement in commercial search. Moreover, our method led to higher overall "success rate" in Bing, showing that users prefer useful question suggestions, and that our search engine has made meaningful progress providing "conversational" search experiences.

We also provide topics for future study on generative conversational question suggestion. We found that `DeepSuggest-NLG` generates syntactically correct and meaningful questions, but might be confused by the variance of *language styles* in queries versus questions, nor our naive use of GPT-2 effectively captures relevance or usefulness. To the best of our knowledge, this is the first thorough study of GPT-2 in conversational search, and our observations point to several interesting future directions.

The next section discusses related work. Section 3 and 4 describe our *usefulness* metric and question suggestion framework. Section 5 presents our conversation leading weak supervision. Section 6, 7, and 8 describes experimental methodologies, the usefulness benchmark, and evaluation. We conclude in Section 9.

[1] Available at http://aka.ms/usefulness.

2 RELATED WORK

Conversational search is deeply rooted in information retrieval [5, 6] and has recently garnered active research. Many popular research topics in information retrieval are under its broad umbrella [10, 26]. Notable topics include session-based search, which improves retrieval accuracy using previous interactions between a user and a search engine [3, 4, 18, 19], query suggestion, which helps a user complete their search utterance using previous context and search log information [32, 34], and related query recommendation, which provide interesting related queries to help user explore the information space [2, 11].

Among these topics, a recent related one is "learning to ask", which studies the interactive aspect of conversational search and aims to proactively ask clarification questions to the user, in order to better understand a user's information needs. This task has been studied in a wide range of application scenarios, for example, in open-domain IR to improve retrieval relevance [1], in recommendation system to ask questions for better recommendation quality [37], and in the question generation setting as an application of natural language generation [30, 31]. Previous work has mainly attempted to better understand a user's information needs by resolving ambiguity, rather than leading the conversation with questions a user may want to ask in the next step.

The evaluations of previous conversational search or dialog systems mostly pertrain to relevance, diversification, and/or the quality of the generated contexts – for example, standard relevance metrics on the final retrieval results [1, 4], the ability to recover the next session queries [3, 32], and the quality of generated classification questions [30, 31]. Our usefulness metric is related to the recent research in dialog systems that evaluates their ability to conduct on-topic and target-oriented conversations [33].

Deep pre-trained language models, especially BERT [8], have shown strong empirical results in ad hoc passage and document retrieval [7, 20, 21]. Nogueira and Cho effectively leverage BERT's strong sequence-to-sequence classification ability in passage ranking by feeding BERT the concatenated query and passage texts [21]. Dai and Callan fine-tune BERT in a similar way in ad hoc ranking using clicks from Bing search logs [7]. Sean et al. combine BERT with previous neural IR models for better accuracy [20]. The strong effectiveness of BERT rankers also leads to various analyses to understand their advantages in retrieval [22, 24]. On the generation front, the most related research is generative query suggestion, which mainly uses an RNN based encoder-decoder structure [32] and not yet the pre-trained deep transformers such as GPT-2 [25].

There are many recent studies on how to better pre-train and fine-tune deep language models. ELMO uses masked language model training [23] and BERT further adds the next sentence prediction task [8]. XL-NET uses permutation language model task [35]. RoBERTa shows that more training data and a longer pre-training period lead to significantly better effectiveness [17]. AL-BERT introduces the sentence coherence task [14]. MT-DNN and T5 combine various language modeling tasks in multi-task learning [16, 27]. The focus of these methods is to improve the capability of deep pre-trained neural networks, not to add new properties to the neural model inductively via specifically constructed weak supervision signals.

3 FROM RELEVANCE TO USEFULNESS

This section defines the *usefulness* metric for conversational question suggestion, including our motivation and labeling guidelines.

3.1 Beyond Relevance

A natural offline quality metric for conversational question suggestion is binary "relevance": whether suggestions are on topic and related to the query. However, as the models we evaluated became more effective, we observed a discrepancy between the relevance offline metric and online user preference. Question suggestions that were more relevant were not always preferred by the web users. For example, the suggestion "What is Deep Learning?" is related to the query "transformer architecture", but the user searching for transformer probably already knows what deep learning is, so the suggested question provides no new value to them.

Nearly all current systems excel when measured on relevance, which makes establishing a relationship between offline and online movements difficult. Also, relying solely on online user feedback signals may result in a dangerous preference for "click-bait" suggestions. There is a need for an offline metric for conversational question suggestion that 1) reflects if a user will find value in the information the suggestion leads her to, and 2) is correlated with real online user preferences.

The *usefulness* metric measures whether the suggestion for a query brings real value to the user, which could be in the form of new information she needs, the next step to complete a task, or helping her ask the right questions to explore a topic. A useful question should further the users' information need by bringing them to a fruitful next state in their search.

We view *usefulness* as the next generation evaluation metric that pushes conversational search models to meet higher standards and to enrich the user experience. We have several online experiments showing that useful suggestions are preferred by users, though they are not necessarily more relevant. We will show one such online experiment in Section 8.1.

3.2 Usefulness Guidelines

Here we define the guidelines of the *usefulness* metric. In addition to the useful label, this metric also reports five failure modes as subcategories of not useful, which help diagnose the common challenges of conversational search systems and also help ensure annotation quality. Some label examples are shown in Table 1.

Misses Intent. Suggestions that are completely off-topic, poorly formatted, nonsense, or look like non-natural language are Misses Intent. For instance, if the query is "play Netflix movies on TV", the suggestion "How to play DVD videos on TV?" is not useful since a user streaming Netflix may not find value in DVDs.

Too Specific. A suggestion that wouldn't apply to a significant portion of the population is Too Specific. For example, for the query "book cheap flights", the question "What are some cheap airlines which fly to London?" is too specific since it only applies to the small fraction of users flying to that specific destination.

Prequel. A suggestion about something the user likely knew when issuing the query is a Prequel. For example, if the user asks "game of thrones S8 release date", a question like "What is game of thrones?" is not helpful since the user has probably watched previous seasons and knows what the series is about.

Table 1: Examples of Query-Question Suggestion Pairs and their Usefulness Labels.

