The University of Stavanger (IAI) at the TREC 2022 Conversational Assistance Track

Weronika Lajewska[†], Nolwenn Bernard[†], Ivica Kostric[†], Ivan Sekulić[‡], and Krisztian Balog[†]

[†]University of Stavanger, Stavanger, Norway
[‡]Università della Svizzera italiana, Lugano, Switzerland
[†]{weronika.lajewska, nolwenn.m.bernard, ivica.kostric, krisztian.balog}@uis.no, [‡]ivan.sekulic@usi.ch

Abstract. This paper describes the participation of the IAI group at the University of Stavanger in the TREC 2022 Conversational Assistance track. We employ an established two-stage passage ranking architecture, i.e., first-pass passage retrieval (with standard BM25 ranking and pseudo-relevance feedback) followed by re-ranking (with mono and duo T5) using a rewritten query (with a T5 model fine-tuned on the CANARD dataset). In our runs, we experiment with intent classification based on MSDialog-Intent and term expansion using beam search scores for query rewriting as well as with clarifying questions for the mixed-initiative subtask.

Keywords: Conversational AI, conversational search, TREC CAsT

1 Introduction

The Conversational Assistance track (CAsT) has been a part of the Text REtrieval Conference (TREC) since 2019. It aims to advance conversational information seeking research and to create a large-scale reusable test collection for open-domain conversational search [6]. The track addresses conversational search that is about to satisfy a "user's information need expressed or formalized through multiple turns in a conversation" [6]. The topics are complex, diverse, and answerable. They require content from multiple information sources and vary in types of conversational structural patterns. The main differences in TREC CAsT'22 are the focus on fluent responses that contain only relevant information (summaries), multiple information needs in a shared topic, and a mixed-initiative subtask.

The focus of our participation is to investigate different query rewriting approaches in order to improve end-to-end system performance. We follow an established two-stage retrieve-then-rerank pipeline architecture, i.e., first-pass passage retrieval followed by re-ranking. We use sparse retrieval with BM25 on queries expanded with pseudo-relevance feedback. For the second step, we use a pointwise monoT5 re-ranker followed by a pairwise duoT5 re-ranker. As a baseline for the query rewriting module, we use a T5 model fine-tuned on the CANARD dataset. Our experiments on CAsT'21 suggest that this is a strong baseline.

2 Related work

What makes conversational search, in the TREC CAsT sense, different from passage retrieval, and truly conversational, is the element of query rewriting. Approaches to query rewriting can be categorized into three main groups: feature-based unsupervised approaches [11], feature-based supervised approaches [13], and supervised neural approaches. The last group is characterized by the utilization of large pre-trained language models. In particular, generative models such as GPT-2 [22] or T5 [5, 8, 25] are used. They are mostly fine-tuned on the CANARD dataset [5, 8, 11, 21, 22] or on QReCC [25]. In terms of end-to-end conversational search performance, systems using a combination of term-based query expansion with generative models for query reformulation are shown to provide the best results [10, 11, 22]. Mele et al. [14] propose a flexible query rewriting method based on the classification of utterances. That is, the strategy used to define the context for the rewrite depends on the utterance class (self-explanatory, referring to the first, or referring to the previous topic in the conversation). In our runs, we experiment with a T5 generative language model, BERT-based intent classification, and term expansion using beam search scores.

The majority of conversational search systems use a two-stage retrieve-then-rerank architecture [13, 21, 22, 26]. Approaches to first-pass retrieval can be categorized into two main groups: sparse retrieval and dense-sparse retrieval. Sparse retrieval models include BM25 [11, 21, 22, 26], query likelihood with Dirichlet smoothing [10, 13, 23], or sequential dependency model [7]. The second group of first-pass retrieval methods takes advantage of dense embedding spaces, e.g., ANCE [24, 25]. Approaches to the re-ranking of candidates retrieved in the first stage are based on large, computationally expensive transformers models. These models are either based on BERT [10, 11, 21–23, 26] or on T5 [5, 8, 25]. Re-ranking is often performed in pointwise and pairwise fashion as proposed in [16]. All our runs use this standard two-stage pipeline with BM25 combined with pseudo-relevance feedback for first-pass retrieval and a pointwise/pairwise T5 re-ranker.

