Interactive Query Clarification and Refinement via User Simulation

Pierre Erbacher
ISIR - Sorbonne University
Paris, France
pierre.erbacher@isir.upmc.fr

Ludovic Denoyer
Sorbonne University
Paris, France Currently working at Ubisoft
ludovic.denoyer@sorbonne-université.fr

Laure Soulier
ISIR - Sorbonne University
Paris, France
laure.soulier@isir.upmc.fr

ABSTRACT

When users initiate search sessions, their queries are often unclear or might lack of context; this resulting in inefficient document ranking. Multiple approaches have been proposed by the Information Retrieval community to add context and retrieve documents aligned with users' intents. While some work focus on query disambiguation using users' browsing history, a recent line of work proposes to interact with users by asking clarification questions or/and proposing clarification panels. However, these approaches count either a limited number (i.e., 1) of interactions with user or log-based interactions. In this paper, we propose and evaluate a fully simulated query clarification framework allowing multi-turn interactions between IR systems and user agents.

CCS CONCEPTS

• **Information systems → Users and interactive retrieval**.

KEYWORDS

Interactive Information Retrieval, Query Clarification, Simulation

1 INTRODUCTION

Understanding information need is a long-standing issue in Information Retrieval (IR) [10, 17, 34], often highlighted by the difficulty for users to formulate open-ended information needs into queries. Queries are thus often under-specified and/or ambiguous [17]. Depending on the user or the context, the same query may refer to different intents. For instance, the query "Orange" may refer to several topics, including company names, locations, songs titles, ... To tackle this issue, numerous works have been proposed. One first line of work relies on query reformulation [3, 21, 33, 44] where the objective is to rewrite the query. A lot of effort has been provided by designing models based on either (pseudo-)relevance feedback [3, 21, 33] or external knowledge resources [44]. But recently, the advances in machine translation models and in large language models have turned this task into a query generation task [11, 13]. Another category of work focuses on search/query diversification [1, 6, 8, 23, 25] to increase the query coverage, particularly when the query is multi-faceted. While early search result diversification work have designed or derived the Maximal Margin Relevance (MMR) model [8], other techniques such as document clustering [1] or document-driven voting scheme [12] has been used to retrieve a diversified list of documents. Recently, MacAvaney et al. [23] have proposed to focus on query diversification by generating queries by designing a Distributional Causal Language Modeling. However, for all these diversification techniques, the issued document list might include some top-ranked documents that do not match with the user's intent [37]. This highlights the need to clarify users' queries before retrieving documents. A last category of work aims to leverage search history to infer user's profile or session context, with the objective to ground the initial query [4, 15, 20, 24, 39]. By encoding short-term behavior or long-term behavior as features of the ranking model, these methods manage to retrieve more personalized documents. But, these approaches might be limited by the user's behavior's variability over time [42]. In parallel, neural ranking models, such as [16, 19], have increased their performance due to the fine-grained capture of queries and documents semantics. Nevertheless, the query ambiguity is often inherent to users, increasing the need to place the user at the center of the IR process.

A promising approach has been proposed in [2] to clarify information needs by proactively interacting with the user. The authors propose a conversation framework that consists in generating clarifying questions when the query is ambiguous. Clarifying questions might be query reformulation (e.g., "Would you like to know how to care for your dog during heat?" for the initial query "dog heat" as in [2]) or questions with possible options (e.g., "what do you want to know about this british mathematician? Options: movie, suicide note, quotes, biography" for the initial query "alan turing" as in [41]). With this in mind, the classic workflow for asking clarifications is based on three main steps [2]: 1) the IR system produces a clarifying question for the user, 2) the latter provides an answer or selects an option, and 3) the IR system ranks documents according to the user's feedback. The pioneering work [2] aims at generating clarifying questions by 1) retrieving a predefined set of questions using a Bert-based model and 2) at each turn, selecting the best query through a conversation history-driven model. The user's answer corresponds to a predefined text built using crowdsourcing. One drawback of this approach is that the multi-turn conversation is log-based, interactively simulated using predefined logs of conversation history (i.e., sequence of questions/answers obtained by HITS). This simulated conversation defined a priori without interaction with the proposed question selection model might hinder the evaluation performance in the sense that we are not sure about the soundness of the conversation flow. Other work [14, 28, 35, 38, 41] tackle this issue by proposing generative models, that create clarification question or query suggestions. But they do not address the multi-turn framework, stopping the clarification process at the first interaction.

