A Test Collection of Synthetic Documents for Training Rankers: ChatGPT vs. Human Experts

Arian Askari
a.askari@liacs.leidenuniv.nl
Leiden University

Mohammad Aliannejadi
m.aliannejadi@uva.nl
University of Amsterdam

Evangelos Kanoulas
e.kanoulas@uva.nl
University of Amsterdam

Suzan Verberne
s.verberne@liacs.leidenuniv.nl
Leiden University

ABSTRACT

In this resource paper, we investigate the usefulness of generative Large Language Models (LLMs) in generating training data for cross-encoder re-rankers in a novel direction: generating synthetic documents instead of synthetic queries. We introduce a new dataset, ChatGPT-RetrievalQA, and compare the effectiveness of strong models fine-tuned on both LLM-generated and human-generated data. We build ChatGPT-RetrievalQA based on an existing dataset, human ChatGPT Comparison Corpus (HC3), consisting of public question collections with human responses and answers from ChatGPT. We fine-tune a range of cross-encoder re-rankers on either human-generated or ChatGPT-generated data. Our evaluation on MS MARCO DEV, TREC DL'19, and TREC DL'20 demonstrates that cross-encoder re-ranking models trained on LLM-generated responses are significantly more effective for out-of-domain re-ranking than those trained on human responses. For in-domain re-ranking, the human-trained re-rankers outperform the LLM-trained re-rankers. Our novel findings suggest that generative LLMs have high potential in generating training data for neural retrieval models and can be used to augment training data, especially in domains with smaller amounts of labeled data. We believe that our dataset, ChatGPT-RetrievalQA, presents various opportunities for analyzing and improving rankers with human and synthetic data. We release our data, code, and model checkpoints for future work.[1]

CCS CONCEPTS

• **Information systems → Learning to rank**; **Novelty in information retrieval**.

KEYWORDS

Large language models, Document Generation, Cross-encoder re-rankers

[1]https://github.com/arian-askari/ChatGPT-RetrievalQA

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615111

1 INTRODUCTION

Generative large language models (LLMs) such as GPT-3 [5] and GPT-3.5 (ChatGPT) have shown remarkable performance in generating realistic text outputs for a variety of tasks such as summarization [42], machine translation [28], sentiment analysis [34, 38], retrieval interpretability [22], and stance detection [41]. Although ChatGPT can produce impressive responses, it is not immune to errors or hallucinations [14]. Furthermore, the lack of transparency in the source of information generated by ChatGPT can be a bigger concern in domains such as law, medicine, and science, where accountability and trustworthiness are critical [1, 6, 30, 33].

Ranking models, as opposed to generative models, retrieve information from existing sources (i.e., documents) and search engines provide the source of each retrieved item [31]. This is why document retrieval – even when generative LLMs are available – remains an important application, especially in situations where reliability is vital. One potential purpose of generative LLMs in information retrieval (IR) is to generate training data for retrieval models. Data generated with generative LLMs can be used to augment training data, especially in domains with smaller amounts of labeled data.

InPars [3], Promptagator [10], InPars-light [4], and InPars-v2 [17] have utilized LLMs to generate synthetic queries given documents. Particularly, InPars-v2 [17] achieves state-of-the-art results on the BEIR dataset in an out-of-domain setting by using an open-source language model, GPT-J-6B [35] and a powerful external re-ranker, MonoT5-MSMARCO [26] to filter the top-10k high-quality pairs of synthetic query-document pairs for data augmentation. In contrast, we generate *documents* (passages) by ChatGPT given a query — as opposed to generating queries in InPars-v2. To the best of our knowledge, augmenting data via generating documents for given queries has not been explored in prior work.

We believe that exploring this reverse direction is important as it allows us to augment training data based on the data that originates from user behavior (i.e., user queries) rather than the (static) document collection itself. This can improve the effectiveness of re-rankers by augmenting the training data with synthetic documents according to the queries that actual search engine users are searching for, increasing the diversity of the training data, while allowing the rankers to better generalize to new queries.