Query	Question Suggestion	Gold Label
used washer and dry	Can I store a washer and dryer in the garage ?	Misses Intent
best questions to ask interviewer	What should I ask in an interview ?	Dup. w/ Q
medicaid expansion	Did Florida accept Medicaid expansion ?	Too Specific
verizon yahoo purchase	Who bought out Yahoo ?	Prequel
jaundice in newborns	How to tell if your newborn has jaundice ?	Dup. w/ Ans.
jonestown massacre	What was in the Kool-Aid at Jonestown ?	Useful
affirmative action	Who does affirmative action benefit ?	Useful
best hair clippers	What clippers do barbers use ?	Useful

Duplicate with Query. If the suggested question has the same intent as the query, it is duplicate. One way to analyze overlapping intent is whether the query and suggestion would be satisfied by the same set of documents, for instance "What is good for heartburn relief?" and the query "heartburn remedies".

Duplicate with Answer. The Answer is the natural language answer shown on top of the search result page. Assuming the user has already read the answer, if the question merely re-states information from it, then it is Duplicate with Answer.

Useful. A useful question leads to valuable information. The first way this can happen is that the answer to a useful question helps complete a task the user has in mind. For instance, if the query is "painting outdoor deck", the suggestion "What is the best weather-proof paint for outdoor decks?" helps address a need that follows from their task. A useful suggestion may also re-frame the user's task in a different perspective, such as the suggestion to "best hair clippers" in Table 1, which reduces the task of purchasing the best clippers to a more practical task of finding which brands are popular among professionals.

A question can be useful if it engages the user in further exploration of a topic that she may want to know more about, in a logically coherent manner. For example it is common for conversation-like sessions to "pivot" on important points of a topic, e.g. see the "jonestown massacre" example in Table 1. However, the stipulation of "logical coherence" warns against drifting too far from the user's original intent. In other words, a suggested question is useful if its answer adds value about a topic that follows a coherent line of thought, so as to lead the "conversation" between the user and the search engine.

4 QUESTION SUGGESTION FRAMEWORK

In this section, we first introduce the multi-task learning setup to train question suggestion models and then two conversational question suggestion approaches.

4.1 Multi-Task Learning

We use four tasks in our multi-task learning setup. The first three described in this section are standard training tasks in PAA. The fourth is a novel task of imitating user trajectories in conversation-like sessions. This section describes the first three PAA tasks, with one using relevance labels and two gathered from online user feedback in PAA. The fourth task is described in Section 5.

Relevance Classification. A natural first choice for offline quality evaluation is standard binary relevance. We use human judgments on about 600K (q, s) pairs, with label $y = 1$ meaning the question s is relevant to the query q and $y = 0$ otherwise.

Relative-CTR Prediction is a classification task using data collected from user clicks on question suggestions in PAA. Specifically, as multiple suggestions are displayed for the same q multiple times, we collect the suggestion pairs that have significantly different click through rates (CTR) for the same query:

$$\text{R-CTR Pairs} = \{(q, s^+, s^-) | \text{CTR}(q, s^+) \geq c + \text{CTR}(q, s^-)\},$$

where $\text{CTR}(q, s)$ is the click through rate (probability of being clicked) of s when shown for q. We use $c = 30\%$ for emphasis on the pairs with significantly different user preferences. We also ensure the pair has been displayed sufficient times to avoid randomness.

We then formulate the tasks as a pointwise learning to rank problem with $(q, s^+, y = 1)$ and $(q, s^-, y = 0)$ [15].

PAA Click Prediction is a standard click prediction task using user clicks as the relevance feedback labels. We collect user click signals from a random sample of the search log, with each impression of PAA as a training instance: i.e. $y = 1$ iff the suggestion s is clicked and $y = 0$ if not.

The three tasks are used jointly to train conversational question suggestion systems in a straightforward multi-task setting: The three tasks are randomly mixed in each training batch and the model is updated by the gradients from all three tasks simultaneously. Among the three, the PAA Click Prediction is the primary task and the model's prediction from this task is used for question suggestion. The Relevance Classification and R-CTR Prediction are auxiliary tasks to avoid irrelevant or known bad suggestions.

4.2 Ranking Suggestions

The first system, `DeepSuggest`, fine-tunes BERT to rank suggestions [8]. We follow the standard way to apply pre-trained BERT in ranking [21]: given the query q and the candidate suggestion s, we concatenate ($\circ$) the two and feed the pair to BERT to get their contextualized representation:

$$\phi(q, s) = \text{BERT}([\text{CLS}] \circ q \circ [\text{SEP}] \circ s \circ [\text{SEP}]). \quad (1)$$

The representation of the last layer's "[CLS]" is used as the representation for the pair $\phi(q, s)$.

Then a linear "learning to rank layer" is applied on the contextualized representation to calculate the ranking score:

$$f(q, s) = \text{Linear}(\phi(q, s)). \quad (2)$$

The BERT Ranker is fine-tuned in the multi-task learning using standard cross entropy loss on our binary labels: relevance, R-CTR, and user clicks, etc. The ranker is used in our standard retrieval and re-ranking pipeline; it ranks the candidate questions retrieved by a base system using $f(q, s)$.

4.3 Generating Suggestions

In addition to the ranking model, we also explore the potential of natural language generation (NLG) models in conversational question suggestion. A fully generative conversational system has many intriguing advantages. It can reduce pipeline complexity: instead of separate systems for candidate curation, retrieval, and ranking, potentially one model is sufficient to generate suggestions solely using the input query; it can also increase the coverage of the suggestions, as no pre-existing suggestion candidates are required.

Our generation model, `DeepSuggest-NLG`, fine-tunes GPT-2, the deep transformer based NLG model [25]. Specifically, GPT-2 outputs the language modeling probability that a token x_i follows the previous tokens $x_{<i} = \{x_1, ..., x_{i-1}\}$ in the sequence:

$$p(x_i | x_{<i}) = \text{softmax}(\text{MLP}(h_L^{i-1})). \quad (3)$$

GPT-2 uses the pre-trained deep unidirectional transformer to obtain the hidden representation h_L^{i-1} of $x_{<i}$ in the last layer (L). Then it uses a language modeling head (MLP) to predict the generation probability of the next token x_i.