The mixed-initiative paradigm of conversational search offers the possibility for the system to ask clarifying questions at any point of the interaction [3]. Asking clarifying questions is proven to be beneficial to conversational search, as the system can provide the user with more relevant results after receiving the answer [1]. Approaches to asking clarifying questions range from selecting the appropriate question from the pre-defined pool of questions [1, 2, 18] to clarifying question generation [19, 27]. Our mixed-initiative submission uses clarifying questions selected from the candidate pool based on question-query semantic similarity to extend the current query.

3 Approach

We briefly cover the approaches used in implementing different components of the system: indexer, query rewriter, first-pass retriever, and re-ranker. The

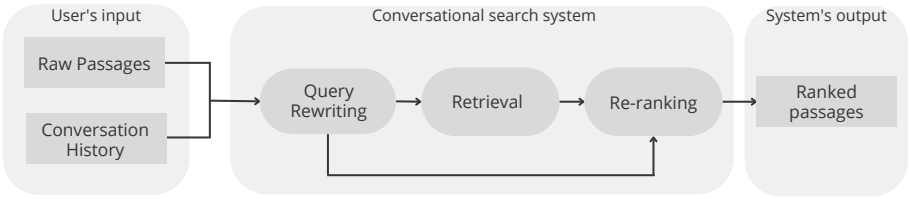

Fig. 1. Architecture of our retrieve-then-rerank pipeline with a query rewriting module.

schema of the architecture used is presented in Figure 1. We focus on the main task in Sections 3.1–3.4 and on the mixed-initiative subtask in Section 3.5. Specific runs are discussed in Section 4.

3.1 Indexing

The default index used by our baseline [9] is based on Elasticsearch (v7.13.3). Each passage consists of three fields: `title`, `body`, and `catch_all`, which is a concatenation of the two. The passages are preprocessed using the Elasticsearch built-in analyzer which is responsible for tokenization (removing punctuation symbols and dividing passages to terms using the Unicode Text Segmentation algorithm), stopwords removal (based on English corpus from the NLTK toolkit), and stemming (KStem).

3.2 Query Rewriting

Baseline The default query rewriter in our baseline is based on a T5 model fine-tuned with the CANARD dataset. We use a trained model shared on Hugging Face (`castorini/t5-base-canard`[1]). Our implementation is based on the Hugging Face transformers library.[2] All the previous rewritten utterances and the canonical response for the last utterance are used as context to reformulate the current question. If the length of the input sentence exceeds 512, the answer passage is cut off.

Sparse Query Rewriter Using a fine-tuned T5 model, we generate the top 10 query rewrites according to their beam search score. Each rewrite is split into terms which are stored together with the beam search score in a weighted bag-of-words (BOW) fashion. For each unique term in the obtained collection of terms, the term weights are summed up from all rewrites. The collection of unique terms and weights is first normalized such that all weights sum up to 1. Then, it is converted into a compound Elasticsearch query consisting of weighted term queries. Effectively, this acts both as a query rewriting and a query expansion

[1] https://huggingface.co/castorini/t5-base-canard
[2] https://github.com/huggingface/transformers

Table 1. Illustration of intent classification-based query rewriting (topic 132). Intents are placed at the end of each sentence (CQ_FQ: clarifying/follow up question, FQ_IR: follow up question, information request).