ChatGPT-RetrievalQA dataset is based on the [15] that use public question-answering datasets and prompts the questions to the ChatGPT user interface for generating answers. The goal of the HC3 dataset is to linguistically compare human and ChatGPT responses and explore the possibility of distinguishing between responses

Table 1: Statistic on the size of Train, Validation, and Test sets across domains for evaluation of cross-Encoders.

Domain	# of queries		
	Train set	Validation set	Test set
All	16788	606	6928
Medicine: Meddialog [7]	862	31	355
Finance: FiQA [24]	2715	98	1120
Reddit: ELI5 [13]	11809	427	4876
Wikipedia: openQA [40]	820	29	338
Wikipedia: csai [15]	582	21	239

generated by ChatGPT and those written by humans. The HC3 dataset contains questions (i.e., queries) from four different domains: medicine (Medical Dialog [7]), finance (FiQA [24]), Wikipedia (WikiQA [40] and Wiki_csai [15]), and Reddit (ELI5 [13]). While there is no study on generating documents to augment training data, a more recent study, QuerytoDoc [36], generates documents given a query and appends the generated document to the query for expanding the query, which is out of the scope of augmenting data for information retrieval. Furthermore, there are various recent studies on ChatGPT with a focus on ranking and retrieval, however, to the best of our knowledge, none of them focus on data augmentation by generating relevant documents. Examples of recent studies are the one by Faggioli et al. [12], who study if ChatGPT can be used for generating relevance labels, and Sun et al. [32], who assess whether ChatGPT is good at searching by giving it a query and a set of candidate documents to re-rank.

While there are various possible studies that can be done on ChatGPT-RetrievalQA, in this resource paper, we focus on analyzing two main research questions based on the ChatGPT-RetrievalQA dataset: **(RQ1)** *How does the effectiveness of cross-encoder re-rankers fine-tuned on ChatGPT-generated responses compare to those fine-tuned on human-generated responses in both in-domain and out-of-domain settings?*; **(RQ2)** *How does the effectiveness of using ChatGPT for generating relevant documents differ between specific and general domains?*

Leveraging ChatGPT-RetrievalQA, we aim to shed light on the potential of using LLMs for data augmentation in cross-encoder re-rankers and the domain dependency of their effectiveness via answering the research questions. Our primary experimental setup involves using $\text{CE}_{\text{ChatGPT}}$[2] for inference (i.e., re-ranking task) on human-generated responses. Through our dataset and analysis, we aim to provide valuable insights into the advantages and limitations associated with the utilization of generative LLMs for augmenting training data in retrieval models.

Our main contributions in this work are three-fold: (i) We release the ChatGPT-RetrievalQA dataset, which is designed specifically for information retrieval tasks in both full-ranking and re-ranking setups. This dataset contains 24,322 queries, 26,882 responses generated by ChatGPT, and 58,546 human-generated responses. (ii) To perform benchmarking, we fine-tune cross-encoder re-rankers on both the human- and ChatGPT-generated responses, evaluating their performance on our dataset in an in-domain setting. We also

[2]We refer to the cross-encoders fine-tuned on ChatGPT-generated and human-generated responses as $\text{CE}_{\text{ChatGPT}}$ and CE_{human}, respectively.

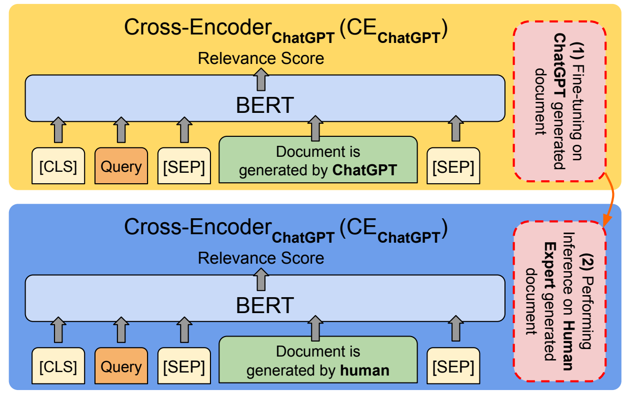

Figure 1: Our main experimental setup involves the above steps.

show the effectiveness of the ChatGPT-trained models in an out-of-domain evaluation on the MS MARCO-passage collection and the TREC Deep Learning tracks. (iii) We conduct an analysis of the effectiveness of ChatGPT-trained cross-encoders on different domains and show that human-trained models are slightly more effective in domain-specific tasks, e.g., in the medicine domain. Our novel findings highlight the potential of using generative LLMs like ChatGPT for generating high-quality responses in information retrieval tasks in order to create training datasets.