To adapt GPT-2 to conversational question generation, we use the positive suggestion $\{(q, s^+) | y = 1\}$ as its training target given the context (we do not want it to generate irrelevant or unclicked suggestions), and concatenate them into the input sequence:

$$x = q \circ [\text{SEP}] \circ s^+ \circ [\text{EOS}]. \quad (4)$$

[EOS] is the "end of sequence" token to stop generation [25].

As the first step to explore the effectiveness of GPT-2 in generating question suggestions, we use the vanilla fine-tuning setup of GPT-2 and leave more sophisticated approaches to future research: We feed the concatenated sequences x to GPT-2 and use maximum log likelihood training on all the tasks in the multi-task learning setup. During inference, we feed the question sequence "$q \circ$ [SEP]" to GPT-2 and let it generate the question suggestions directly, without using any candidate questions.

5 IMITATING USER SEARCH TRAJECTORIES

Relevance feedback from users, though widely used as weak supervision, has several challenges due to its self-biasing nature [12]. Feedback signals are only available for those suggestions ranked highly by the production system and presented to the user; a system trained solely on this biased feedback may get stuck in a local optimum. In addition, a suggestion that receives many user clicks is not necessarily useful, and we don't want to be vulnerable to "click bait" suggestions. While attracting users' attention and clicks is important, a responsible AI system should empower all its users by providing valuable information to them.

In this section, we present a novel weak supervision method that imitates the information-seeking trajectories underlying conversation-like search sessions. We first develop a new search log mining technique, which automatically identifies search sessions that include more "next-turn" information provided by users. Then we discuss how to inject the "next-turn" information to our neural models as inductive weak supervision signals.

5.1 Mining "Conversational" Search Sessions

The search sessions include sequences of queries from users, thus are more decoupled from the search system. A session may capture the user's information-seeking trajectory when completing a task, their train of thought when learning a concept, or shift of interests when exploring a topic. Search sessions are also noisy: users may be multi-tasking and switch between different information needs; the queries in a session may not even be related to each other.

To identify conversation-like sessions we first clean sessions to reduce noise and identify those with meaningful engagement. Then we develop a new embedding-based search log mining technique that identifies sessions that are coherent and information-seeking. Our approach generalizes the work on identifying intrinsically diverse search sessions [28, 29] to conversational search sessions by using a graph-based model of coherence, an embedding optimized for representing search intent, and an emphasis on natural language questions. The technique includes the following three steps.

Clean. The first step is to clean the raw sessions. Starting from the raw sessions grouped by the standard 30 minutes gap rule, we discard navigational queries, e.g., "Facebook Login", and mal-intent queries, using an in-house query classifier. Then we only retain sessions which have at least three remaining queries and with at least one satisfied user click on a document. This ensures that there exists some basic information need expressed in the session, with some complexity (multiple queries), and the trajectory was successful to some degree. We call this data `Clean` sessions.

Coherent. The next step is to ensure the queries in a session have a coherent information need and do not merely co-occur within 30 minutes. We utilize the GEN Encoder, which maps queries with similar intent closely in the embedding space [36], to embed the session queries and determine their coherence.

Specifically, for each `Clean` session, we construct a graph using its queries as nodes and connect queries by edges if they have an encoding similarity above the "unrelated" threshold (0.4) [36]. Then the largest connected sub-graph in the session is retained; the other queries are discarded. If the largest connected component still contains at least one satisfied click and at least three queries, then the session is kept as "`GEN` sessions". This ensures that the intents between queries in a session do not drift too drastically, yielding more coherent information needs.

Information-Seeking. To focus on information-seeking search sessions, we use the QA intent classifier in the search engine to find search sessions that are information-seeking. The assumption is that if a query is satisfied by an extracted natural language answer on the SERP, then the query itself describes an information-seeking intent rather than a navigational, transitional, or functional one.

Following this intuition, the third step filters the `GEN` sessions to those that had at least one satisfied interaction with a natural language answer in one of its queries. This leads to our final `QA-GEN` sessions that are cleaner, more coherent, and more informational.

5.2 Inductive Weak Supervision Task

The mined sessions are used as inductive weak supervision signals to train our question suggestion models. We use the standard Next Query Prediction task from the query suggestion literature to inject the "next-turn" signals in the sessions [32, 34].

Specifically, for a session $\mathcal{S} = [q_1, ..., q_{n-1}, q_n]$, the next query prediction task is to predict the last query q_n using the previous queries $q_{<n}$. Following the pointwise setting in Section 4.1, we format the positive pairs as $(q_{<n}, q_n, y = 1)$:

$$(q_1 \circ [\text{NQY}] \circ q_2 ... \circ [\text{NQY}] \circ q_{n-1}, q_n, y = 1). \quad (5)$$

The session queries are delimited with the special separator "[NQY]".

The generation model GPT-2 can be directly trained with these positive pairs as the fourth task in its multi-task learning. The training of BERT ranking models further requires negative instances ($y = 0$). We follow previous research in query suggestion and procure negatives using the the ADJ method [32]:

$$\text{ADJ}(\mathcal{S}) = \{q^- | q^- \text{ appears frequently after } q_{n-1}\}. \quad (6)$$

It includes the K most frequent queries that appear after q_{n-1} in the search log. We use them as negative instances and train our BERT ranker with the next query prediction task as the fourth task in the same multi-task learning setup.

6 EXPERIMENTAL METHODOLOGIES

Our experiment methodologies include several phases. The first is offline training—the data collection and training for the four (weak) supervision tasks. The second is to conduct *intrinsic evaluations* on the four offline tasks. Third, we form a usefulness benchmark by conducting TREC-style zero-shot human evaluation of usefulness on the various techniques pooled together. Finally, the best methods in offline evaluation are sent to online A/B test with real user traffic.

This section describes the training phrase (Sec. 6.1), the intrinsic evaluation phrase (Sec. 6.1), and the evaluated methods (Sec. 6.3). The usefulness and online evaluation are described in later sections.

6.1 Multi-Task Training Setup

There are in total four training tasks in our multi-task learning: three from People Also Ask (Sec. 4.1) and one inductive weak supervision from mined search sessions (Sec. 5.2).

PAA Click Prediction. This is the primary PAA task in this study, which we formulated as a pointwise learning-to-rank task to label clicked questions for a query as positive and rest as negative.