Turn	Raw query	Rewritten query
2-5	How are developed countries helping with that? [CQ_FQ]	How are developed countries helping with climate change adaptation?
2-7	Are they meeting them? [FQ_IR]	Are developed countries meeting the Paris Agreement?
2-9	That's not too relevant to my question. [NF] By the way, is that related to last year's conference? [FQ_IR]	By the way, is COP26 related to last year's conference?

method. The highest scoring rewrite is the query rewrite, while the other rewrites determine the term-query importance and further extend the rewritten query with additional terms obtained from the context (i.e., the conversation history). The normalization step is needed so that we can combine the results with that of other query modeling approaches, e.g., pseudo-relevance feedback (PRF). The terms obtained by PRF can be linearly interpolated with the weighted BOW terms.

Intent Classification We train a BERT intent classifier using the MSDialog-Intent dataset [17], which has 12 intent classes. For each query, the intent classification is performed at the sentence level. We identify 4 classes: Junk (JK), Greetings/Gratitude (GG), Negative Feedback (NF), and Positive Feedback (PF); we assume that these intents can introduce some noise in the query and thus negatively impact its rewrite. For example, in the query "Okay. Are there better calorie-burning alternatives to try?" the classifier returns the label PF for the first sentence and FQ (Follow Up Question) for the second. We believe that the first sentence (i.e., "Okay") will not provide useful context to perform the rewrite. Therefore, the idea is to remove the sentences with these classes in the case of multi-sentence queries before performing the query rewrite. Furthermore, if negative feedback is detected, we replace the last rewritten query with the raw query in the context based on the assumption that it can reduce the propagation of potential noise from the last query rewrite. Table 1 presents an example where negative feedback is detected in the first sentence of turn 2-9. Therefore, the last query in the context is the raw query instead of the rewritten query of the turn 2-7. In this example, one can think that *they* and *them* from the raw query in turn 2-7 do not correspond to *developed countries* and *Paris Agreement* in the rewritten query of the same turn.

3.3 First-pass Retrieval

Baseline We rank passages in first-pass retrieval using BM25 with the parameters tuned on 2020 and 2021 CAsT datasets ($k1 = 0.95$, $b = 0.45$) using the `catch_all` index field. The top 1000 candidates for each turn are selected for re-ranking.

Pseudo-Relevance Feedback Optionally, pseudo-relevance feedback (PRF) can be added to the pipeline for query expansion. To generate an expanded query, we use the popular RM3 method. RM3 extends the initial query with the highest-weighting terms from top-k scoring documents (we use $k = 10$ and the number of terms $m = 10$). The term weights are normalized such that weights of all terms to be added sum to 1 and linearly interpolated with the original query terms.

3.4 Passage Re-ranking

Baseline The passages retrieved during first-pass retrieval are reranked with a neural re-ranker. Our baseline re-ranker is a pointwise monoT5 re-ranker, followed by a pairwise duoT5 re-ranker [16]. We use the monoT5 model introduced by Nogueira et al. [15], published on Hugging Face (`castorini/monot5-base-msmarco`[3]). The tokenizer associated with this model is used for encoding the query-passage pairs for the input. The top 50 passages from monoT5 are passed for re-ranking with duoT5. Our implementation of duoT5 is based on the Hugging Face transformers library and the `castorini/duot5-base-msmarco` model published on Hugging Face.[4]

3.5 Mixed-initiative

Our approach to the mixed-initiative sub-task consists of two modules: clarifying question selection and answer processing.

Clarifying question selection. We select clarifying questions to ask from the candidate pool of questions provided by the organizers. In order to ensure that the selected questions are of high quality, we first filter out potentially misleading, unreliable, and faulty questions from the pool. The aim of this step is to remove questions that resemble queries, rather than clarifying questions (e.g., "What is food like in Nigeria?" vs. "Would you like to know more about food in Nigeria?"). To this end, we utilize a RoBERTa-based [12] classifier to distinguish between regular questions taken from CAsT'20/21 and clarifying questions taken from ClariQ [2]. Specifically, we fine-tune the `roberta-base`[5] model on several hundred positive and negative examples, achieving the accuracy of 97% on a development set comprising 400 questions taken from CAsT'21 and ClariQ, which are not part of the training set. Finally, we apply the classifier to the provided candidate question pool and filter out around 20% of the questions.

