TKB48 at TREC 2021 Conversational Assistance Track

Yubo Fang
Graduate School of Comprehensive Human Sciences, University of Tsukuba
Tsukuba, Ibaraki, Japan
s2021712@s.tsukuba.ac.jp

Hideo Joho
Faculty of Library, Information and Media Science, University of Tsukuba
Tsukuba, Ibaraki, Japan
hideo@slis.tsukuba.ac.jp

Sumio Fujita
Yahoo Japan Corporation
Tokyo, Japan
sufujita@yahoo-corp.jp

ABSTRACT

In this paper, we present TKB48's methods and submitted runs for the TREC Conversational Assistance Track of Y3. We incorporated dense retrieval methods into the conversational task. We leveraged a Dual-encoder structure[2] to encode the user's utterance together with the conversation context and each document of the corpus into dense vector representation. After embedding we computed their relevance score by the dot product of the dense vectors. Our four submitted runs show an competitive performance compared to a sparse retrieval model. In addition to the submitted runs, we further conducted experiments and created two unofficial runs, which followed ConvDR's [29] strategy and trained the conversational dense retrieval model and performed inference on CAsT21 dataset. The results of these two unofficial runs show an effective use of multiple loss functions for conversational search.

KEYWORDS

Conversational search, Dense retrieval

1 INTRODUCTION

With the development of natural language processing technologies and intelligent mobile devices (like Apple Siri on iPhone, Amazon Echo, etc.), intelligent conversational assistants have played an essential role in people's everyday lives by assisting users with various tasks through spoken or text dialogues. Along with this trend, the IR community also pay strong attention to such research and thus dialogue system aiming to satisfy users' information needs and perform conversational information seeking (CIS), e.g., conversational search comes out and becomes one of the most noticeable research areas in IR [4, 7]. Conversational Assistance Track (CAsT) of TREC held from 2019 is an initiative to facilitate conversational information seeking research and aims to create a large-scale reusable test collection for conversational search systems [7]. This year has been the third year of this track, and a large number of excellent studies have shown up during the last two years and made significant progress for conversational search research [6, 7].

CAsT defines conversational search as a retrieval task in a conversational context [7]. The primary initial focus is on system understanding of information needs in a conversational format and finding relevant responses. In conversational search, users' utterances are usually ambiguous with various linguistic phenomena including anaphora, ellipsis, etc. [25]. For the previous two years' task, the proposed studies mainly leveraged a multi-stage pipeline framework for such conversational search task [17], including 1) conversational query reformulation and rewriting [16, 25], 2) first-stage retrieval using traditional IR models like BM25, 3) reranking with a fine-tuned neural language model. By query reformulation or rewriting, the ambiguous utterances of users are reformulated and rewritten to decontextualized queries with omitted information supplemented and thus can be directly processed by a search engine. While such methods have been proven to be very effective in previous TREC overview [6, 7], they still have some unsolved problems. First, a multi-stage pipeline framework comprises multiple pre-trained transformer-based language models for conversational query rewriting (GPT-2, BART, T5, etc.) and reranking (BERT, ALBERT, T5, etc.) and thus spend a long time for inferencing. Second, although such methods leverage conversational query rewriting to decontextualize ambiguous utterances, they still stay on the lexical level of query understanding and cannot resolve vocabulary mismatching problems [29]. Finally, such methods serve query understanding and retrieval as individual stages and optimize them separately, which may be stuck into a local optimum rather than achieve the global optimum for the whole task [15].

In recent years dense retrieval technologies have developed rapidly and provided a novel way for resolving IR tasks. It has achieved remarkable processes in QA and ad hoc retrieval [12, 27]. Such a system usually adopts a Dual-encoder structure which includes a query encoder that encodes queries into high-dimension dense vectors and a document encoder that encodes each document of the corpus into dense vectors of the same high-dimension vector space. The relevant score is then computed as the dot product or cosine similarity of the query embedding and document embedding. Such a dense retrieval method can directly learn the encoder model for query understanding and relevant document retrieving end-to-end [12]. By embedding the query into a dense vector, it performs query understanding at a semantic level and thus avoids the problem of vocabulary mismatching. This paper also tries incorporating dense retrieval technologies into conversational search to better capture users' information needs.