We experiment with two different settings in the PAA Click Prediction. The first is the single-turn setting where only the current query is provided. The second is the Contextual (`+ Context`) mode where all previous queries in the corresponding session are provided, separated with special delimiter tokens, to explore the impact of conditioning the model on more session context. We sample 2.5M, 5.0M, and 7.5M-instance datasets for both settings, uniformly and randomly, from the search log at Bing in January-February 2019.

PAA Relative CTR Classification. This task requires models to classify whether a question suggestion historically has higher relative-CTR than another for the query. We sample 2.5M triples of the form (query, high-CTR question, low-CTR question) collected from 2018 search logs; all questions have at least 10 clicks and the difference in CTR between the two suggestions is at least 30%.

PAA Question Relevance Classification. This is the minimum requirement of a conversational suggestion system. It includes about 600K query-question pairs with human-judged binary labels on the relevance of the question to the query.

Next Query Prediction in Mined Sessions. This task uses the inductive weak supervision to promote more useful next-turn question suggestions. All variations of mined sessions (`Clean`, `GEN`, and `QA-GEN`) consist of 2.5M sessions mined from the first two months of 2019. Given the context, the task is to rank the true last query over the ADJ negative queries [32]. The frequency in ADJ are calculated from a large fraction of the 2018 search log.

Multi-Task Training. For all experiments, we randomly interleave batches from each constituent task training, the simplest multi-task setup. We vary the number of PAA click labels for different task combinations, to make sure all models see *exactly the same amount of labels (7.5M), only from different task combinations.*

6.2 Intrinsic Evaluation

The intrinsic evaluation studies whether the model effectively fits the four training tasks using held out data. A model also must perform well in the relevance task which is the minimum requirement—we want the question suggestions to be both relevant and useful.

Intrinsic Evaluation Datasets and Metrics. The intrinsic evaluation is conducted for each of the four training tasks.

(1) PAA Click Prediction includes 10k validation and 10k testing. This ranking task is evaluated by `MRR`. There are four question candidates per query from the online system.
(2) PAA Relative CTR Classification includes 10k validation and 10k testing instances, in the same format with its training split. It is evaluated by classification accuracy (`R-CTR ACC`).
(3) PAA Relevance Classification includes 10k validation and 1k testing split. There are 520 negative and 440 positive labels in the testing split. It is evaluated with AUC (`Rel-AUC`).
(4) Next Query Prediction also includes 10k-10k validation and testing split, and is evaluated by MRR (`Q-MRR`). There are at most ten candidates per query, mined by ADJ [32].

There is no overlap between any testing and training splits.

BERT Ranking Inference. At test time, for BERT ranking models, if the test task was *not* present during training, the PAA Click Prediction score is used as a surrogate score. Otherwise, the model's dedicated scoring layer for each respective task is used by default.

GPT-2 Generation Inference. Evaluating generated textual results is challenging. The n-gram overlap with the target question/query may not reflect their closeness in meaning, relevance, or online user preference; often human evaluation is required—which is the case for usefulness evaluation.

In the *intrinsic evaluations*, we use the GPT-2's perplexity on the candidate question/query as a surrogate score, in the same way as the BERT output scores are used. Thus the intrinsic evaluations only test the GPT-2 models' ability to rank or classify items based on perplexity, not generation.

6.3 Compared Methods

The rest of this section describes the baselines, our methods—including their variations—and implementation details.

Baselines. include an `online` production system and the vanilla, single-task, `BERT` and `GPT-2`.

`Online` is a recent version of Bing PAA. It is a highly optimized retrieval and re-rank pipeline system. The retrieval consists of various search log mining techniques and provides thousands of candidate questions per query. The candidate questions are then fed into a strong ELMO [23]-like ranker, in a standard seq2seq with attention setup. The ranker is trained using similar tasks as described in Section 6.1, including R-CTR and relevance.

The `online` pipeline also includes various stages for relevance lower bounding, de-duplication, and diversification, etc. All our ranking-based methods use the exact same pipeline, except replacing the ranking component with our models.

`BERT` is the ranking model in Sec. 4.2, only trained by the PAA Click Task, but using the same *total label amount* as our methods.

`GPT-2` is the generation model described in Sec. 4.3, using the same single task training data as `BERT`.

There are two variations of BERT and GPT-2 (Sec. 6.1): one only sees the current query; the other takes all previous session queries concatenated with the current query (`+ Contexts`).

Our Methods include `DeepSuggest` and `DeepSuggest-NLG`.

`DeepSuggest` is the BERT ranker trained with all four training tasks. All tasks share the same parameters except the last task-specific linear layer. In usefulness and online testing, the PAA Click Prediction task score is used to rank the candidate suggestions from the retrieval pipeline of `online`.

`DeepSuggest-NLG` is the GPT-2 generator trained with the same four tasks. We found it more effective to use the same parameters for all tasks. For usefulness evaluation we generate 10 outputs from GPT-2 using nucleus sampling with $p = 0.6$, with the user queries as *the sole input*. The generation is stopped when producing "[EOS]" or 40 tokens. The generated outputs are deduplicated and ranked by their perplexities.

Model Variations. To study the effectiveness of the inductive weak supervision from the mined conversational sessions (Sec. 5.2), we evaluate variations of our methods with only the Next Query Prediction task added *as the only auxiliary task* to PAA Click Prediction in the context mode. Varying the search sessions used in the auxiliary task leads three variations for BERT and GPT-2 each.

`Clean` uses the clean sessions in the first search log mining step described in Sec. 5.1. It is the "baseline" search sessions.

`GEN` uses coherent and clean sessions filtered by Gen Encoder [36].

`QA-GEN` further adds the QA intent requirement on the `GEN` sessions. It is the "conversational" session used in our final methods.

Implementation Details. All BERT models use BERT Large and GPT-2 models use the 345 Million parameter instance. Both have 24 transformer layers. Our implementations are based on pytorch-transformers[2]. Our models start from the released pre-trained parameters and are fine-tuned on our tasks for three epochs.

We used the Apex fp16 Adam optimizer and distributed training on 4 Nvidia V100 GPUs. The *effective* batch size is 64. We truncated all training instances (queries and questions concatenated) to maximum 128 tokens and each query to at most 32.