We formulate the task of asking clarifying questions as a ranking task. More specifically, for each query, we rank the potential clarifying question candidates based on their semantic similarity. To this end, we use MPNet [20] from Sentence-Transformers, trained for general-purpose semantic matching. For each query,

[3] https://huggingface.co/castorini/monot5-base-msmarco
[4] https://huggingface.co/castorini/duo5-base-msmarco
[5] https://huggingface.co/roberta-base

we select the clarifying question with the highest score, as predicted by MPNet. (The run corresponding to this method is called **uis_clearboat**.)

Clarifying question generation. We utilize a template-based method for generating clarifying questions. First, we employ a topic model on the top 200 passages retrieved in response to the original utterance. Specifically, we utilize Top2Vec [4], a neural topic model that automatically detects topics from text. Then, we formulate a template-based question with up to top three extracted topics (e.g., "Are you interested in Topic_1, Topic_2, or Topic_3?"). We adjust the template accordingly if less than 3 topics are extracted from the passages. (This method for constructing clarifying questions generated the run **uis_vagueboat**.)

Answer processing. We define three possible actions, based on the current utterance, the clarifying question asked, and the answer from the organizers. Specifically:

(1) In case the answer is affirmative (e.g., "yes" or "Yes, that is what I'm looking for"), we expand the current utterance by appending the clarifying question asked.
(2) In case the answer is deemed useful, i.e., a certain degree of the underlying information need is expressed, we expand the current utterance by appending the answer.
(3) If neither (1) nor (2) is selected, we do not expand the utterance.

To classify a (utterance, clarifying question, answer) triplet in one of the aforementioned classes, we fine-tune RoBERTa on 150 samples from ClariQ, which we manually annotated. We perform a grid search for the optimal parameters and settle for a learning rate of 4×10^{-4} for 2 epochs. The classifier is then run on the CAsT'22 data. Finally, in cases (1) and (2), where the original utterance was expanded, we run a T5-based model for query rewriting, explained in Section 3.2. The rewritten utterance is further fed into the standard retrieval and re-ranking pipeline explained in Sections 3.3–3.4.

4 Submitted Runs

This section contains a high-level description of our submitted runs. The differences between runs lie mainly in the query rewriting component.

uis_duoboat This run is considered as our baseline. It uses the most basic versions of our components. First-pass retrieval is based on BM25 with the parameters tuned on 2020 and 2021 CAsT datasets. It is followed by monoT5 re-ranking and duoT5 re-ranking fine-tuned on MS MARCO. Query rewriting is performed with a HuggingFace model fine-tuned on the CANARD dataset, using previously rewritten utterances and the last canonical response as context.

Table 2. Overview of submitted runs.

RunID	Query rewriting	First-pass retr.	Re-ranking	Task	Priority
uis_duoboat	T5 fine-tuned on CANARD (T5_CANARD)	BM25	mono/duoT5	Main	4
uis_sparseboat	sparse query rewriting	BM25 + PRF	mono/duoT5	Main	1
uis_cargoboat	sparse query rewriting with intent classification	BM25 + PRF	mono/duoT5	Main	3
uis_mixedboat	mixed-initiative query rewriting	BM25 + PRF	mono/duoT5	MI	2

uis_sparseboat This run is similar to **uis_duoboat** with the addition of sparse query rewriting and pseudo-relevance feedback.

uis_cargoboat For this run, the input query is pre-processed based on its intents before performing sparse query rewriting (cf. Section 3.2). The first-pass retrieval is based on BM25 with the same parameters as our baseline and additionally employs pseudo-relevance feedback. Finally, the re-ranking stage is the same as in **uis_duoboat**.

uis_mixedboat For each query, we select the highest scoring clarifying question and process the given answer as described in Section 3.5 (**uis_clearboat**). The expanded query is then fed into the pipeline of BM25 first-pass retrieval with pseudo-relevance feedback and mono/duo T5 re-ranking (as in **uis_duoboat**).