2 METHODOLOGY

2.1 Dataset and pool preparation

Our ChatGPT-RetrievalQA dataset is based on the HC3 dataset produced by Guo et al. [15], which contains 24,322 queries and 26,882 ChatGPT-generated responses, as well as 58,546 human-generated responses. There is on average one ChatGPT-generated and 2.4 human-generated response per query. For pool preparation and ranking experiments (an experimental setup different from [15]), we convert the dataset files to a format similar to the MSMarco dataset [25], in both full-ranking and re-ranking setups.[3] We divide the data into training, validation, and test sets. To facilitate training, we provide training triples files in TSV format, including both textual and ID-based representations, where the structure of each line is composed of 'query, positive passage, negative passage' and 'qid, positive pid, negative pid'. We consider the actual response by ChatGPT or human as the relevant answer and we randomly sample 1000 negative answers for each query similar to MS MARCO. In addition, we provide the top 1000 documents, ranked by BM25, per query to enable re-ranking studies. Table 1 shows the size of the train, validation, and test sets for each domain.

2.2 First-stage ranker: BM25

Lexical retrievers use word overlap to produce the relevance score between a document and a query. Several lexical approaches have been developed in the past, such as vector space models, Okapi BM25 [29], and query likelihood. We use BM25 as the first-stage ranker because of its popularity and effectiveness. BM25 calculates a score for a query–document pair based on the statistics of the

[3]This allows for easy reuse of available scripts on MS MARCO.

Table 2: Comparing the effectiveness of cross-encoder re-rankers fine-tuned on human and ChatGPT responses in in-domain and out-of-domain settings. † indicates that a CE achieves statistically significant improvement for that dataset among all of the cross-encoder re-rankers and BM25 on that dataset. Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing. The cutoff for MAP, NDCG, and MRR are 1000, 10, and 10.

	In-domain setting			Out-of-domain setting								
Model	ChatGPT-RetrievalQA **(Ours)**			TREC DL'19			TREC DL'20			MS MARCO DEV		
	MAP	NDCG	MRR	MAP	NDCG	MRR	MAP	NDCG	MRR	MAP	NDCG	MRR
BM25	.143	.184	.240	**.377**	.506	.858	.286	.480	.819	.195	.234	.187
$\text{MiniLM}_{\text{human}}$	**.310†**	**.384†**	**.460†**	.326	.451	.833	.269	.376	.913	.130	.155	.118
$\text{MiniLM}_{\text{ChatGPT}}$	.294	.362	.444	.342†	**.510†**	.903	**.344†**	**.539†**	**.978†**	**.226†**	**.267†**	**.218†**
$\text{TinyBERT}_{\text{human}}$	.244	.310	.367	.294	.360	.741	.277	.364	.791	.128	.154	.116
$\text{TinyBERT}_{\text{ChatGPT}}$	.231	.291	.358	.328	.488	**.942†**	.303	.460	.972	.194	.231	.185

words that overlap between them:

$$s_{lex}(q,d) = BM25(q,d) = \sum_{t \in q \cap d} rsj_t \cdot \frac{tf_{t,d}}{tf_{t,d} + k_1\{(1-b) + b\frac{|d|}{l}\}} \quad (1)$$

where t is a term, $tf_{t,d}$ is the frequency of t in document d, rsj_t is the Robertson-Spärck Jones weight [29] of t, and l is the average document length. k_1 and b are parameters.