This paper describes our work for TREC CAsT track year 3. Our approach incorporates dense retrieval techniques into the conversational search for the task. We encode the user's current utterance with conversation contexts into a dense vector representation. After that, we retrieve and compute relevance score by computing the dot product of query embedding and document embedding, which is computed in previous instead of based on bag-of-words representation and TF-IDF score, in order to better understand user's information needs on a semantic level and avoid vocabulary mismatching problem. The rest of this paper presents our methodology and experiments results for the task.

2 RELATED WORK

2.1 Conversational Search

Research on the conversational search and interactive information retrieval has been conducted since the 1980s [1, 3]. Recently, Radlinski and Craswell [23] proposed a theoretical framework for conversational search, which has presented a theory and model of information interaction in a chat setting and designed some basic formulas and attributes of the conversational system. Zhang [30] proposed a unified conversational search/recommendation framework and trained a Multi-Memory Network that accomplished it. Trippas [24] conducted a laboratory-based observational study and concluded that the spoken conversational search paradigm is much more complex and interactive. From 2019, TREC started Conversational Assistance Track (CAsT) [6, 7] which aims to create a reusable benchmark for open-domain information-centric conversation dialogues. Most previous studies leveraged transformer-based pre-trained language models for query rewriting in order to rewrite and decontextualize the user‘s current turn’s utterance and degenerate the conversational search task to ad hoc information retrieval task [17]. Lin [16] presented an empirical study of conversational question reformulation with sequence-to-sequence architectures and pre-trained language models. Vakulenko [25] addressed the conversational QA task by decomposing it into question rewriting and question answering subtasks and employing a unidirectional Transformer decoder [22] for both encoding the input sequence and decoding the output sequence. Besides the methods of treating the query rewriting task as a sequence-to-sequence task, [26] modeled the query resolution task as a binary term classification problem and proposed a neural query resolution model based on bidirectional transformers for the task. Yang [28] proposed both a rule-based method and a pre-trained language model-based method to extract knowledge from historical dialogues. These studies proposed various methods for query rewriting, while none of them solved the vocabulary mismatching problem, which is significant in conversational search.

2.2 Dense Retrieval

With the development of deep learning, various neural ranking models have come up over the past few years like DRMM [10], KNRM [5] and Duet [18]. Such models embed queries and documents into a learned dense vector space and directory compute their relevance by modeling local interactions of their vector representations. In recent years with the development of pre-trained language models like ELMo [20], and BERT [9], many dense retrieval methods fine-tuning pre-trained language models for estimating relevance emerged and made significant progress in various IR tasks. Khattab [13] presents a novel ranking model ColBERT that adapts BERT for efficient retrieval by introducing a late interaction architecture. Karpukhin [12] proposed the DPR model, which leveraged BERT pre-trained model and a dual-encoder [2] architecture for open-domain question answering tasks. Xiong [27] proposed ANCE, which is a novel approximate nearest neighbor negative contrastive learning mechanism that selects hard training negatives globally from the entire corpus using an asynchronously updated ANN index for passage index.

Not only in ad hoc IR tasks, but dense retrieval has also emerged in recent years‘ conversational search research. Lin [15] adopt a Dual-encoder model and propose to teach a pre-trained standalone query encoder to encode each user utterance alone with its conversational context into contextualized query embeddings for dense retrieval serving the scenario of conversational search. Yu [29] presented a conversational dense retrieval system that learns contextualized embeddings for multi-turn conversational queries and retrieves documents solely using embedding dot products. Such methods embedded users' utterances and the conversation context into dense vector representations, thus resolving the vocabulary mismatching problem and better understanding users‘ information needs on a semantic level. This paper also leverages the dense retrieval method in the conversational search task.