5 Results

This section presents the performance our runs on the TREC CAsT'21 and '22 datasets, and discusses the obtained results.

5.1 Results on TREC CAsT 2021

Table 3 reports the performance of the systems used for generating this year's submission runs on the TREC CAsT'21 dataset. Additionally, we report on the performance of the basic retrieve-then-re-rank system with raw (No rewrites), automatically rewritten (TREC-Auto), and manually rewritten (TREC-Manual) queries, as provided by the track organizers. We find that our baseline system (**uis_duoboat**) outperforms the system using TREC-Auto in the re-ranking stage (higher values of NDCG at early rank positions) but achieves lower results in first-pass retrieval, which is most likely caused by a worse-performing query rewriting module. Our top-priority run (**uis_sparseboat**) achieves the highest results in almost all reported metrics (the only exception is MRR). The most noticeable improvement is in first-pass retrieval, where we observe a drop of only 4% in recall compared to the system using manual rewrites. The **uis_cargoboat** run outperforms our baseline system in first-pass retrieval. However, the intent classification used in query rewriting yields lower performance in the re-ranking stage (NDCG at early positions is lower than for the baseline).

Table 3. Performance of our approaches on TREC CAsT'21.

Approach/RunID	R@1000	MAP	MRR	NDCG	NDCG@3	NDCG@5
No rewrites + BM25 + monoT5	0.3497	0.1217	0.2875	0.2605	0.2051	0.2041
TREC-Auto + BM25 + monoT5	0.6319	0.2684	0.5575	0.4842	0.3972	0.3969
TREC-Manual + BM25 + monoT5	0.7733	0.3858	0.7326	0.6293	0.5611	0.5654
uis_duoboat	0.6037	0.2544	0.5563	0.4724	0.4110	0.4048
uis_sparseboat	**0.7424**	**0.2986**	0.5979	**0.5475**	**0.4405**	**0.4380**
uis_cargoboat	0.6930	0.2977	**0.6605**	0.5132	0.3870	0.3918

Table 4. Clarification question evaluation of MI runs using crowdsourced jugdments.

Approach/RunID	Relevance @1	Novelty @1	Diversity @1
(baseline) T5_raw	0.232	0.166	0.185
(baseline) T5_rewrite	0.320	0.229	0.210
(baseline) bm25_baseline_mi	0.345	0.293	0.307
(baseline) miniLM_bert_sample_mi_run	0.371	0.317	0.395
TREC best	0.852	0.536	0.607
uis_vagueboat	0.237	0.322	0.381
uis_clearboat	0.639	0.488	0.371

5.2 Clarifying Question Evaluation

Table 4 reports the performance of our clarifying question construction methods described in Section 3.5, i.e., **uis_clearboat** and **uis_vagueboat**. Performance is measured in terms of relevance, novelty, and diversity, and compared against organizers' baselines and the top performing MI subtask submission (GPT-3_full_context). The results show significant improvements of our question selection method (**uis_clearboat**) over relevant baselines across all three metrics. On the other hand, the clarifying question generation method (**uis_vagueboat**) demonstrates significantly worse performance.

5.3 Results on TREC CAsT 2022

Table 5 reports the performance of the submitted runs on TREC CAsT'22 according to the evaluation results provided by the organizers. These correspond to a *strict* evaluation, where passages must be of at least relevance level 2. Note that the numbers slightly differ from those we reported in our working notes paper, due to a bug in the organizers' evaluation script. We include the updated numbers here, which we received from the organizers in private communication. For reference, we include the results of the baseline systems provided by the organizers (BM25_T5_BART_automatic and BM25_T5_BART_manual),[6] the best performing participant system (udinfo_m_b2021) as well as the median for the main evaluation measure (NDCG@3). Note that these values are taken from the

[6] https://github.com/daltonj/treccastweb/tree/master/2022/baselines/main_task

Table 5. Performance of our approaches on TREC CAsT'22 (*strict* evaluation).