2.3 Cross-encoder re-rankers

The common approach to employ pre-trained Transformer models with a cross-encoder architecture in a re-ranking setup is by concatenating the input sequences of query and passage. This method, known as MonoBERT or CE_{CAT}, is illustrated in Figure 1 and has been utilized in several studies. In CE_{CAT}, the sequences of query words $q_1 : q_m$ and passage words $p_1 : p_n$ are joined with the [SEP] token, and the ranking model of CE_{CAT} calculates the score for the representation of the [CLS] token obtained by cross-encoder (CE) using a single linear layer W_s:

$$CE_{CAT}(q_{1:m}, p_{1:n}) = CE([CLS]\ q\ [SEP]\ p\ [SEP]) * W_s \quad (2)$$

We use CE_{CAT} as our cross-encoder re-ranker with a re-ranking depth of 1000. In our experiments, both $\text{CE}_{\text{ChatGPT}}$ and CE_{human} follow the above design.

3 EXPERIMENTAL DESIGN

Evaluation setup. We conduct our out-of-domain evaluation experiments on the MS MARCO-passage collection [25] and the data from two TREC Deep Learning tracks (TREC-DL'19 and DL'20) [8, 9]. To make our results comparable to previously published and upcoming research, we use standard IR metrics for evaluation, namely, MAP@1000, NDCG@10, and MRR@10 [8, 9]. The MS MARCO-passage dataset contains about 8.8 million passages (average length: 73.1 words) and about 1 million natural language queries (average length: 7.5 words) and has been extensively used to train deep language models for ranking. Following prior work on MS MARCO [19, 21, 23, 44, 45], we only use the dev set ($\sim 7k$ queries) for our empirical evaluation. The TREC DL'19 and DL'20 collections share the passage corpus of MS MARCO and have 43 and 54 queries respectively with much more relevant documents per query. We measure the same metrics in the in-domain setting on the test set of ChatGPT-RetrievalQA. The average length of human responses is 142.5 words and 198.1 words for ChatGPT in the ChatGPT-RetrievalQA dataset.

Training configuration. We use the Huggingface library [39], and PyTorch [27] for the cross-encoder re-ranking training and inference. Following prior work [16], we use the Adam [20] optimizer with a learning rate of $7 * 10^{-6}$ for all cross-encoder layers, regardless of the number of layers trained. We use a training batch size of 32. For all cross-encoder re-rankers, we use the cross-entropy loss [43]. For the lexical retrieval with BM25, we use the similarity function of Elasticsearch [11]. We cap the query length at 30 tokens and the passage length at 200 tokens following prior work [2, 16].

4 RESULTS

4.1 Main results (RQ1)

Table 2 shows a comparison of the effectiveness of CE_{human} and $\text{CE}_{\text{ChatGPT}}$. Please note that for both models, during inference, we evaluate their effectiveness in retrieving *human* responses in the in-domain or out-of-domain settings. We choose MiniLM (w/ 12 layers) [37] for the experiments due to its competitive results in comparison to BERT re-ranker [2] while being three times smaller and six times faster. In addition, we conduct experiments with TinyBERT (w/ 2 layers) [18] to assess the generalizability of our results. In the **in-domain setting** where we evaluate on the test set queries with human documents of our ChatGPT-RetrievalQA dataset, $\text{MiniLM}_{\text{human}}$ significantly outperforms all other cross-encoder re-rankers. Although performing worse than the human-trained models, $\text{MiniLM}_{\text{ChatGPT}}$ and $\text{TinyBERT}_{\text{ChatGPT}}$ still outperform the strong baselin [3], BM25 [29], statistically significantly by a large margin in this setting. In the **out-of-domain setting**, the $\text{MiniLM}_{\text{ChatGPT}}$ consistently outperforms the other cross-encoder re-rankers including $\text{MiniLM}_{\text{human}}$ and BM25 significantly across the TREC DL'20 and MS MARCO DEV. However, on TREC DL'19, BM25 achieves the highest effectiveness for MAP@1000, $\text{MiniLM}_{\text{ChatGPT}}$ for NDCG@10, and $\text{TinyBERT}_{\text{ChatGPT}}$ for MRR@10. Overall, we can see the models fine-tuned on ChatGPT-generated responses are significantly more effective in the out-of-domain setting compared to those fine-tuned on human-generated responses.