The learning rate is scheduled as a linear function of batch number that increases from 0.0 to a maximum over 6k steps, and then exponentially decays over the next 360k steps to a minimum. The maximum and minimum were $5e^{-6}$ and $1e^{-6}$ for BERT, and $5e^{-5}$ and $5e^{-6}$ for GPT-2. We have obtained better results with more tuning of learning rates and new schedulers, but decided to demonstrate the most straightforward settings.

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

Table 2: The overlap of question suggestion results among our five BERT ranker or GPT-2 generator variations. The number of Unique question suggestions VS. Total suggestions at different ranking Depths are shown.

Group	Depth	# Unique	# Total	% Unique
Bert Rankers	1	2188	3900	56%
	2	3950	7800	51%
	3	5477	11700	47%
	4	6963	15600	45%
GPT-2 Generators	1	2990	3794	79%
	2	5838	7582	77%
	3	8589	11139	77%
	4	11250	14372	78%

Table 3: Aggregated co-occurance rate of each of the judges' labels (columns) with the Mode (rows) for each instance, normalized per row. The number of pairs with each label is shown in brackets. Columns are initials of method names.

Majority Vote	MI	TS	P	D w/ Q	D w/ A	U
Misses Intent (1536)	**0.77**	0.03	0.03	0.02	0.01	0.14
Too Specific (46)	0.17	**0.62**	0.00	0.04	0.01	0.16
Prequel (103)	0.20	0.00	**0.59**	0.07	0.04	0.11
Dup w/ Query (431)	0.07	0.01	0.03	**0.75**	0.06	0.09
Dup w/ Answer (40)	0.10	0.01	0.06	0.10	**0.50**	0.24
Useful (666)	0.16	0.03	0.01	0.05	0.04	**0.70**

7 BENCHMARKING USEFULNESS

We benchmark usefulness by conducting a "TREC" style evaluation: we first construct a judgment candidate set by *pooling* the top results from all evaluated methods, then we recruit judges to *manually annotate* the pool, overseen by various *quality controls*, to reach reasonable judge *agreements*. This also produces a reusable *usefulness benchmark* for future research.

Judgment Pool Construction. The first step is to construct a set of (query, question suggestion) pairs for judges to label. In total we sampled about 780 queries that triggered Bing PAA; none of them appear in our training data. We use the standard TREC approach and pool the top K question suggestions from all methods.

Specifically, we pool from 13 systems in this study: the five baselines, `Online`, `BERT`, `BERT + Contexts`, `GPT-2`, `GPT-2 + Contexts`, our two main methods, `DeepSuggest` and `DeepSuggest-NLG`, and the six two-task version of BERT and GPT-2 with three different mined sessions. The pooling depth (K) is set to four, the same number Bing PAA displays. The size of the pool for the 780 queries is shown in Table 2. There is little overlap between the ranked and generated questions.

Manual Annotation. In total about 2800 query-question pairs from a uniform subsample of 50 queries (from the total 780) are labeled. After a judge training period with our guidelines (Sec. 3.2), annotations were conducted by about 25 qualified judges over several weeks. The judges are professional annotators whose full time job is to provide high quality labels for search.

Table 4: Intrinsic evaluation results on four training tasks: MRR of PAA Click Prediction, R-CTR ACCuracy of PAA Relative CTR Classification, AUC of PAA Relevance, and Q MRR in Next Query Prediction. Zero-shot results are marked by *. GPT-2 perplexity is used as the prediction score in intrinsic evaluation.

	People Also Ask			Session
	MRR	R-CTR ACC	Rel AUC	Q MRR
`Random`	≈ 58	≈ 50	≈ 50	≈ 37
`Online`	–	76.0	80.0	–
`BERT`	73.5	*82.8	*81.2	*29.2
`+ Contexts`	74.1	*81.8	*80.3	*32.9
`+ Clean Session`	74.6	*82.1	*81.2	**85.4**
`+ GEN Session`	**75.1**	*82.0	***82.4**	80.9
`+ QA-GEN Session`	74.4	*82.3	*82.0	84.8
`DeepSuggest`	71.3	**84.8**	80.4	65.2
`GPT-2`	70.5	*65.8	*39.7	*43.5
`+ Contexts`	70.9	*64.4	*41.9	*40.2
`+ Clean Session`	69.8	*65.2	*52.7	32.7
`+ GEN Session`	68.7	*66.9	*52.4	34.4
`+ QA-GEN Session`	68.9	*67.7	*46.1	33.9
`DeepSuggest-NLG`	69.3	81.1	52.3	34.1

For each query-question pair, the judges were presented with the search result page from Bing for the query, with the PAA part omitted. They were encouraged to search additional queries if more information is needed. The speed is about one label per minute.

Quality Control is conducted by several approaches. The first is the hidden "spam" test. About 1-2% data have gold labels; judges are required to be correct on at least 70% of them. The second uses instructive items in 5% of the pool, where the gold label with detailed explanations would appear after an annotation is submitted. Judge accuracy here is about 64%±4%. Lastly, we manually reviewed about 5% of instances chosen with a preference toward those with less agreement among judges. On these, judges agreed with us about 60% ± 4%. Judges who under-performed in our manual reviews were removed; their labels were discarded and re-done by others.

Agreement Each query-question pair was judged by at least five judges, with two more judges to break ties if necessary. Their majority vote was used as the final label. The pairwise agreement between judges is 53% ± 1%. The Cohen's Kappa score on this six category labeling task is 0.36 with a standard deviation of 0.15.

In Table 3 we show a confusion matrix of how often each label co-occurs with the majority vote (final label). The diagonal represents how many votes are concentrated in the mode label. The number of pairs labeled to each category is shown in brackets. Overall, the agreement with the majority is 67% ± 1%.

Usefulness Evaluation. *All* evaluated methods were represented equally in the pool and were judged at the same time period, by the same group of judges. The usefulness evaluation is a direct head-to-head comparison for all bench-marked systems. It is also *zero-shot*, as no usefulness labels were used in training. The resulting *usefulness* benchmark, with the personal and sensitive data removed, is publicly available[3].

[3] http://aka.ms/usefulness.

Table 5: Results of usefulness and categories of non-usefulness. BERT methods rank the same candidate questions with `Online`. GPT-2 methods directly generate question suggestions without using any inputs beside the input query and/or previous queries (+ `Context`) in the session. Relative improvements (%) are compared with `Online`. All our methods are trained with the same amount of labels but different combination of tasks.