Approach/RunID	R@1000	MAP	MRR	NDCG	NDCG@3
BM25_T5_BART_automatic	0.324	0.150	0.527	0.299	0.362
BM25_T5_BART_manual	0.465	0.231	0.716	0.423	0.503
TREC best	0.771	0.246	0.656	0.557	0.452
TREC median					0.347
uis_duoboat	0.365	0.154	0.476	0.323	0.345
uis_sparseboat	**0.507**	**0.189**	**0.566**	**0.409**	**0.388**
uis_cargoboat	0.450	0.180	0.526	0.377	0.373
uis_mixedboat	0.445	0.186	0.499	0.374	0.363

notebook version of the track overview paper and might have changed slightly due to the aforementioned bug; at the time of writing, the updated results for other runs were not available.

The priorities assigned by us for the runs are in accordance with the performance of the corresponding systems (with the exception of the **uis_cargoboat** run performing slightly better than **uis_mixedboat**). Similarly to the evaluation on TREC CAsT'21, **uis_sparseboat** achieves the highest results in all reported metrics. The most noticeable differences with other runs are observed in recall. The run **uis_cargoboat**, which differs from **uis_sparseboat** only in adding intent classification, yields lower performance. It implies that our intent handling is removing some important information from the queries, which negatively impacts both first-pass retrieval and end-to-end performance. The performance of **uis_mixedboat** is comparable to that of **uis_cargoboat**. However, it is difficult to reason about the impact of mixed-initiative components on the results, due to significant differences between the underlying ranking pipelines. Our baseline (**uis_duoboat**) outperforms the organizers' baseline with automatic query rewrites (BM25_T5_BART_automatic) in first-pass retrieval. However, the re-ranking stage is less effective in pushing the most relevant documents to the top of the final ranking (NDCG@3 is lower than the organizers' baseline).

6 Conclusion

In TREC CAsT'22, our team experimented with different approaches for query rewriting. All the experiments with this module were performed using a well-established two-stage retrieval-re-ranking pipeline using BM25 combined with pseudo-relevance feedback for first-pass retrieval and a pointwise/pairwise re-ranker. Results indicate that our sparse query rewriting works well, while leveraging intent classification in query rewriting causes a slight drop in performance. Performance analysis of the mixed-initiative run remains to be done as evaluation details for this sub-task were not available at the time of writing.

Bibliography

[1] M. Aliannejadi, H. Zamani, F. Crestani, and W. B. Croft. 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 '19, pages 475–484, 2019.

[2] M. Aliannejadi, J. Kiseleva, A. Chuklin, J. Dalton, and M. Burtsev. Building and evaluating open-domain dialogue corpora with clarifying questions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, EMNLP '21, pages 4473–4484, 2021.

[3] J. E. Allen, C. I. Guinn, and E. Horvtz. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications*, 14(5):14–23, 1999.

[4] D. Angelov. Top2Vec: Distributed representations of topics. *arXiv*, cs.CL/2008.09470, 2020.

[5] C.-Y. Chang, H.-H. Chen, N. Chen, W.-T. Chiang, C.-H. Lee, Y.-H. Tseng, M.-F. Tsai, and C.-J. Wang. Query expansion with semantic-based ellipsis reduction for conversational IR. In *The Twenty-Ninth Text REtrieval Conference Proceedings*, TREC '20, 2020.

[6] J. Dalton, C. Xiong, and J. Callan. TREC CAsT 2019: The conversational assistance track overview. In *The Twenty-Eighth Text REtrieval Conference Proceedings*, TREC '19, 2019.