In this paper, we propose to build a fully simulated query clarification framework allowing multi-turn interactions between IR

and user agents. Following [2], the IR agent identifies candidate queries and ranks them in the context of the user-system interactions to clarify the initial query issued by the user agent. We particularly target simple information needs, multiple information needs are let for future work since it might impact the modeling of the query ranking function. Our framework can be seen as a basis and a proof-of-concept for future work willing to integrate sequential models (namely reinforcement learning models) for question clarification. It is worth noting that large language models relying on attention mechanisms (transformers) are not yet well suited to handle sequential interactions and long-term planning, as current models are hardly trainable with current reinforcement learning algorithms [9]. Therefore, all agent components in our framework are based on continuous and simple models. To validate our simulation framework, we conduct an experimental analysis on the MS Marco dataset. We show the benefit of multi-turn interactivity and evaluate the effectiveness of different question selection strategies.

2 QUESTION CLARIFICATION SIMULATION FRAMEWORK

2.1 Overview and Research Hypotheses

Our query clarification simulation framework is inspired from [2], but provides the possibility of leveraging user-system agents' interactions sequentially. More particularly, our framework is illustrated in Figure 1 and relies on the following workflow:

- A) The user issues an initial query q_0 associated to her/his information need i to the IR system.
- B) The IR system generates a set $Q = \{q_1, q_2, ..., q_m\}$ of candidate queries which might express different query reformulations or diversified queries to better explore the information need i.
- C) The IR system selects N queries to display to the user. To do so, we propose to follow [2] and design a model ranking the candidate query set Q to identify the top N queries.
- D) The user selects one of the N queries, enabling to extract positive and negative feedback, resp. noted (q^+, q^-).
- Steps C) and D) can be repeated several times to model multi-turn interactions. The query set ranking function (step C) integrates the user's sequential feedback (q^+, q^-) to improve the query ranking along with the interaction simulation.
- E) After T turns, the IR system considers the best ranked query as the optimal query reformulation and runs a ranking model to retrieve documents.

The design of this evaluation framework is guided by some choices/research hypothesis.

• First, following [2], we consider a fixed set of candidate queries $Q = \{q_1, q_2, ..., q_m\}$ constituting the reformulation of the initial query q_0. All the interactions are leveraged to improve step by step the ranking of this candidate query set so as, at the end of the session, the final query used for retrieving documents is a good clarification of its initial one. Obviously, this means that the set of candidate queries includes a large variety of queries which, for some of them, improve the search performance.

• Second, following [41], we propose to model question clarification as a possible option between two reformulated queries. In other words, expressed in natural language, the IR system agent would ask the user agent the following question: "Which reformulated query do you prefer? A or B". This implies that the user is willing to judge queries A or B regarding its information need.

• Third, guided by the motivation to propose a framework for future work on sequential models, we consider here that each agent component is modeled at the embedding level. Indeed, leveraging large language models for generating/ranking questions is very effective, but integrating them into reinforcement learning models is still challenging (one main reason being the computational cost). This means that we processed *a priori* all queries and documents to represent them using text embeddings. This processing is done offline, alleviating the sequential modeling of the text encoding.

In what follows, we present the different components behind the IR system and user agents.

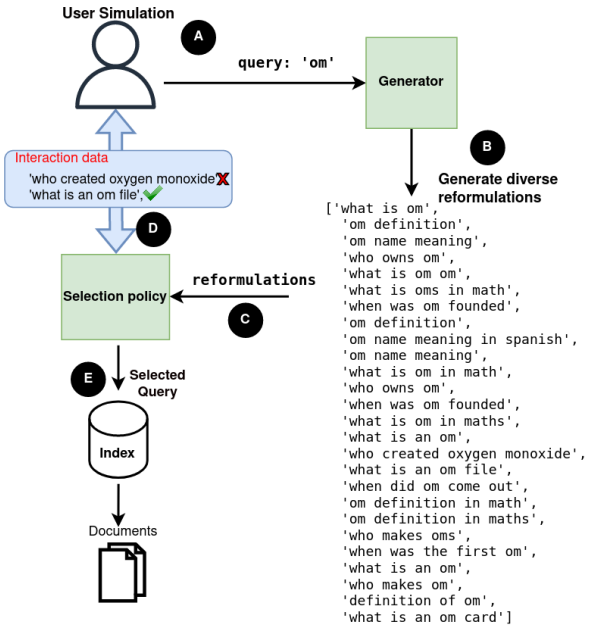

Figure 1: Query clarification simulation framework

2.2 The IR System Agent

The IR agent has three objectives in our framework: 1) generating the set of candidate reformulated queries willing to be presented to the user, 2) ranking this set to identify the most relevant queries according to the interaction history, 3) ranking documents using the best-ranked query (ending the interactive session).