3 METHODOLOGY

For this year's CAsT track, we leveraged dense retrieval for the conversational search task. We submitted four runs, including three runs using manual rewritten utterances and one run using automatic rewritten utterances. Our system adopted a Siamese/Dual Encoder structure[2] for the passage and query embedding and relevant passage retrieval. The following section describes our dual-encoder dense retrieval system and runs we created.

3.1 Dual-encoder Dense Retrieval

To perform dense retrieval, usually, there are two separate encoders for both the query embedding and passage embedding, which constructs a Dual-encoder structure [12, 27]. The goal of the passage encoder is to map each passage in the corpus to a d-dimensional dense vector of a continuous vector space and then build an approximate nearest neighbor index for relevant passage searching [27]. After embedding the corpus and building the ANN index, the query encoder maps each query into a d-dimensional dense vector of the same continuous vector space where the similarity of the query and the passage can be computed easily by the Euclidean distance or dot product of their vector representation:

$$sim(q, p) = E_Q(q)^T E_P(p) \tag{1}$$

For the conversational task, the input of the query encoder is the user's current utterance and the conversation history since the user's information need usually depends on the whole conversation context and may have some omitted information appearing in the previous turn's utterance. At the same time, the passages are directly encoded by the passage encoder, which is the same as ad hoc IR task since the information represented by the dense vector will not be changed either for ad hoc task or conversational search task:

$$E_Q(q) = QueryEncoder(u_0 \oplus u_1 \oplus ... \oplus u_{cur}) \tag{2}$$

$$E_P(p) = PassageEncoder(p) \tag{3}$$

For four submitted runs, we adopt the DPR [12] model as our query encoder and passage encoder and Faiss [11] to build the ANN index. We adopted Pyserini [14] for constructing the whole system.

3.2 Runs

We submitted four runs produced by our system for this year's task.

sparse_manual. this run was produced using the traditional sparse retrieval method and the manual rewritten utterances. We adopted Pyserini's default BM25 setting to construct the searcher and retrieved the top 1000 results for each turn of each topic.

dense_manual. This run was produced using our constructed bi-encoder dense retrieval system and the manual rewritten utterances. We adopted Pyserini's build-in DPR document encoder setting for embedding and constructing the ANN index and constructing the dense searcher based on the DPR query encoder provided by Pyserini. Like the *sparse_manual* run, we also retrieved the top 1000 results for each turn of each topic.

hybrid_manual. For this run, we adopted the hybrid search method [14] provided by Pyserini, which searches the corpus using sparse retrieval and dense retrieval and performs weighted interpolation on the individual results to arrive at a final ranking.

bm25_automatic. For this run, we used the automatic rewritten utterance and searched the top 1000 results using the same sparse retrieval setting as *sparse_manual*.

3.3 Datasets

TREC CAsT 2021 dataset. The text collection of this' years task is similar to previous years [8], while in this year, document collections are used instead of passage collection. The text collection is a combination of three data sources, including KILT [21] which is a benchmark for knowledge-intensive language tasks that are grounded in the same snapshot of Wikipedia, MS MARCO Document Ranking data [19], and TREC Washington Post V4 (WaPo V4). The detailed statics of each data source can be seen in Table 1.

Each document of the collection was split into passage segmentation using tools in the TREC CAsT tools repository with fixed sentence boundaries. Duplicate handing was performed on WaPo V4 and MS MARCO in order to remove duplicates in the corpus. The test topics are the same with Y1 [8], which includes raw utterances, utterances rewritten using automatic query rewriter, and utterances rewritten manually by a human. For Y3, a canonical document and text passage from the document is also provided as context for each turn.

3.4 Experiment Setup

Document Embedding. In order to perform dense retrieval for the task, first, it needs to embed each document of the text collection into a continuous dense vector representation to construct an ANN index. We adopt Pyserini's built-in DPR document encoder for the four submitted runs to do this job. To accelerate the document embedding process, we divided the whole corpus into four shards, embedded them individually, and merged them into the final results for constructing the ANN index.