4.2 Domain-level re-ranker effectiveness (RQ2)

Table 3 shows the effectiveness of $\text{MiniLM}_{\text{human}}$ and $\text{MiniLM}_{\text{ChatGPT}}$ re-rankers in the in-domain settings – on the test set of our dataset – across all of the domains including Medicine, Finance, Reddit, and Wikipedia. Overall, the results show that $\text{MiniLM}_{\text{human}}$ achieves

Table 3: Comparing the effectiveness of CE_C and CE_H in in-domain setting across different domains where CE, C, and H refer to the MiniLM, human, and ChatGPT. The OpenQA and Wiki_csai datasets are in the Wikipedia domain.

Domain	Model	MAP@1K	NDCG@10	MRR@10
All	CE_H	**.310**	**.384**	**.460**
	CE_C	.294	.362	.444
Medicine [7]	CE_H	**.397**	**.419**	**.395**
	CE_C	.379	.400	.377
Finance [24]	CE_H	**.257**	**.399**	**.251**
	CE_C	.250	.368	.245
Reddit [13]	CE_H	**.323**	**.418**	**.543**
	CE_C	.302	.391	.522
OpenQA [40]	CE_H	.322	**.345**	.320
	CE_C	**.331**	.341	**.328**
Wiki_csai [15]	CE_H	.149	.152	.135
	CE_C	**.163**	**.159**	**.144**

higher effectiveness than $\text{MiniLM}_{\text{ChatGPT}}$ for all domains except Wikipedia. However, the difference in effectiveness is small, and $\text{MiniLM}_{\text{ChatGPT}}$ still achieves a reasonable level of effectiveness. In the Finance domain, both $\text{MiniLM}_{\text{human}}$ and $\text{MiniLM}_{\text{ChatGPT}}$ achieve relatively low effectiveness compared to other domains. In the Wikipedia domain, $\text{MiniLM}_{\text{human}}$ and $\text{MiniLM}_{\text{ChatGPT}}$ achieve relatively similar levels of effectiveness. In the Medicine domain, the CE_{human} shows the highest effectiveness. Overall, these results suggest that $\text{MiniLM}_{\text{human}}$ performs more effectively in in-domain settings, particularly in specific domains such as Medicine, even though the difference in performance is small.

5 DISCUSSION

Data overlap. It is worth noting that in the in-domain setting, the collection of documents used for training and testing is shared for CE_{human} re-rankers. Therefore, some documents may be seen during both training and evaluation. This setup is very common when working with human-assessed data, and similar to the MS MARCO dataset [25]. The shared collection could be a potential benefit for CE_{human} re-rankers in the in-domain setting, as the models may have already seen some of the documents during training. To further investigate this hypothesis, it would be worth exploring a different setup in future work in which the collection of documents is completely separated between the training and test sets.

Effectiveness of BM25. Table 4 shows an analysis of the effectiveness of BM25 on human- and ChatGPT-generated responses in the train, and test sets. BM25 is less effective for human-generated responses than for ChatGPT-generated responses on the train and test sets, as evidenced by lower scores for all metrics. We observed the same pattern for the validation set. These results suggest that the task of retrieving human-generated responses is more challenging for BM25 than for ChatGPT-generated responses. This is probably related to the lexical overlap discussed below.

Table 4: Analyzing the effectiveness of BM25 on human/ChatGPT responses in train, validation, and test set.

Split	Source	MAP@1K	NDCG@10	Recall@1K
Test	human	.143	.184	.520
	ChatGPT	.370	.396	.898
Train	human	.158	.202	.560
	ChatGPT	.413	.443	.903

Table 5: Analyzing the effectiveness of CEs_{ChatGPT} on the seen queries of the train set and unseen documents of human-generated documents collection.