Generation of the candidate reformulated query set. The objective here is to instantiate various and diverse reformulation covering a wide range of relevant topics for the initial query q_0. Different techniques might be used, leveraging large language models [27, 31, 32], query diversification [6, 23, 40] or query expansion [29]. We propose here to use the T5 model [31] which is designed to translate token sequences into other token sequences. It has already been used for query reformulation tasks, demonstrating its ability for our approach [9, 22, 31]. On the top of that model, the generation process is driven by beam diversity [36] which aims at generating a set Q of diversified query reformulation, $Q = \{q_1, q_2, ..., q_m\}$.

Ranking of queries based on the interaction history. The role of the selection policy is to select queries used to interact with the user agent. Following [2] which proposes to rank queries according to both performance criteria and the interaction context, we use a conditional ranker [5] which computes a pairwise score between two candidates queries given the context, namely the initial query q_0 and the additional information provided by interaction with the user. Let q_i and q_j be the candidate queries with their supervised effectiveness scores, resp. y_i, y_j. The ranking model relies on:

$$P(y_i > y_j | q, q_i, q_j, feedback_{t-1}, ..., feedback_1) \quad (1)$$

For sake of simplicity, we assume that each query (initial or candidate) are represented through text embeddings. In the following, q refers to query embedding and d to document embedding.

In practice, the ranking model estimates a score for each query q_i and q_j given all the context, $\{q, q_i, q_j, feedback_{t-1}, ..., feedback_1\}$ and then compare these scores to identify which one is the most relevant. $feedback_t$ corresponds to selected or not selected queries (resp. q^+ and q^-) by the user agent at interaction turn t. These queries are concatenated as follows: $feedback_t = (q^+, q^-)$ and feedback overall interaction turns are aggregated, the whole process using a Hierarchical Recurrent neural network (RNN) to encode at the interaction level and also the sequence of interactions. Note that queries q^+ and q^- are encoded differently using resp. a *cosin* and *sin* function. Moreover, we do not encode the position in which each query is presented to the user agent, as this latter does not have position bias on the clarification query selection.

Final ranking of documents. Documents are retrieved with the top ranked query using a Dense Retriever model [16].

2.3 The User Agent

After issuing the initial query q_0, the user agent interacts with the IR system agent to refine her/his information need. With this in mind, we hypothesize that the user is greedy toward her/his intent and fully cooperative. Therefore, he always selects the preferred query as the most similar to the intent. Despite being unrealistic, we ignore the click bias problem for clarification panel presented in [42, 43] (as mentioned earlier). Other choices for user simulation could be done, following [7], but we let these variations for future work.

In practice, let d be a user intent, q_i and q_j the clarification queries presented to the user agent. The user agent selects the best query (noted q^+ for highlighting positive feedback from the user) according to a similarity metric (in our case, the dot product) between the proposed queries q_i and q_j and intent d:

$$q^+ = argmax_{q_i}(\langle q_i, d \rangle) \quad (2)$$

The non-selected query, q^-, expresses negative feedback.

3 EVALUATION PROTOCOL

Evaluating our simulation framework consists in measuring the effectiveness of the final ranking after T clarification interactions. Since the user behavior is greedy and follows a simple behavior dependent on the query selection process, the effectiveness results mainly denote the quality of this latter component. Other components (candidate set generation and final document ranking) do not depend on the interaction feedback, so we mainly focus on understanding whether the selection policy integrates users' feedback and takes good decisions to select the N clarification questions. For reproducibility, the code of our simulation framework and the evaluated baselines/scenarios will be released upon acceptance.