Indexing. After the document embedding process, the ANN index was constructed above the embedded dense vector using Faiss. For the submitted four runs, the whole progress of the document embedding process in shard, merging the sub results, and indexing was directly performed using Pyserini's dindex command. It should be noted that we also constructed a spare index using Pyserini's index command for sparse retrieval and hybrid retrieval.

Document Retrieval and Ranking. We used Pyserini's searching API for both sparse retrieval and dense retrieval. We searched top1000 results for each query on a pre-constructed sparse index for sparse runs. We used the same encoder setting for dense runs, e.g., Pyserini's built-in DPR encoder for query embedding, and searched top1000 results on constructed ANN index. For the hybrid run, we initiated the hybrid searcher using the same setting with sparse retrieval and dense retrieval to search results in both ways and perform the fusion.

4 RESULTS

The results of submitted runs can be seen in Table 2. The evaluation metrics are ndcg@3, ndcg@5, ndcg@500, and ap@500. We compared the results of our submitted runs with the median of each metric across all submitted runs of participants.

The table shows that when considering precision-oriented metrics, namely ndcg@3 and ndcg@5, using a hybrid retrieval method that combines results from sparse retrieval and dense retrieval achieves the best performance among all three manual runs. While for recall-oriented metrics like ndcg@500 and ap@500, the sparse retrieval method outperforms all other methods either purely based on dense retrieval or in a hybrid way.

When comparing to the median score of all submitted runs, all of our submitted runs perform worse than the baseline for each metric. It may be because we used the default setting for both sparse and dense retrieval models, and did not optimize them to fit the specific task. We used the default Pyserini's built-in DPR encoder trained for QA tasks for the dense retrieval method. We did not fine-tune it for our conversational passage retrieval so that it could learn to better understand the task and performs well.

5 UNOFFICIAL RUNS

5.1 Method

In addition to the submitted runs, we developed two unofficial runs. We followed the ConvDR [29] teacher-student framework to train the conversational query encoder with an ad-hoc teacher. we then fine-tuned it using CAsT21's datasets. Similar to ConvDR's experiments on CAsT20, we fixed the document embedding computed from the ANCE checkpoint. At the same time, for training the ConvDR query encoder, we used BM25 to sample negative passages and generate the training data instead of ANCE, which is used in ConvDR's experiment, to better fit the document corpus of CAsT21. In ConvDR, they proposed a teacher-student framework that use ad hoc query encoder ANCE as the teacher to train their conversational query encoder with MSE loss in order to solve the relevance-oriented supervision signals limitation in conversational search task, which is represented as below:

$$E_{adhoc}(q^*) = AdHocQueryEncoder(q^*) \tag{4}$$

$$\mathcal{L}_{MSE} = MSE(E_Q(q), E_{adhoc}(q^*)) \tag{5}$$

Given the query embedding $E_{adhoc}(q^*)$ obtained from an ad hoc dense retrieval encoder E_{adhoc} on manual oracle query q^*, the conversational query encoder E_Q is trained by computing the MSE loss between the conversational query embedding $E_Q(q)$ and the manual oracle embedding $E_{adhoc}(q^*)$. In this paper, we followed ConvDR's strategy of using ANCE as the ad hoc teacher. Besides training on MSE loss, ConvDR also proposed to combine it with the NLL loss, which is to optimize the model to learn retrieval-oriented

Table 1: Text Collection Info

datasource	contents
KILT	Approximately 5 Million articles
MS MARCO Document Ranking	3.2 million documents from Bing search
WaPo V4	728,626 news articles from the WaPo from 2012-2020