[7] C. Gemmell and J. Dalton. Glasgow representation and information learning lab (GRILL) at the conversational assistance track 2020. In *The Twenty-Ninth Text REtrieval Conference Proceedings*, TREC '20, 2020.

[8] J.-H. Ju, C.-T. Yeh, C.-W. Lin, C.-Y. Tsao, J.-E. Ding, M.-F. Tsai, and C.-J. Wang. An exploration study of multi-stage conversational passage retrieval: Paraphrase query expansion and multi-view point-wise ranking. In *The Thirtieth Text REtrieval Conference Proceedings*, TREC '21, 2021.

[9] I. Kostric, K. Balog, M. Book, T. Linjordet, and V. Setty. The University of Stavanger (IAI) at the TREC 2021 conversational assistance track. In *The Thirtieth Text REtrieval Conference Proceedings*, TREC '21, 2021.

[10] V. Kumar and J. Callan. Making information seeking easier: An improved pipeline for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, EMNLP '20, pages 3971–3980, 2020.

[11] S.-C. Lin, J.-H. Yang, R. Nogueira, M.-F. Tsai, C.-J. Wang, and J. Lin. Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting. *ACM Transactions on Information Systems*, 39(4):1–29, 2021.

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*, cs.CL/1907.11692, 2019.

[13] I. Mele, C. I. Muntean, F. M. Nardini, R. Perego, N. Tonellotto, and O. Frieder. Topic propagation in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 2057–2060, 2020.

[14] I. Mele, C. I. Muntean, F. M. Nardini, R. Perego, N. Tonellotto, and O. Frieder. Adaptive utterance rewriting for conversational search. *Information Processing & Management*, 58(6): 102682, 2021.

[15] R. Nogueira, Z. Jiang, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, EMNLP '20, pages 708–718, 2020.

[16] R. Pradeep, R. Nogueira, and J. Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv*, cs.IR/2101.05667, 2021.

[17] C. Qu, L. Yang, W. B. Croft, J. R. Trippas, Y. Zhang, and M. Qiu. Analyzing and characterizing user intent in information-seeking conversations. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, pages 989–992, 2018.

[18] C. Rosset, C. Xiong, X. Song, D. Campos, N. Craswell, S. Tiwary, and P. Bennett. Leading conversational search by suggesting useful questions. In *Proceedings of The Web Conference 2020*, WWW '20, pages 1160–1170, 2020.

[19] I. Sekulić, M. Aliannejadi, and F. Crestani. Towards facet-driven generation of clarifying questions for conversational search. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, pages 167–175, 2021.

[20] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. MPNet: Masked and permuted pre-training for language understanding. *arXiv*, cs.CL/2004.09297, 2020.

[21] S. Vakulenko, S. Longpre, Z. Tu, and R. Anantha. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, pages 355–363, 2021.

[22] S. Vakulenko, N. Voskarides, Z. Tu, and S. Longpre. A comparison of question rewriting methods for conversational passage retrieval. In *European Conference on Information Retrieval*, ECIR '21, pages 418–424, 2021.

[23] N. Voskarides, D. Li, P. Ren, E. Kanoulas, and M. de Rijke. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 921–930, 2020.

[24] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. N. Bennett, J. Ahmed, and A. Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, ICLR '20, 2020.

[25] X. Yan, C. L. Clarke, and N. Arabzadeh. WaterlooClarke at the TREC 2021 conversational assistant track. In *The Thirtieth Text REtrieval Conference Proceedings*, TREC '21, 2021.

[26] S. Yu, J. Liu, J. Yang, C. Xiong, P. Bennett, J. Gao, and Z. Liu. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1933–1936, 2020.

[27] H. Zamani, S. Dumais, N. Craswell, P. Bennett, and G. Lueck. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*, WWW '20, pages 418–428, 2020.