3.1 Dataset

We carry out our experiments on MS Marco 2020 passages which regroups 8.8M passages and more than 500K Query-Passage relevance pairs. Following [26], we evaluate our model on 2 sets. The small test set (43 queries) and a subset of the dev set (1000 queries sampled from 59 000). One motivation to consider these two datasets is their difficulty level: in the dev set, only one passage per query is labeled relevant in the ground truth, while several passages are considered as relevant in the test set.

3.2 Baselines and Scenarios

To evaluate the effectiveness of our selection policy component, we compare with:

(1) Non-interactive settings to show the gain of interacting with users. We measure the ranking effectiveness of the user's initial user query (noted **User Query**), the **Best Reformulation** in the candidate query set - which can be seen as an oracle, and the **MonoT5** Documents re-ranker which acts as a strong ranking baseline [30]. This model is a pointwise ranking, estimating relevance scores for query/document pairs. This model relies on a large sequence to sequence language model pretrained on various task [31]. Please note that using this model for the selection policy, and therefore integrating user's feedback, is not obvious since this is a seq2seq pointwise model, labels associated with queries are binary (relevant or not) and has to be grounded relative to a value. For that reason, we only consider its non-interactive scenario.

(2) **Naive interactive selection**: At each step, we select the 2 top ranked queries from the current query rank and then remove the query which has not been selected by the user agent. The re-ranking of the candidate query set is only carried out once, at the beginning of the session, and the size of this list decreases with the interaction number.

To instantiate the selection policy after each interaction-driven query ranking step (step C in Figure 1), we consider these scenarios:

(1) **Interact. + Random Sampl**: we sample 2 queries from the ranked candidate query set to constitute the interaction pair.

(2) **Interact. + Top 2**: we select the top 2 query reformulations at each turn.

(3) **Interact. + random sampl@5**: we randomly select 2 queries among the top 5 query reformulations at each turn.

(4) **Interact. + Kmeans selection**: At each turn, queries in the candidate set are clustered in 2 groups using Kmeans. Queries from each cluster are ranked by the model. The best-ranked query from each cluster is then selected for interaction with the user. The cluster of the query not selected by the user is removed for the next turn from the set of candidate queries. This strategy corresponds to a refinement strategy, removing a group of semantically similar queries that have not been chosen by the user and going deeper in the other cluster.

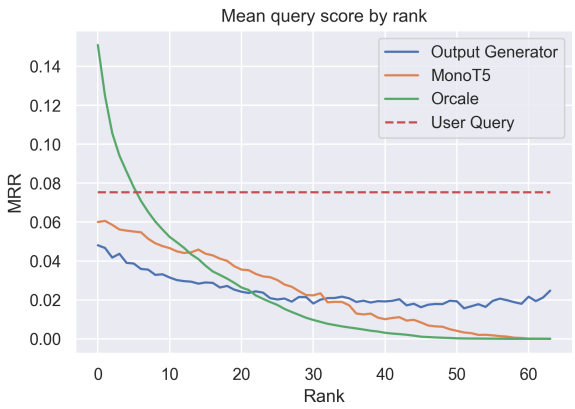

Figure 2: Effectiveness score of query reformulation by rank.

3.3 Model Implementation

All queries and passages embeddings are pre-computed using the Dense Retriever proposed by [16]. Embeddings are stored and indexed using faiss HSWN32 index [18]. The candidate query set is generated by diversity beam with a group penalty equal to 0.6. The size of the candidate set is of 64. The number of queries displayed to the user agent is set to $N = 2$. For the model hyperparameters, we use batches of size 128, the optimizer is Adam ($\beta_1 = 0.9, \beta_2 = 0.99$) with weight decay ($= 0.01$). We use batch normalization and dropout ($p = 0.3$) between each layer. The learning rate is set to 1×10^{-4}.

4 RESULTS

4.1 Preliminary Analysis

We present here a preliminary analysis to quantify the potential retrieval performance gain of the candidate query set within the question clarification step. To do so, we compare the performance of different query rankings: 1) the candidate query generated by the T5 model ranked by decreasing order of likelihood resulting from the Diversity Beam search (without application of our ranking function); 2) the Oracle corresponding to the candidate query set ranked in a decreasing order according to their performance according to Mean Marginal Rank metric in the ground truth. We emphasized that this Oracle curve shows the performance of our T5 model at generating search oriented reformulations. 3) The MonoT5 ranking corresponding to candidate queries re-ranked by MonoT5. Figure 2 illustrates the performance of queries depending on their rank in the different mentioned lists. From Figure 2, we can see that ranking queries with MonoT5 allows to improve the performance for the top k queries (MonoT5 vs. Output Generator). This has a negative effect for the end of the list, but it is not critical in our case, since we consider selection policy regarding the top query list. Moreover, one can notice that, although performance are increased, there is still a gap between the curve of the MonoT5 ranked list and the Oracle curve. Our intuition is that leveraging users' interactions will lower this gap, which leads to the evaluation we performed in what follows.