Table 2: Results of Submitted Runs

run	ndcg_cut_3	ndcg_cut_5	ndcg_cut_500	ap_cut_500
dense_manual	0.4172	0.4032	0.4277	0.1832
sparse_manual	0.4069	0.3981	**0.5103**	**0.2580**
hybrid_manual	**0.4380**	**0.4237**	0.4670	0.2024
bm25_automatic	0.3174	0.3072	0.4049	0.1885
median	0.5547	0.5503	0.6120	0.3714

representations:

$$\mathcal{L}_{Rank} = -log\frac{exp(E_Q(q) \cdot E_p(p^+))}{exp(E_Q(q) \cdot E_p(p^+) + \sum_{p^- \in P^-} exp(E_Q(q) \cdot E_p(p^-))} \quad (6)$$

which was proved effective for supervised-learning settings when there were enough training data while degrading the performance for a few-shot setting. Differing from what ConvDR has done, in this paper, we trained the model not by combining the NLL loss and MSE loss as multi-task learning but by first training the ANCE checkpoint using NLL loss and then doing the warm-up and training it using KD loss, namely in a *sequential* way. We found that by training the ConvDR model in this way it could better learn from both the NLL loss and MSE loss and thus improve its ability to retrieve relevant documents. The results of these two unofficial runs show that we can achieve better performance by performing training in this way.

5.2 Runs

ConvDR_KD. This run was created using raw utterances and our trained ConvDR model on CAsT21, which is trained following the ConvDR's original strategy that adopted ANCE as ad hoc teacher and trained the model with KD Loss.

ConvDR_SEQ. This run was created using the ConvDR model that first trained using NLL Loss as initialization and then did the warm-up using OR-QuAC and trained using KD loss as *ConvDR_KD*.

5.3 Experiments

For *ConvDR_KD* and KD-Loss training part of *ConvDR_SEQ*, we started from the ANCE checkpoint and first warmed it up using OR-QuAC, after which we continued to train it using KD-Loss and set ANCE as the ad hoc teacher following ConvDR's strategy. For *ConvDR_SEQ*, the ANCE checkpoint was directly trained with NLL-Loss, and BM25 sampled negative samples. The intermediate model was then warmed up with OR-QuAC and fine-tuned with KD-Loss. For NLL-Loss, we did the negative sampling using BM25 to create the training data, which differs from ConvDR's original experiments that used ANCE. In our experiments, the BM25 negative sampling usually performed better than ANCE.

For document embedding, We followed ConvDR's setting that used ANCE to embed the document and fixed it for the two unofficial runs. Due to memory limitation, we separated the whole corpus into two parts and embedded them separately. When searching, we searched the separated ANN index individually and performed interpolation fusion using Pyserini to get the final result.

The results of unofficial runs are in Table 3.The results of each of the two parts of the whole collection were evaluated separately, as well as a fused one of these two sub results, which represented the general performance. The evaluation metrics included nDCG@3 and MRR. The first row represents the results of *ConvDR_KD* that was trained only using KD-Loss.The second row represents results of *ConvDR_SEQ* that was first trained using NLL-loss followed by the warm-up and KD-Loss. The third row is the offical convdr run *org_convdr* that used the ConvDR model trained on CAsT20 with KD loss. From the table, it can be seen that *ConvDR_SEQ* outperformed *ConvDR_KD* with 27.7% improving on nDCG@3 and 14.6% on MRR for CAsT21 and achieved a competitive performance with resepect to the official run on nDCG@3 and outperformed on MRR, which demonstrated the effectiveness of our proposed approach.

6 DISCUSSION

In this paper, we focused on incorporating dense retrieval methods into conversational search tasks. Dense retrieval can embed both query and document into the same continuous vector space and computed their similarity as the distance between two vectors, which could better understand the contents in semantic level and avoid vocabulary mismatching problems, thus potentially outperforming the traditional sparse method. From our experiment results of Table 2 and Table 3, dense retrieval methods demonstrated their effectiveness to some extents. Although we did not optimize the encoder model for the submitted official runs, it still has a better performance for ndcg@3 and ndcg@5 compared to the sparse method.

Table 3: Results of Unofficial Runs

RUN	nDCG@3			MRR		
	Part1	Part2	Fusion	Part1	Part2	Fusion
ConvDR_KD	0.2414	0.2825	0.2890	0.4921	0.5447	0.5567
ConvDR_SEQ	0.3066	0.3525	**0.3691**	0.5979	0.6433	**0.6382**
org_convdr	0.361			0.505		

Our fine-tuned ConvDR model has achieved competitive results for our two unofficial runs compared to the official ConvDR run.