Model	MAP@1K	NDCG@10	MRR@10
$\text{MiniLM}_{\text{ChatGPT}}$	.318	.388	.510
$\text{TinyBERT}_{\text{ChatGPT}}$	.254	.318	.420

Queries without label. In Table 5, we investigate a common scenario in real-world search engines where query logs and a collection of human-generated documents are available, and there are not any judged documents for part or all of the query logs. To simulate and analyze this situation, we evaluate CE_{ChatGPT} on the seen queries of the train set and unseen documents of the human-generated documents collection. Table 5 shows that CE_{ChatGPT} rankers are fairly effective in this scenario. Especially, they are more effective than BM25 in the same setup, in that the NDCG@10 for $\text{MiniLM}_{\text{ChatGPT}}$ is 0.388 and 0.202 for BM25 (see the third row of Table 4). This suggests the potential of augmenting training data with generative LLMs for fine-tuning models to effectively re-rank sourced and reliable human-generated documents from the corpus given the query logs where there are no judged documents for the queries.

Lexical overlap. Our data analysis reveals that ChatGPT-generated responses have a slightly higher lexical overlap than human-generated ones with the queries. The average percentage of query words that occur in ChatGPT-generated responses is 34.6%, compared to 25.5% for human-generated ones. The Q1, median, and Q3 are also on average 7% points higher for ChatGPT compared to human responses. We suspect that this higher lexical overlap compared to the human response happens because ChatGPT often repeats the question or query in the response, and it tends to generate lengthier responses compared to human, increasing the chance of repeating query words in the response. It is noteworthy that lexical overlap is not the best indicator of response quality for fine-tuning effective cross-encoders, as there may be cases where responses with low lexical overlap responses compared to hare still relevant and informative, especially in question-answering tasks.

6 CONCLUSION

We present the ChatGPT-RetrievalQA dataset in both full-ranking and re-ranking setups with 24, 322 queries, 26, 882 responses generated by ChatGPT, and 58, 546 human-generated responses. To perform benchmarking, we analyzed the effectiveness of fine-tuning cross-encoders on human-generated responses compared to ChatGPT-generated responses. Our results show that the $\text{cross-encoder}_{\text{ChatGPT}}$

is more effective than cross-encoder$_{\text{human}}$ in the out-of-domain setting while MiniLM$_{\text{human}}$ is slightly more effective in the in-domain setting and this is consistent across different domains. Furthermore, we show that BM25 is less effective on human-generated responses than on ChatGPT-generated responses, indicating that human-generated responses are more challenging to match with queries than ChatGPT-generated responses.

Overall, our findings suggest that ChatGPT-generated responses are more useful than human-generated responses for training effective re-ranker in out-of-domain retrieval, at least based on our dataset and experiments, and highlight the potential of using generative LLMs for generating effective and useful responses for creating training datasets in natural language processing tasks. Our study can be particularly advantageous for domain-specific tasks where relying on LLM-generated output as a direct response to a user query can be risky. Our results confirm that it is possible to train effective cross-encoder re-rankers by training them on ChatGPT-generated responses even for domain-specific queries. Further work is needed to determine the effect of factually wrong information in the generated responses and to test the generalizability of our findings on open-source LLMs.

ACKNOWLEDGMENTS

This work was supported by the EU Horizon 2020 ITN/ETN on Domain Specific Systems for Information Extraction and Retrieval (H2020-EU.1.3.1., ID: 860721).

REFERENCES

[1] Amin Abolghasemi, Suzan Verberne, Arian Askari, and Leif Azzopardi. 2023. Retrievability Bias Estimation Using Synthetically Generated Queries. In *The 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023)*. ACM.

[2] Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. 2023. Injecting the BM25 Score as Text Improves BERT-Based Re-rankers. In *Advances in Information Retrieval*. Springer Nature Switzerland, Cham, 66–83.

[3] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144* (2022).

[4] Leonid Boytsov, Preksha Patel, Vivek Sourabh, Riddhi Nisar, Sayani Kundu, Ramya Ramanathan, and Eric Nyberg. 2023. InPars-Light: Cost-Effective Unsupervised Training of Efficient Rankers. *arXiv preprint arXiv:2301.02998* (2023).