		No interaction	1	2	3	4	5
User Query	mrr@10	0.4554	-	-	-	-	-
	map@10	0.3382	-	-	-	-	-
Best Reformulation	mrr@10	0.8720	-	-	-	-	-
	map@10	0.5646	-	-	-	-	-
Monot5 (query ranker)	mrr@10	0.4713	-	-	-	-	-
	map@10	0.3209	-	-	-	-	-
Naive selection	mrr@10	0.2135	0.3270	0.3597	0.4036	0.4191	0.4271
	map@10	0.1222	0.1943	0.2205	0.2553	0.2688	0.2766
Interact. + random sampl	mrr@10	0.4031	0.4786	0.4814	0.4903	0.4814	0.5019
	map@10	0.2531	0.3344	0.3413	0.3529	0.3480	0.3685
Interact. + Top 2	mrr@10	0.4031	0.4746	0.4693	0.4903	0.4786	0.5019
	map@10	0.2531	0.3294	0.3436	0.3520	0.3471	0.3428
Interact. + random sampl@5	mrr@10	0.4031	0.4734	0.4670	0.4903	0.4798	0.5019
	map@10	0.2531	0.3287	0.3420	0.3517	0.3469	0.3451
Interact. + Kmean	mrr@10	0.4031	0.5232	0.4658	0.4692	0.4863	0.5515
	map@10	0.2531	0.3706	0.3207	0.3402	0.3181	0.3347

Table 1: Effectiveness results on the Test set of MS Marco passage 2020 (43 queries - multiple relevant documents per query)

		No interaction	1	2	3	4	5
User Query	mrr@10	0.2090	-	-	-	-	-
Best Reformulation	mrr@10	0.4119	-	-	-	-	-
Monot5 (query ranker)	mrr@10	0.1557	-	-	-	-	-
Naive selection	mrr@10	0.1228	0.1513	0.1659	0.1767	0.1866	0.1911
Interact. + random sampl	mrr@10	0.1719	0.2012	0.1990	0.1954	0.2003	0.2016
Interact. + Top 2	mrr@10	0.1719	0.2020	0.1987	0.1973	0.2017	0.1990
Interact. + random sampl@5	mrr@10	0.1719	0.2020	0.1983	0.1966	0.2007	0.2008
Interact. + Kmean	mrr@10	0.1719	0.1748	0.1984	0.2016	0.2158	0.2224

Table 2: Effectiveness results on the subset of MS Marco passage 2020 dev set (1000 queries - 1 relevant document per query)

4.2 Effectiveness Results

We analyze here the performance of the query ranker at different interaction turns using mmr@10 and map@10. Tables 1 and 2 resp. show the results on the MS Marco passage 2020 test set and dev set. From a general point of view, we can see that performance metrics are lower for the dev set than for the test set. This can be explained by the task difficulty, which is higher for the dev set in which only one document per query is assessed as relevant. By comparing all baselines and scenarios, we can outline the following trends. 1) The first candidate query ranking within our interactive models (No interaction columns) provides lower performance than non-interactive baselines. For instance, the **Interact. + Top2** scenario observes a decrease of 12% in terms of mrr@10 for the test set w.r.t. the initial user query. 2) But this trend is reversed with each interaction turn to obtain for certain scenarios performance higher than baseline ones (see all interaction models in the test set, and the **Interact + Kmeans** for the dev set). 3) The interaction model with Kmean strategy looks to be the best selection policy for question clarification since it obtains the highest mrr@10 for both datasets. This is somehow intuitive because this strategy might correspond to a refinement strategy, going deeper and deeper into clusters. This is also connected with the dataset peculiarity since MS Marco is mainly composed mono-faceted questions in natural language.

