HBKU at TREC 2020: Conversational Multi-Stage Retrieval with Pseudo-Relevance Feedback

Haya Al-Thani
hayaalthani@hbku.edu.qa
Hamad Bin Khalifa University
Doha, Qatar

Bernard J. Jansen
jjansen@acm.org
Qatar Computing Research Institute,
Hamad Bin Khalifa University
Doha, Qatar

Tamer Elsayed
telsayed@qu.edu.qa
Qatar University
Doha, Qatar

ABSTRACT

Passage retrieval in a conversational context is extremely challenging due to limited data resources. Information seeking in a conversational setting may contain omissions, implied context, and topic shifts. TREC CAsT promotes research in this field by aiming to create a reusable dataset for open-domain conversational information seeking (CIS). The track achieves this goal by defining a passage retrieval task in a multi-turn conversation setting. Understanding conversation context and history is a key factor in this challenge. This solution addresses this challenge by implementing a multi-stage retrieval pipeline inspired by last year's winning algorithm. The first stage in this retrieval process is a historical query expansion step from last year's winning algorithm where context is extracted from historical queries in the conversation. The second stage is the addition of a pseudo-relevance feedback step where the query is expanded using top-k retrieved passages. Finally, a pre-trained BERT passage re-ranker is used. The solution performed better than the median results of other submitted runs with an NDCG@3 of 0.3127 for the best performing run.

KEYWORDS

Conversational Information Seeking, Conversational Search Systems, Multi-Stage Retrieval Systems, Open-Domain

ACM Reference Format:
Haya Al-Thani, Bernard J. Jansen, and Tamer Elsayed. 2021. HBKU at TREC 2020: Conversational Multi-Stage Retrieval with Pseudo-Relevance Feedback. In *Proceedings of* . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Conversational Search Systems have been getting a lot of attention from the IR and NLP communities. The advancement of machine learning techniques has led to the rise of conversational agents such as digital personal assistants and smart speakers like Amazon's Alexa, Apple's Siri, and many others. Digital Assistants are often used to help people increase their productivity.Conversational information seeking (CIS) is an important emerging research area. As opposed to traditional information seeking, in CIS a user searches for information regarding single or multiple topics in a series of sequential questions or turns. Previous turns usually have an impact on subsequent turns. As a result, later turns in a conversation are often not well-formed or ambiguous. One of the field's key challenges is how to ensure context-awareness throughout the whole conversation. Another challenge is limited labeled data appropriate for training and evaluating CIS models. The Conversational Assistance Track (CAsT) was organized starting in TREC 2019 to address this challenge.

The goal of CAsT is to create a reusable benchmark for open-domain conversational search where answers are retrieved passages from a large text corpus. TREC CAsT aims to address some of the first major challenges in building conversational search systems (CSS). Their goal is to create a large-scale reusable test collection for open-domain CSS. This year's task is to retrieve a ranked list of passages over a large collection to satisfy the information need of a multi-turn conversation. Conversational search queries are often ambiguous due to coreference and omission problems. A user may often ask questions referring to previous turns in the dialogue. This year, the CAsT track's primary focus is to understand the information need behind different turns in the conversation context and find relevant responses. Table 1 shows an example of a conversation in the CAsT training set. It can be noted that subsequent turns in the conversation can only be understood by looking at the conversation's turn history. For example, the second utterance in this conversation references the first turn, and in later utterances, such as at turn 5, context is completely omitted. Incorporating context history accurately in such CSS problems is essential to improve performance.

To solve this task while still addressing these challenges, a multi-stage retrieval solution is proposed. The solution is inspired by last year's winning algorithm which applies a query expansion technique to add context to ambiguous conversation turns [18]. The historical query expansion algorithm (HQE) is a non-parametric algorithm that extracts topic and subtopic keywords from the previous conversation turns and uses these keywords to expand the turn query. After performing this expansion, passages are retrieved using a BM25 retrieval model. After that, a BERT passage re-ranker trained on the MS MARCO passage dataset is used to re-rank retrieved passages [13]. In this solution, a pseudo-relevance feedback stage is added to this pipeline. Pseudo-relevance feedback (PRF or blind feedback) uses top retrieved documents to extract terms to use in the query expansion stage. The user is not involved in the selection of relevant documents. PRF techniques can improve performance of many retrieval models [19]. For this year's CAsT task, this additional stage is added to the retrieval pipeline. After the HQE stage, a passage query expansion (PQE) step is added. PQE uses PRF to further expand the turn query by adding terms from the top-k retrieved passages based on TF-IDF. The terms of the top-k passages retrieved from the HQE stage are ranked by the TF-IDF scoring scheme. Then the top terms from these passages are used to expand the turn query further. After this additional