	Useful		Misses Intent		Dup w/ Query		Too Specific		Dup w/ Ans		Prequel	
Online	0.320	–	0.446	–	0.120	–	0.017	–	0.034	–	0.063	–
BERT	0.245	-23%	0.365	-18%	0.315	+163%	0.005	-71%	0.02	-41%	0.045	-29%
+ Context	0.24	-25%	0.41	-8%	0.295	+146%	0.005	-71%	0.02	-41%	0.03	-52%
DeepSuggest	**0.434**	+36%	0.379	-15%	0.131	+9%	0.005	-71%	0.015	-56%	0.035	-44%
GPT-2	0.253	-21%	0.516	+16%	0.172	+43%	0.005	-71%	0.011	-68%	0.043	-32%
+ Context	0.296	-8%	0.43	-4%	0.218	+82%	0.006	-65%	0.022	-35%	0.028	-56%
DeepSuggest-NLG	**0.376**	+18%	0.412	-8%	0.139	+16%	0.01	-41%	0.015	-56%	0.046	-27%

Table 6: Usefulness results of `BERT + Contexts` when only adding the inductive training task from variant search sessions: `Clean` sessions, `GEN` coherent sessions, and `QA-GEN` coherent and informational sessions.

	Useful		Misses Intent		Dup w/ Query		Too Specific		Dup w/ Ans		Prequel	
BERT	0.245	–	0.365	–	0.315	–	0.005	–	0.020	–	0.045	–
+ Contexts	0.240	-2%	0.410	+12%	0.295	-6%	0.005	0%	0.020	0%	0.030	-33%
+ Clean Session	0.295	+20%	0.380	+4%	0.265	-16%	0.005	0%	0.015	-25%	0.04	-11%
+ GEN Session	0.300	+22%	0.350	-4%	0.285	-10%	0.010	100%	0.025	+25%	0.030	-33%
+ QA-GEN Session	**0.320**	+31%	0.395	+8%	0.230	-27%	0.015	+200%	0.015	-25%	0.025	-44%
DeepSuggest	**0.434**	+77%	0.379	+4%	0.131	-58%	0.005	0%	0.015	-25%	0.035	-22%

8 EVALUATION

Our evaluation starts with the *overall quality* of intrinsic measures and head-to-head usefulness; then we investigate the *influence of inductive weak supervision* from mined search sessions. We also provide *online A/B test* results of the best performing model and study the *challenges of generative question suggestion.*

8.1 Overall Quality

We first present the intrinsic evaluation of the four training tasks and then the zero-shot *usefulness* evaluation.

Intrinsic Evaluation. Table 4 presents the intrinsic evaluation results. `DeepSuggest` outperforms `Online` on the relative CTR and relevance classification tasks; the other two tasks were not evaluated for the `Online` system. The Relevance AUC is saturated and does not fully correlate with online user preferences (we don't expect that 80% of users are satisfied with PAA suggestions per se). Furthermore, inclusion of other tasks does not significantly impact MRR on PAA click task; the stubbornness of these two metrics motivated the development of the *usefulness* metric.

The GPT-2 perplexity "score" provides reasonable rankings of PAA questions, and subsequently the R-CTR pairs which are similar in nature. However, when trained with all tasks, `DeepSuggest-NLG` does not perform well on relevance AUC or next query MRR compared to `DeepSuggest`. Notwithstanding, using the perplexity as a ranking score is merely for intrinsic study; later experiments provide more thorough evaluations on GPT-2 and discussions.

Usefulness Evaluation. The *usefulness* results are in Table 5. `DeepSuggest` outperforms `Online` with 35.6% more useful question suggestions, by ranking the same question candidates as `Online`. There are significant drops in Misses Intent, Too Specific, and Prequel suggestions. In comparison, `BERT` and `BERT + Context`, the same neural model, trained with the same amount of signals but only using user clicks, perform much worse than `Online`.

That `DeepSuggest` significantly improves *usefulness* over the single-task `BERT` is in stark contrast with their comparable intrinsic evaluation results. This confirms that solely focusing on optimizing user clicks can promote dangerous "click baits" that are not actually useful. For instance, users may click on a suggestion that is a better reformulation of their query, even though it is a duplicate. We see a need for the inductive weak supervision task to provide complementary indications of what information could further empower our users. Section 8.2 further evaluates the influence of inductive weak supervision on our ranking models.

On the generation side, `DeepSuggest-NLG` also significantly outperforms `Online`. The single task `GPT-2` models perform better than the corresponding `BERT`, which is promising as the generation model is self-contained, requires no candidate retrieval, and can potentially increase the PAA coverage to a wider variety of queries. `DeepSuggest-NLG` also creates well-formed outputs, as syntactically incorrect suggestions are labeled as Misses Intent (<0.412).

Still, `DeepSuggest-NLG` performs worse than `DeepSuggest`, especially in Miss Intents and Prequel. This observation aligns with its low relevance in Table 4. Session 8.4 provides a further study on the challenges of generative question suggestion.

8.2 Inductive Weak Supervision in Ranking

This experiment studies the influence of inductive weak supervision in BERT based models. The *usefulness* results of BERT variations with different search sessions (Sec. 6.3) are shown in Table 6.

Table 7: Online A/B test results of DeepSuggest. The relative percentages are compared with Bing PAA online production. CTR improvements on the entire PAA block when triggered on TOP of the search result page and on Bottom are shown. The Overall Success Rate is the ultimate online metric that measures the effectiveness of the entire search engine.

	+% to Online
Online Click Rate (TOP)	+8.9%
Online Click Rate (Bottom)	+6.4%
Online Overall Success Rate	+0.05%
Offline Usefulness	+35.6%
Offline Relevance	+0.5%

Adding Clean sessions significantly improves BERT in *usefulness*. It stably reduces most of the non-useful categories, but misses more search intents: Queries in Clean sessions co-occur in 30 minutes but not necessary originate from similar information needs. Switching to more coherent GEN sessions leads to a 9% reduction in Misses Intent. We surmise the coherent queries in the session knitted by Gen Encoder [36] provides weak supervision that points the model towards more in-topic question suggestions. At the same time, the coherent sessions include many re-issued query paraphrases, misleading BERT to more duplicates.