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Marco Cascella, Jonathan Montomoli, Valentina Bellini, and Elena Bignami. 2023. Evaluating the feasibility of ChatGPT in healthcare: an analysis of multiple clinical and research scenarios. *Journal of Medical Systems* 47, 1 (2023), 1–5.

[7] Shu Chen, Zeqian Ju, Xiangyu Dong, Hongchao Fang, Sicheng Wang, Yue Yang, Jiaqi Zeng, Ruisi Zhang, Ruoyu Zhang, Meng Zhou, et al. 2020. Meddialog: a large-scale medical dialogue dataset. *arXiv preprint arXiv:2004.03329* (2020).

[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *arXiv preprint arXiv:2102.07662* (2021).

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[10] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755* (2022).

[11] BV Elasticsearch. 2023. Elasticsearch. *software], version* 6, 1 (2023).

[12] Guglielmo Faggioli, Laura Dietz, Charles Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. 2023. Perspectives on Large Language Models for Relevance Judgment. *arXiv preprint arXiv:2304.09161* (2023).

[13] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. *arXiv preprint arXiv:1907.09190* (2019).

[14] Josh A Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, and Katerina Sedova. 2023. Generative Language Models and Automated Influence Operations: Emerging Threats and Potential Mitigations. *arXiv preprint arXiv:2301.04246* (2023).

[15] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv preprint arXiv:2301.07597* (2023).

[16] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666* (2020).

[17] Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. *arXiv preprint arXiv:2301.01820* (2023).

[18] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

[19] 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.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies* 14, 4 (2021), 1–325.

[22] Michael Llordes, Debasis Ganguly, Sumit Bhatia, and Chirag Agarwal. 2023. Explain like I am BM25: Interpreting a Dense Model's Ranked-List with a Sparse Approximation. *arXiv preprint arXiv:2304.12631* (2023).

[23] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via prediction of importance with contextualization. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 1573–1576.

[24] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*. 1941–1942.

[25] 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*.

[26] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713* (2020).

[27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[28] Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards Making the Most of ChatGPT for Machine Translation. *arXiv preprint arXiv:2303.13780* (2023).

[29] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*. Springer, 232–241.

[30] Malik Sallam, Nesreen Salim, Muna Barakat, and Alaa Al-Tammemi. 2023. ChatGPT applications in medical, dental, pharmacy, and public health education: A descriptive study highlighting the advantages and limitations. *Narra J* 3, 1 (2023), e103–e103.

[31] Mark Sanderson and W Bruce Croft. 2012. The history of information retrieval research. *Proc. IEEE* 100, Special Centennial Issue (2012), 1444–1451.

[32] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *arXiv preprint arXiv:2304.09542* (2023).

[33] Zhongxiang Sun. 2023. A short survey of viewing large language models in legal aspect. *arXiv preprint arXiv:2303.09136* (2023).

[34] Teo Susnjak. 2023. Applying BERT and ChatGPT for Sentiment Analysis of Lyme Disease in Scientific Literature. *arXiv preprint arXiv:2302.06474* (2023).

[35] Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

[36] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. arXiv:2303.07678 [cs.IR]

[37] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems* 33 (2020), 5776–5788.

[38] Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023. Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study. *arXiv preprint arXiv:2304.04339* (2023).

[39] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

[40] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2013–2018.

[41] Bowen Zhang, Daijun Ding, and Liwen Jing. 2022. How would Stance Detection Techniques Evolve after the Launch of ChatGPT? *arXiv preprint arXiv:2212.14548* (2022).

[42] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Extractive Summarization via ChatGPT for Faithful Summary Generation. *arXiv preprint arXiv:2304.04193* (2023).

[43] Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems* 31 (2018).

[44] Shengyao Zhuang, Hang Li, and Guido Zuccon. 2021. Deep query likelihood model for information retrieval. In *European Conference on Information Retrieval*. Springer, 463–470.

[45] Shengyao Zhuang and Guido Zuccon. 2021. TILDE: Term independent likelihood moDEl for passage re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1483–1492.