Table 1: TREC CAsT sample topic from the 2019 Train Dataset

Title: Blood sugar	
Description: Blood sugar levels and possible complications that may arise due to it.	
Turn	**Conversation Utterances**
1	What is a normal blood sugar level?
2	What does it mean if it's higher than this?
3	What is a dangerous level?
4	How do you bring it down quickly?
5	How fast can it rise?
6	And what if it's lower than normal?
7	How does this make you feel?
8	Do different activities lead to an imbalance?
9	Is there a relation between age and sugar levels?

stage, the resulting list of retrieved passages are re-ranked using the pre-trained BERT re-ranker.

Conversations are made up of a series of related or unrelated questions. In a regular conversation, turns can heavily depend on previous questions or answers, or shift to a completely new topic. In order to handle these omissions or shifts in conversation, turn queries are syntactically analyzed and categorized. The first query category proposed are 'explicit' queries. Explicit queries are considered complete and contain enough context. The second category would be 'implicit' queries that contain omissions or coreferences. Different methods were tested to categorize queries but, in the end, a simple syntactical method was implemented. A turn that contains no pronouns is assumed to be 'explicit', while a turn that contains at least one pronoun is 'implicit'.

For the submitted runs, the parameters of the HQE stage were kept constant and set to the same parameters that achieved best performance on last year's task. The runs experimented with different parameters for both the stage 2: PQE and stage 3: BERT re-ranker. Some runs implement the proposed query categorization scheme while others don't perform any categorization.

2 LITERATURE REVIEW

Question answering is the process of finding answers to a given question given some context. This area has seen a lot of progress due to the successful application of deep learning architectures and the availability of large scale datasets such as MS MARCO, SQUAD, and HotpotQA [12, 14, 20]. A large focus of the field in recent years has been targeted towards neural QA [15]. Conventionally, neural QA is a two-stage process: first, relevant passages are retrieved and then a neural network model extracts the likeliest answer [11]. Devlin *et. al.* introduced BERT or bidirectional Encoder Representations from Transformers [6]. BERT is a language model that is pre-trained to learn deep bidirectional representations from text. A pre-trained BERT model can be fine-tuned on a specific task by adding an output layer. BERT has made a massive impact in many NLP tasks, including QA. In the work of Nogueira *et. al.*, BERT is re-implemented as a passage re-ranker and achieves state-of-the-art results on the MS MARCO passage re-ranking task [13].

In **open domain question answering**, the system returns answers to user questions from a wide range of domains. The pipeline of an open-domain system involves a retriever for selecting relevant documents from a large corpus of text such as Wikipedia and a machine reading comprehension model for inferring the answer from the retrieved documents [5]. Open-domain QA was popularized by the TREC-8 task [16]. It has recently gained a lot of traction due to the emergence of multiple datasets such as SearchQA, TriviaQA, and Quasar [7, 8, 10].

Multi-stage retrieval systems can be taken as a two-step process. First, a list of candidate documents are generated, and then the list goes through one or more re-ranking stage. The number of stages have to be considered with efficiency and effectiveness in mind[3]. The baseline of this system is a cascade pipeline of BM25 candidate generation followed by a BERT re-ranker. The effectiveness of this has been proven in multiple IR datasets such as TREC CAR and MS MARCO.

Conversational Search is a major research problem in the IR community. Conversational search has been applied in many domains such as conversational recommendation systems, e-health systems, and personality recognition [1]. In the past few years rule-based conversational IR has given way to methods based on deep learning [9]. A significant topic of research in this field involves identifying user need while searching for information. One work that focuses on this problem has been to include query suggestion to clarify user's intent. By asking clarifying questions, user's intent can be better understood and the search can be redirected to achieve better results [2]. One major factor to consider when designing a conversational agent is how to maintain the conversation context [17]. Maintaining context is essential to user experience since formulating long questions and sentences is not natural in a normal conversation setting. Addressing this context problem is the focus of last year and this year's CAsT track.