5 CONCLUSION AND PERSPECTIVES

This exploratory work focuses on sequential click-based interaction with a user simulation for clarifying queries. We provide a simple and easily reproducible framework simulating multi-turn interactions between a user and a IR system agent. The advantage of our framework is the simplicity of interaction, as there is no need for dataset of real and annotated user-system interactions. Experiments highlight performance gain in terms of document retrieval through the multi-turn query clarification process and provide a

comparative analysis of selection strategies. The next steps for this work are: 1) leveraging reinforcement learning for the selection policy. 2) test more diverse and more sophisticated user simulation, as done in [7] for multi-faceted information needs.

6 ACKNOWLEDGMENTS

We would like to thank the ANR JCJC project SESAMS (Projet-ANR-18-CE23-0001) for supporting Pierre Erbacher and Laure Soulier from Sorbonne Univeristé in this work.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying Search Results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining* (Barcelona, Spain) *(WSDM '09)*. Association for Computing Machinery, New York, NY, USA, 5–14. https://doi.org/10.1145/1498759.1498766

[2] 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* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 475–484. https://doi.org/10.1145/3331184.3331265

[3] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. *ACM Trans. Inf. Syst.* 20, 4 (oct 2002), 357–389. https://doi.org/10.1145/582415.582416

[4] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the impact of short- and long-term behavior on search personalization. In *SIGIR '12*.

[5] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to Rank Using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning* (Bonn, Germany) *(ICML '05)*. Association for Computing Machinery, New York, NY, USA, 89–96. https://doi.org/10.1145/1102351.1102363

[6] Fei Cai, Ridho Reinanda, and Maarten De Rijke. 2016. Diversifying Query Auto-Completion. *ACM Trans. Inf. Syst.* 34, 4, Article 25 (jun 2016), 33 pages. https://doi.org/10.1145/2910579

[7] Arthur Câmara, David Maxwell, and Claudia Hauff. 2022. Searching, Learning, and Subtopic Ordering: A Simulation-based Analysis. *CoRR* abs/2201.11181 (2022). arXiv:2201.11181 https://arxiv.org/abs/2201.11181

[8] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 335–336. https://doi.org/10.1145/290941.291025

[9] Jerry Zikun Chen, Shih Yuan Yu, and Haoran Wang. 2020. Exploring Fluent Query Reformulations with Text-to-Text Transformers and Reinforcement Learning. *ArXiv* abs/2012.10033 (2020).

[10] Steve Cronen-Townsend and W. Bruce Croft. 2002. Quantifying Query Ambiguity. In *Proceedings of the Second International Conference on Human Language Technology Research* (San Diego, California) *(HLT '02)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 104–109.

[11] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. CAsT 2020: The Conversational Assistance Track Overview. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020 (NIST Special Publication, Vol. 1266)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.C.pdf

[12] Van Dang and W. Bruce Croft. 2012. Diversity by Proportionality: An Election-Based Approach to Search Result Diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Portland, Oregon, USA) *(SIGIR '12)*. Association for Computing Machinery, New York, NY, USA, 65–74. https://doi.org/10.1145/2348283.2348296

[13] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 5918–5924. https://doi.org/10.18653/v1/D19-1605

[14] J. Guo, Xueqi Cheng, Gu Xu, and Xiaofei Zhu. 2011. Intent-aware query similarity. In *CIKM '11*.

[15] Morgan Harvey, Fabio A. Crestani, and Mark James Carman. 2013. Building user profiles from topic models for personalised search. *Proceedings of the 22nd ACM international conference on Information & Knowledge Management* (2013).

[16] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. *Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling*. Association for Computing Machinery, New York, NY, USA, 113–122. https://doi.org/10.1145/3404835.3462891

[17] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. 2000. Real life, real users, and real needs: A study and analysis of user queries on the Web. *Information Processing and Management* 36, 2 (1 March 2000), 207–227. https://doi.org/10.1016/S0306-4573(99)00056-4

[18] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2021), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572

[19] Omar Khattab and Matei Zaharia. 2020. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/3397271.3401075

[20] Weize Kong, Rui Li, Jie Luo, Aston Zhang, Yi Chang, and James Allan. 2015. Predicting Search Intent Based on Pre-Search Context. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 503–512. https://doi.org/10.1145/2766462.2767757

[21] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, Louisiana, USA) *(SIGIR '01)*. Association for Computing Machinery, New York, NY, USA, 120–127. https://doi.org/10.1145/383952.383972

[22] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy J. Lin. 2020. TREC 2020 Notebook: CAsT Track. In *TREC*.