Inductive weak supervision signals from QA-GEN Sessions provides the largest boost on *usefulness*; adding it improves BERT by 30%+. The more genuine information-seeking user behavior in these sessions guides the BERT model to imitate suggestions that are more forward-looking, less Duplicate or Prequel. On the other hand, QA-GEN sessions may include more pronounced topic drift as reflected by more Misses Intent and Too Specific.

The inductive bias that (weak) supervision signals introduce in a neural system is as important as the network architecture, but is often overlooked. This experiment demonstrates the effectiveness of our inductive weak supervision: without any changes in the model architecture or the number of training labels, and *without any usefulness supervision*, simply adding the weak user imitation signals from more conversational sessions leads to significantly more useful question suggestions.

8.3 Online A/B Test

We served the question suggestions from DeepSuggest in an online A/B test for one week using the standard online experiment setup in Bing. Some of the tracked metrics are shown Table 7.

The strong offline *usefulness* improvements of DeepSuggest are consistent with its impact on real online users. Compared to Online, our method received 8.9% more user clicks when shown near the top of the search result page and 6.4% more when shown on the bottom, both strong improvements in commercial search standard. This result is measured on millions of user feedback signals, and is statistically significant with a p-value of $3e^{-9}$. There is also notable improvement in Overall "Success Rate" of our search engine. Users prefer–and benefit from–useful question suggestions from our inductively trained DeepSuggest.

Table 8: Usefulness results of GPT-2 variations. Three different multi-task approaches are explored: 1) added as auxiliary (Classify) tasks, the same with BERT; 2) as additional generation target, controlled by a functional Delimiter token; 3) the standard multi-task setup. Only the first two label categories are shown; the trend on the rest are similar.

	Useful	Misses Intent
GPT-2	0.253	0.516
+ Context	0.296	0.430
+ Clean Session (Classify)	0.120	0.660
+ GEN Session (Classify)	0.154	0.538
+ QA-GEN Session (Classify)	0.136	0.583
DeepSuggest-NLG (Classify)	0.200	0.549
+ Clean Session (Delimiter)	0.224	0.521
+ GEN Session (Delimiter)	0.214	0.500
+ QA-Gen Session (Delimiter)	0.187	0.545
DeepSuggest-NLG (Delimiter)	0.225	0.509
+ Clean Session (Standard)	0.125	0.665
+ GEN Session (Standard)	0.160	0.610
+ QA-Gen Session (Standard)	0.180	0.680
DeepSuggest-NLG (Standard)	**0.376**	0.412

Table 9: Average perplexity of GPT-2 models on the "positive" item in each task vs. the "negative" (pos/neg). Lower on positive is better. Zero-shot results are marked by *.

	People Also Ask			Session
	Click	R-CTR	Rel	Query
GPT-2	1.82/2.40	*17.3/25.7	*11.2/8.86	*4k/46k
+ Contexts	1.95/2.44	*15.8/24.8	*7.31/5.48	*7k/43k
+ Clean Session	2.03/2.55	*6.91/9.42	*2.72/2.83	13.7/55.0
+ GEN Session	2.03/2.60	*6.48/9.16	*4.19/4.38	12.2/114
+ QA-GEN Session	2.08/2.60	*6.39/8.61	*2.67/2.62	13.3/42.5
DeepSuggest-NLG	1.99/2.48	2.14/4.66	2.77/2.77	11.8/51.6

In offline relevance, DeepSuggest performs on par with Online, while in online A/B test, users strongly prefer more *useful* results from DeepSuggest. This confirms the necessity of going beyond relevance in order to promote more next-gen search experiences.

8.4 Challenges of Generative Suggestion

Compared to the ranking results from BERT, the impact of the generated questions from GPT-2 is more mixed. GPT-2 performs better than BERT in *usefulness* when both are solely trained on the PAA clicks, but it is unable to reconcile the next-turn signals from the search sessions. We further investigate this to have a better understanding of the challenges of NLG in conversational search.

In Table 8 we experiment with two additional modifications to GPT-2 besides the (Standard) multi-task learning setup. The first is to add to GPT-2 the classification MLP layers used in BERT (Classify) for each task. The second is to add a special delimiter token when the generation target is a question as opposed to a query, to control the generation of different text forms (Delimiter) [13].

Adding classification tasks to the generation model significantly hurt GPT-2. Using the question Delimiter works a little better

Table 10: Examples from `Online` (ranking), `DeepSuggest` (ranking), `DeepSuggest-NLG` (generation) and their [usefulness labels].

User Query: "bitcoin price":	
`Online`	
what is the value of bitcoins? [Dup w/ Query]	is it time to buy bitcoin? [Useful]
what was the lowest price of bitcoin? [Useful]	what is the value of 1 bitcoin? [Dup w/ Query]
`DeepSuggest`	
how much does 1 bitcoin cost to buy? [Dup w/ Query]	how to buy bitcoins at walmart? [Too Specific]
what will be the price of bitcoin in 2020 [Useful]	what is the cheapest way to buy bitcoin [Useful]
`DeepSuggest-NLG`	
what is bitcoin cash (bch)? [Miss Intent]	how to buy bitcoin cash? [Miss Intent]
how to buy bitcoin cash (bch)? [Miss Intent]	what is the current bitcoin price? [Dup w/ Query]
User Query: "direct deposit form":	
`Online`	
how to do a direct deposit? [Useful]	what is direct deposit bank of america? [Too Specific]
what is a direct deposit? [Prequel]	how to set up bank of america direct deposit? [Too Specific]
`DeepSuggest`	
what do i need for direct deposit? [Useful]	how to get a chase direct deposit form? [Too Specific]
how to start direct deposit? [Prequel]	how to fill out a direct deposit form? [Useful]
`DeepSuggest-NLG`	
how to set up direct deposit? [Useful]	what is the difference between direct deposit and wire transfer? [Miss Intent]
how do i change my direct deposit on my social security? [Too Specific]	how do i change the bank for my social security deposit? [Miss Intent]

than `Standard`, but `Standard` performs much better when the two remaining tasks are included.