3 PROBLEM DEFINITION

In the CAsT track, conversational search is defined as an information retrieval task in a conversational setting. The goal of the task is to fulfill user's information need which is expressed through turns in a conversation. The response is a list of top-k relevant short passages retrieved from a large collection of passages. The task in year 1 focused on candidate response retrieval for conversational information seeking. Year 2 is similar to year 1 and focuses on

candidate response ranking in context. The difference between the 2019 and 2020 CAsT task is the conversations will not include topic titles and descriptions. Topics will unfold through the conversation turns. A canonical system response will also be included with each previous turn.

The goal is to keep the task simple in order to create a reusable collection. Formally, a conversation S is made up of a series of n turn utterances u such that $S = \{ u_1, u_2, u_3, \ldots, u_n \}$. The task is to retrieve a list of top-k passages P_i for each turn u_i to satisfy the information need of turn i.

Submission categories for this year will be 'automatic' runs that use the raw turns. 'Manual' runs that use manually rewritten turns that are rewritten to remove ambiguity and to add context. The third and last submission category is the 'canonical' run that use both the raw turns and the included canonical responses for previous turns.

Passage collections used this year are made up of passages from MS MARCO and the TREC Complex Answer Retrieval Paragraph Collection:

- MS MARCO has 1 million real search queries each with 10 passages from top ranked results. This results in a pool of approximately 8 million passages. The MARCO collection does contain near duplicates.
- TREC CAR paragraph Corpus V2.03 is used. It's made up of paragraphs from Wikipedia '16. This corpus has been deduplicated. It contains approximately 30 million unique paragraphs.

4 METHODOLOGY

The solution is implemented as a three-stage retrieval pipeline. The first stage is the historical query expansion stage (HQE). HQE is a BM25 retrieval model that first extract context from previous turns and uses these extracted keywords to expand the turn query. The HQE algorithm along with a BERT re-ranker achieved best performance in 2019 CAsT challenge. The second stage is the passage query expansion stage (PQE) and is another BM25 retrieval phase but it adds context using pseudo-relevance feedback. The third and final stage is a pre-trained BERT re-ranker pre-trained on the MS MARCO dataset. The Anserini toolkit was used for collection indexing and retrieval [1]. The SpaCy library was used to syntactically analyze the conversation turns[2].

4.1 System Design

Figure 1 shows the overall system design. The multi-stage retrieval system consists of three stages. The system first starts with the raw turn query as the input. The query is then expanded using the historical query expansion algorithm. Then the expanded query is used to retrieve the first ranked list of passages. The passages are retrieved from an indexed collection of the combination of the MS MARCO and TREC CAR datasets. After that, the turn query is expanded again using the top terms from the top passages retrieved from the first stage HQE. The terms are selected based on TF-IDF. The turn query is expanded with the passage terms and a new ranked list of passages is retrieved from the collection. Finally, a pre-trained BERT re-ranker is used to re-rank the list to get the final ranked list of retrieved passages.

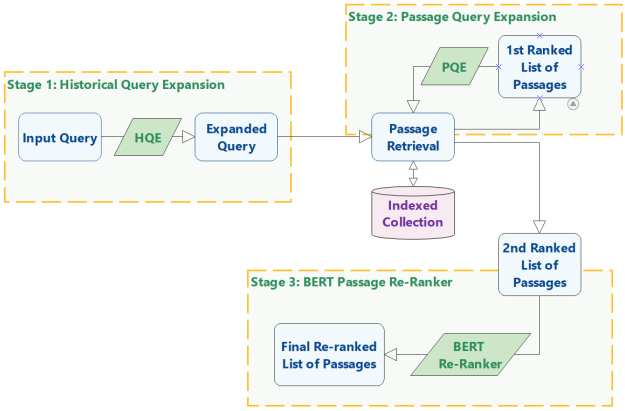

Figure 1: Multi-Stage Retrieval System Pipeline.

4.2 Stage 1: Historical Query Expansion

The first stage of this multi-phase solution is to extract context from conversation turns using the historical query expansion algorithm that achieved best performance on the 2019 CAsT data (HQE) [18]. This method consists of mainly three main steps:

(1) Extraction of general topic and sub-topic keywords from historical turns within a conversation.
(2) The measurement of turn ambiguity.
(3) Query expansion using topic and subtopic keywords extracted from historical turns.