7 CONCLUSION

In this paper, we presented our method that incorporated dense retrieval models into conversational search tasks for the TREC CAsT track. The dense retrieval method adopts a dual-encoder structure that uses a document encoder to embed each document of the corpus into a d-dimensional dense vector representation and construct an ANN index, and using a query encoder to embed each query into the same d-dimensional dense vector. It performs retrieval based on dot product, which can better understand user's information needs on a semantic level and improve search results. While our submitted manual runs were weaker than other submitted ones, the further experimental results of the two unofficial runs show an effective use of multiple loss functions which can be useful for few shot settings.

ACKNOWLEDGMENTS

This work was partly supported by JSPS KAKENHI Grant Number 19H04418.

REFERENCES

[1] Nicholas J Belkin. 1980. Anomalous states of knowledge as a basis for information retrieval. *Canadian journal of information science* 5, 1 (1980), 133–143.
[2] Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence* 7, 04 (1993), 669–688.
[3] Helen M Brooks and Nicholas J Belkin. 1983. Using discourse analysis for the design of information retrieval interaction mechanisms. In *Acm sigir forum*, Vol. 17. ACM New York, NY, USA, 31–47.
[4] J Shane Culpepper, Fernando Diaz, and Mark D Smucker. 2018. Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in lorne (swirl 2018). In *ACM SIGIR Forum*, Vol. 52. ACM New York, NY, USA, 34–90.
[5] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 126–134.
[6] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. CAsT 2020: The Conversational Assistance Track Overview. In *TREC*.
[7] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The Conversational Assistance Track Overview. *CoRR* abs/2003.13624 (2020). arXiv:2003.13624 https://arxiv.org/abs/2003.13624
[8] Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. 2020. Cast-19: A dataset for conversational information seeking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1985–1988.
[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[10] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. 55–64.
[11] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).
[12] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. (2020), 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550 arXiv:2004.04906
[13] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.
[14] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use Python toolkit to support replicable IR research with sparse and dense representations. *arXiv preprint arXiv:2102.10073* (2021).
[15] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. Contextualized Query Embeddings for Conversational Search. (2021). arXiv:2104.08707 http://arxiv.org/abs/2104.08707
[16] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational Question Reformulation via Sequence-to-Sequence Architectures and Pretrained Language Models. In *Arxiv.org*. 2–6. arXiv:2004.01909 http://arxiv.org/abs/2004.01909
[17] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Multi-Stage Conversational Passage Retrieval: An Approach to Fusing Term Importance Estimation and Neural Query Rewriting. (2020), 1–2. arXiv:2005.02230 http://arxiv.org/abs/2005.02230
[18] Bhaskar Mitra and Nick Craswell. 2019. An updated duet model for passage re-ranking. *arXiv preprint arXiv:1903.07666* (2019).
[19] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.
[20] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).
[21] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252* (2020).
[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
[23] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*. 117–126.
[24] Johanne R Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the design of spoken conversational search: Perspective paper. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*. 32–41.
[25] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. Question rewriting for conversational question answering. *arXiv* (2020), 0–8. arXiv:2004.14652
[26] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query Resolution for Conversational Search with Limited Supervision. In *Arxiv.org*. https://doi.org/10.1145/3397271.3401130 arXiv:2005.11723
[27] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk Microsoft. [n. d.]. APPROXIMATE NEAREST NEIGHBOR NEGATIVE CON-TRASTIVE LEARNING FOR DENSE TEXT RETRIEVAL. ([n. d.]). arXiv:2007.00808v2 https://aka.ms/ance.

[28] Jheng-hong Yang, Sheng-chieh Lin, Jimmy Lin, Ming-feng Tsai, and Chuan-ju Wang. [n. d.]. Query and Answer Expansion from Conversation History. ([n. d.]), 1–4.

[29] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-Shot Conversational Dense Retrieval. In *SIGIR'21*. 829–838. https://doi.org/10.1145/3404835.3462856 arXiv:2105.04166

[30] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*. 177–186.