One potential reason is the lack of explicit guidance in *semantics*. In intrinsic evaluation, though GPT-2 fits user clicks well, its relevance (Rel AUC) is nearly random. Compared to ranking models which only need to distinguish the good questions from the rest, generative models are required to produce each question token from scratch while also makes the whole question relevant and useful. This is a much harder task that neither our fine-tuning nor decoding strategy explicitly accounts for.

Another possibility is that the different language styles of PAA questions (natural language) and session queries (mainly keywords) may make it hard for GPT-2 to absorb the underlying "next-turn" information in the search sessions. The perplexity results of `Standard` in Table 9 indicate this possibility: `DeepSuggest-NLG` has much lower perplexity on questions than on queries. (Note that the perplexity numbers are averaged over all the queries and their candidates, which is sensitive to outliers especially for Next Query Prediction.) Controlling the style of generated language and the underlying information it conveys is a significant challenge in natural language generation.

This experiment indicates more work remains to be explored for generative models in conversational search.

8.5 Examples

Table 10 lists some examples from `Online` and `DeepSuggest`, as well as generated questions from `DeepSuggest-NLG`. The `DeepSuggest` results are in general quite on topic, interesting, and "conversation leading'; For example, suggesting the required details for direct deposit and how to fill the form.

On the other hand, the `DeepSuggest-NLG` results are all syntactically correct, mostly on topic, and interesting. We do observe several missed intents or even irrelevant suggestions, showing the challenge of the generative system in capturing user's search intent.

9 CONCLUSION AND FUTURE WORK

This paper studies conversational question suggestion, which facilitates conversational search experiences by proactively suggesting meaningful, informative, and engaging questions. We establish the *usefulness* metric to push conversational search beyond relevance, and to lead the interaction with users via forward-looking suggestions. We also develop a new training regimen to guide models to imitate users' information-seeking trajectories using Gen Encoder mined search sessions; we show this technique leads to more useful suggestions.

We conduct thorough experiments to investigate various aspects of conversational question suggestion. Although all our systems are adept at predicting historical clicks and relevance, only our `DeepSuggest` excels in *usefulness*, showing the advantage of the inductive weak supervision from conversational sessions.

The offline effectiveness of our method and the significance of usefulness are validated by our online A/B test in Bing. Millions of users confirm that they prefer our more useful suggestions: they interact with our conversational question suggestions more often, spend more time reading them, and in general are more satisfied.

Our explorations on generative question suggestion find it challenging for GPT-2 to learn the underlying information from inductive weak supervision signals and may get confused by the superficial language styles. This leads to several important future research directions, for example, on distinguishing the meaning imparted by language from its syntactic forms, on grounding the generations to accurate and desired information, and on new training regimen for language generation.

10 ACKNOWLEDGMENT

We thank Bing PAA team, especially Lokesh Chittoori, for helping with the offline data and online experimental platform. We also thank Hongfei Zhang and Maria Kang for their help with the online A/B tests.

REFERENCES

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. ACM, 475–484.

[2] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. 2004. Query recommendation using query logs in search engines. In *International Conference on Extending Database Technology*. Springer, 588–596.

[3] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2008)*. ACM, 875–883.

[4] Ben Carterette, Evangelos Kanoulas, Mark Hall, and Paul Clough. 2014. Overview of the TREC 2014 session track. In *Proceedings of The 23rd Text Retrieval Conference (TREC 2014)*.

[5] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Reading.

[6] W Bruce Croft and Roger H Thompson. 1987. I3R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science* 38, 6 (1987), 389–404.

[7] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. 985–988.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*. ACL, 4171–4186.

[9] Rand Fishkin. 2019. Less than Half of Google Searches Now Result in a Click. sparktoro.com/blog/less-than-half-of-google-searches-now-result-in-a-click/. (2019). Accessed:2020-01-23.

[10] Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019. Neural approaches to conversational AI. *Foundations and Trends® in Information Retrieval (FnTIR)* 13, 2-3 (2019), 127–298.

[11] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. 2019. Slice: Scalable Linear Extreme Classifiers Trained on 100 Million Labels for Related Searches. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM 2019)*. ACM, 528–536.

[12] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM 2017)*. ACM, 781–789.

[13] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).

[14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*.

[15] Tie-Yan Liu. 2009. Learning to rank for Information Retrieval. *Foundations and Trends in Information Retrieval (FnTIR)* 3, 3 (2009), 225–331.

[16] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. ACL, 4487–4496.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[18] Jiyun Luo, Xuchu Dong, and Hui Yang. 2015. Session search by direct policy learning. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval (ICTIR 2015)*. ACM, 261–270.

[19] Jiyun Luo, Sicong Zhang, and Hui Yang. 2014. Win-win search: Dual-agent stochastic game in session search. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval (SIGIR 2014)*. ACM, 587–596.

[20] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. 1101–1104.

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

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

[23] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACCL 2018)*. ACL, 2227–2237.

[24] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. *arXiv preprint arXiv:1904.07531* (2019).

[25] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019).

[26] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval (CHIIR 2017)*. ACM, 117–126.

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[28] Karthik Raman, Paul Bennett, and Kevyn Collins-Thompson. 2013. Toward Whole-Session Relevance: Exploring Intrinsic Diversity in Web Search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2013)*. ACM.

[29] Karthik Raman, Paul Bennett, and Kevyn Collins-Thompson. 2014. Understanding Intrinsic Diversity in Web Search: Improving Whole-Session Relevance. *ACM Transactions on Information Systems (TOIS)* 32, 4 (2014), 1–45.

[30] Sudha Rao and Hal Daumé III. 2018. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. ACL, 2737–2746.

[31] Sudha Rao and Hal Daumé III. 2019. Answer-based Adversarial Training for Generating Clarification Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*. ACL, 143–155.

[32] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM 2015)*. ACM, 553–562.

[33] Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. 2019. Target-Guided Open-Domain Conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. ACL, 5624–5634.

[34] Bin Wu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Query suggestion with feedback memory network. In *Proceedings of the 2018 World Wide Web Conference (WenConf 2018)*. IW3C2, 1563–1571.

[35] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems (NeurIPS 2019)*. 5754–5764.

[36] Hongfei Zhang, Xia Song, Chenyan Xiong, Corby Rosset, Paul N Bennett, Nick Craswell, and Saurabh Tiwary. 2019. Generic Intent Representation in Web Search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. ACM, 65–74.

[37] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018)*. ACM, 177–186.