Main topic and subtopic keywords are extracted using what the authors call the keyword extractor (KE). The aim of KE is to compute the score of each token within an utterance. The score of that token indicates the importance of the token in the utterance. The authors use the BM25 retrieval score of the token's most relevant document as that indicator. The theory behind this design is that the importance of a token can be judged based on those documents that are most relevant to it. If the token's importance score is higher than a certain threshold it is considered a keyword.

The other main component to this solution is the query performance predictor (QPP). This component measures the turn utterance's ambiguity. They establish that ambiguity is closely related to its ambiguity with respect to the collection being searched. Thus, it is measured by analyzing retrieval score. If a turn's ambiguity score is higher than a certain threshold, the query is expanded using the HQE method.

4.3 Stage 2: Passage Query Expansion

The second stage in this multi-stage retrieval process would be passage query expansion (PQE). In this stage, the queries are expanded through pseudo-relevance feedback where query expansion terms are extracted from top-ranked documents retrieved from the initial query [4]. PQE utilizes tf-idf based pseudo-relevance feedback where the expansion terms are obtained from the top retrieved passages ranked by BM25.

[1] Anserini, https://github.com/castorini/anserini
[2] SpaCy, https://spacy.io/

In order to avoid over loading the query with extra terms, the query is first categorized into 'explicit' and 'implicit' queries. Explicit queries are queries that are assumed to be complete and not in need of further expansion. Implicit queries are queries that are assumed to contain ambiguity like coreferences and omissions.

To categorize the turn utterances, the turns were syntactically analyzed using the SpaCy library. After tokenizing the turn, SpaCy parses and tags the given query which enables SpaCy to predict which tag or label most likely applies to the token in context. using this method, the turn was broken down into its composite objects such as verbs, nouns and pronouns. After analyzing different categorization methods using the syntactical structure of the query, the two categories were defined by focusing on the pronouns in a query. A turn was considered 'explicit' if it contained no pronouns, while 'implicit' queries contain at least one pronoun. If the turn utterance belongs to the 'implicit' query category, then it is further expanded using pseudo relevance feedback.

4.4 Stage 3: BERT Passage Re-Ranker

The final stage is to re-rank passages using a BERT re-ranking model. In order to compensate for the sparsity of CAsT training data, a pre-trained BERT model which is trained for passage re-ranking on another larger dataset is used. The pre-trained BERT re-ranker is publicly available, and is pre-trained on both MS MARCO and TREC CAR [13]. The MS MARCO dataset was used for training in this stage.

5 EXPERIMENTAL EVALUATION

To evaluate the performance of this retrieval pipeline on the 2020 data, the passage collection was first pre-processed and indexed. The evaluation measures and metrics are presented as well as an explanation of the four submitted runs.

5.1 Dataset Pre-processing and Indexing

This year's CAsT track used two collections: MS MARCO and TREC CAR. The first step in data pre-processing is to remove duplicate passages from the collections using the provided duplicate passage list provided by the organizers. This mostly affects MS MARCO passages since CAR has already been deduped.

After that, the two collections are combined as a single collection with the schema of 'docid' and 'content'. This combined collection is later indexed using the Anserini toolkit.

5.2 Evaluation Measures

The submitted runs are evaluated at NIST using the standard TREC style pooling and relevance assessment. Response pooling of the top results for the system was performed across participants. Response is assessed using a five-point relevance scale where:

- 0- Not relevant and fails to meet requirement.
- 1- Slightly meets and the answer can be inferred from the passage with some effort.
- 2- Moderately meets requirement and the passage answers the turn but is focused on something related.
- 3- Highly meets requirement and the passage answers the turn and is focused on the answer.
- 4- Fully meets requirement and the passage is the perfect response to the turn

5.3 Metrics

The evaluation metrics used are NDCG@1, NDCG@3, NDCG@5 and MAP@1000. The turn depth evaluates the system performance at the *n-th* turn in the conversation. Better performance at deeper turns in a conversation (larger *n*) indicates that the system is better at interpreting context.