[23] Sean MacAvaney, Craig Macdonald, Roderick Murray-Smith, and Iadh Ounis. 2021. IntenT5: Search Result Diversification using Causal Language Models. *CoRR* abs/2108.04026 (2021). arXiv:2108.04026 https://arxiv.org/abs/2108.04026

[24] Nicolaas Matthijs and Filip Radlinski. 2011. Personalizing Web Search Using Long Term Browsing History *(WSDM '11)*. Association for Computing Machinery, New York, NY, USA, 25–34. https://doi.org/10.1145/1935826.1935840

[25] Rodrigo Nogueira, Jannis Bulian, and Massimiliano Ciaramita. 2019. Multi-agent query reformulation: Challenges and the role of diversity. In *DeepRLStructPred@ICLR*.

[26] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy J. Lin. 2019. Multi-Stage Document Ranking with BERT. *ArXiv* abs/1910.14424 (2019).

[27] Rodrigo Nogueira, Wei Yang, Jimmy J. Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *ArXiv* abs/1904.08375 (2019).

[28] Umut Ozertem, Olivier Chapelle, Pinar Donmez, and Emre Velipasaoglu. 2012. Learning to Suggest: A Machine Learning Framework for Ranking Query Suggestions. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Portland, Oregon, USA) *(SIGIR '12)*. Association for Computing Machinery, New York, NY, USA, 25–34. https://doi.org/10.1145/2348283.2348290

[29] Dipasree Pal, Mandar Mitra, and Kalyankumar Datta. 2013. Query Expansion Using Term Distribution and Term Association. *CoRR* abs/1303.0667 (2013). arXiv:1303.0667 http://arxiv.org/abs/1303.0667

[30] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *CoRR* abs/2101.05667 (2021). arXiv:2101.05667 https://arxiv.org/abs/2101.05667

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[32] 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-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 143–155.

[33] Joseph John Rocchio. 1971. Relevance feedback in information retrieval. *Gerard Salton, editor, The SMART Retrieval System - Experiments in Automatic Document Processing* (1971), 313–323.

[34] Mark Sanderson. 2008. Ambiguous Queries: Test Collections Need More Sense. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Singapore, Singapore) *(SIGIR '08)*. Association for Computing Machinery, New York, NY, USA, 499–506. https://doi.org/10.1145/1390334.1390420

[35] Rodrygo L. T. Santos, Craig MacDonald, and Iadh Ounis. 2012. Learning to rank query suggestions for adhoc and diversity search. *Information Retrieval* 16 (2012), 429–451.

[36] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models. *CoRR* abs/1610.02424

(2016). arXiv:1610.02424 http://arxiv.org/abs/1610.02424

[37] Jun Wang and Jianhan Zhu. 2009. Portfolio Theory of Information Retrieval. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Boston, MA, USA) *(SIGIR '09)*. Association for Computing Machinery, New York, NY, USA, 115–122. https://doi.org/10.1145/1571941.1571963

[38] Bin Wu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Query Suggestion with Feedback Memory Network. *Proceedings of the 2018 World Wide Web Conference* (2018).

[39] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. 2010. Context-aware ranking in web search. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (2010).

[40] Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2Set: Generating Diverse Keyphrases as a Set. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 4598–4608. https://doi.org/10.18653/v1/2021.acl-long.354

[41] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. *Generating Clarifying Questions for Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 418–428. https://doi.org/10.1145/3366423.3380126

[42] Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N. Bennett, Nick Craswell, and Susan T. Dumais. 2020. Analyzing and Learning from User Interactions for Search Clarification. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1181–1190. https://doi.org/10.1145/3397271.3401160

[43] Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N. Bennett, Nick Craswell, and Susan T. Dumais. 2020. Analyzing and Learning from User Interactions for Search Clarification. *CoRR* abs/2006.00166 (2020). arXiv:2006.00166 https://arxiv.org/abs/2006.00166

[44] Ingrid Zukerman and Bhavani Raskutti. 2002. Lexical Query Paraphrasing for Document Retrieval. In *COLING 2002: The 19th International Conference on Computational Linguistics*. https://aclanthology.org/C02-1161