5.4 Official Runs

The four submitted runs were for the 'automatic' category which focuses on retrieving passages using the raw utterances exclusively. The parameters at the HQE stage are constant and set to the parameters suggested by the algorithm authors. The submitted runs are as follows:

HBKU_t2_1v1: This run expands queries in the PQE stage using top 3 terms from the top 3 retrieved passages. No query categorization was applied. Both HQE and PQE expanded queries were fed to the BERT re-ranker as input.

HBKU_t2_1v2: This run again expands queries in the PQE stage using top 3 terms from the top 3 retrieved passages without applying query categorization. only HQE expanded query was used at the BERT re-ranking phase.

HBKU_t5_1v1: This run is similar to the first one where the query was expanded using top 3 terms from the top 3 retrieved passages, however only queries of the 'implicit' category was expanded. Both HQE and PQE expanded queries were fed to the BERT re-ranker as input.

HBKU_t5_1v2: This run is similar to the previous run, however, in the BERT re-ranking phase only HQE expanded terms were used

6 EXPERIMENTAL RESULTS

Table 2 shows a summary of the overall performance of this solution for the CAsT 2020 dataset. The proposed pipeline's performance is compared against the average median performance of all the submitted 'automatic' runs. From the table, it can be observed that all runs surpass the median performance on almost all the metrics. The best performing run is the HBKU_t5_1v2 where query categorization was applied to only supplement 'implicit' queries and where BERT was fed queries with HQE expansion. This shows that using the pseudo-relevant feedback method of PQE based on the TF-IDF scoring scheme might be best used for queries that are more ambiguous.

It is also interesting to see how performance varies across different turn depths. Table 3 shows the performance of the best submitted run (HBKU_t5_1v2) across different turns in a single topic. The topic being investigated is topic 81 which consists of 8 turns.

Table 3 illustrates that performance can fluctuate at different turn depths. The NDCG@3 and NDCG@5 score of turn IDs 81_4 and 81_7 was zero. The values for the different metrics fluctuate greatly from one turn to the other. This emphasizes how important it is to tailor retrieval for the conversation scenario. Context can be lost from one turn to the other when working with heavily interrelated turns, while topic shifts in a conversation can indicate a

Table 2: Submitted Run Performance Compared to Median of the Submitted 'Automatic' Runs

	NDCG@3	NDCG@5	NDCG@1000	AP@1000
Median	*0.279578*	*0.273523*	*0.374911*	*0.180143*
HBKU_t2_1v1	0.2958	0.29	0.3692	0.2038
HBKU_t2_1v2	0.3089	0.2994	0.377	0.2077
HBKU_t5_1v1	0.3066	0.2964	0.3736	0.2061
HBKU_t5_1v2	**0.3127**	**0.3026**	**0.379**	**0.2083**

Table 3: Performance of Topic 81 Across Different Turn Depths

Turn ID	*NDCG@3*	*NDCG@5*	*NDCG@1000*	*AP@1000*
81_1	0.3348	0.4364	0.4218	0.1936
81_2	0.3616	0.4489	0.4768	0.2952
81_3	0.5312	0.6778	0.8407	0.7543
81_4	0	0	0.3263	0.0412
81_5	0.0848	0.0664	0.4247	0.1258
81_6	0.4693	0.3392	0.3352	0.0683
81_7	0	0	0.3345	0.0786
81_8	0.2346	0.1969	0.2422	0.0637

need to reset conversation history. Better methodologies of query categorization might help understanding a conversation's context flow and where to add context and where not to.

7 CONCLUSION AND FUTURE WORK

Passage Retrieval in a conversational setting is a very challenging field that introduces new problems for the IR and NLP communities. In a conversation, turns are often related and can contain coreferences and omissions. Context and history of a conversation is a very important factor in understanding user's information need. TREC CAsT aims to create a reusable benchmark by introducing a conversational passage retrieval task.

This submission to the 2020 TREC CAsT challenge tries to introduce context to conversation turns through a multi-stage retrieval pipeline inspired by last years winning algorithm. A pseudo-relevance feedback step is introduced to the pipeline to try to enrich queries with terms from retrieved passages. In a conversation both the questions and answers usually direct the flow of the conversation. Using this theory, the passage query expansion stage is added to the pipeline. It is also very important to understand the type of turns in a conversation. Often times, a conversation turn can contain missing information or topic shifts. In this solution, queries are categorized into 'explicit' or 'implicit' queries based on whether the query contains pronouns. 'implicit' queries contain at least one pronoun and are assumed to need further clarification.

The solution performed better than the median across all submitted runs. However, for the future a more advanced method can be used to select terms in the PQE phase. This solution uses TF-IDF, but more advanced machine reading comprehension models can be used to select these terms. Using these sophisticated models, an 'answer' from the top retrieved answers can be used to add context to the turn. Turn categorization can be a valuable tool to understand the turns across different depths in the conversation. Better query categorization can help direct how to best add context to different turns.

REFERENCES

[1] Mohammad Aliannejadi, Manajit Chakraborty, Esteban Andrés Ríssola, and Fabio Crestani. 2020. Harnessing evolution of multi-turn conversations for effective answer retrieval. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*. 33–42.

[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*. 475–484.

[3] Nima Asadi and Jimmy Lin. 2013. Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13* (Dublin, Ireland). ACM Press, 997–1000. https://doi.org/10.1145/2484028.2484132

[4] Hiteshwar Kumar Azad and Akshay Deepak. 2019-09. Query expansion techniques for information retrieval: A survey. 56, 5 (2019-09), 1698–1735. https://doi.org/10.1016/j.ipm.2019.05.009

[5] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019-05-14. Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering. (2019-05-14). arXiv:1905.05733 http://arxiv.org/abs/1905.05733

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019-05-24. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2019-05-24). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[7] Bhuwan Dhingra, Kathryn Mazaitis, and William W. Cohen. 2017-08-08. Quasar: Datasets for Question Answering by Search and Reading. (2017-08-08). arXiv:1707.03904 http://arxiv.org/abs/1707.03904

[8] Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017-06-11. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. (2017-06-11). arXiv:1704.05179 http://arxiv.org/abs/1704.05179

[9] Jianfeng Gao, Michel Galley, and Lihong Li. 2019-09-10. Neural Approaches to Conversational AI. (2019-09-10). arXiv:1809.08267 http://arxiv.org/abs/1809.08267

[10] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017-05-13. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. (2017-05-13). arXiv:1705.03551 http://arxiv.org/abs/1705.03551

[11] Bernhard Kratzwald, Anna Eigenmann, and Stefan Feuerriegel. 2019. Rankqa: Neural question answering with answer re-ranking. *arXiv preprint*

arXiv:1906.03008 (2019).

[12] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2017. MS MARCO: A HUMAN GENERATED MACHINE READING COMPREHENSION DATASET. (2017), 10.

[13] Rodrigo Nogueira and Kyunghyun Cho. 2020-04-14. Passage Re-ranking with BERT. (2020-04-14). arXiv:1901.04085 http://arxiv.org/abs/1901.04085

[14] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018-06-11. Know What You Don't Know: Unanswerable Questions for SQuAD. (2018-06-11). arXiv:1806.03822 http://arxiv.org/abs/1806.03822

[15] Maartje ter Hoeve, Robert Sim, Elnaz Nouri, Adam Fourney, Maarten de Rijke, and Ryen W. White. 2020-03-14. Conversations with Documents: An Exploration of Document-Centered Assistance. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval* (Vancouver BC Canada). ACM, 43–52. https://doi.org/10.1145/3343413.3377971

[16] Ellen M Voorhees. 1999. The TREC-8 question answering track report. In *TREC*, Vol. 99. 77–82.

[17] Alexandra Vtyurina, Denis Savenkov, Eugene Agichtein, and Charles L. A. Clarke. 2017. Exploring Conversational Search With Humans, Assistants, and Wizards. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '17* (Denver, Colorado, USA). ACM Press, 2187–2193. https://doi.org/10.1145/3027063.3053175

[18] Jheng-Hong Yang, Sheng-Chieh Lin, Chuan-Ju Wang, Jimmy Lin, and Ming-Feng Tsai. 2019. Query and Answer Expansion from Conversation History.. In *TREC*.

[19] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. 2018-06-27. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems. (2018-06-27), 245–254. https://doi.org/10.1145/3209978.3210011 arXiv:1805.00188

[20] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018-09-25. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. (2018-09-25). arXiv:1809.09600 http://arxiv.org/abs/1809.09600