A Test Collection of Synthetic Documents for Training Rankers

ChatGPT vs. Human Experts

Askari, A.; Aliannejadi, M.; Kanoulas, E.; Verberne, S.

DOI
10.1145/3583780.3615111

Publication date
2023

Document Version
Final published version

Published in
CIKM ’23

License
CC BY

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
ABSTRACT
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, the human ChatGPT comparison corpus (HC3), consisting of multiple public question collections featuring both human- and ChatGPT-generated responses. 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, however, 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 supplement training data, especially in domains with less labeled data. ChatGPT-RetrievalQA presents various opportunities for analyzing and improving rankers with both human- and LLM-generated data. Our data, code, and model checkpoints are publicly available.¹

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

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

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

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 [21], and stance detection [41]. Although ChatGPT can produce impressive responses, it is not immune to errors or hallucinations [13]. 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 less labeled data.

InPars [3], Promptagator [10], InPars-light [4], and InPars-v2 [16] have utilized LLMs to generate synthetic queries given documents. Particularly, InPars-v2 [16] 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 generated from 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.
Table 1: Statistic on the size of train, development, and test sets across domains for evaluation of cross-encoders.

<table>
<thead>
<tr>
<th>Domain</th>
<th># of queries</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train set</td>
</tr>
<tr>
<td>All</td>
<td>16788</td>
</tr>
<tr>
<td>Finance: FiQA [23]</td>
<td>2715</td>
</tr>
<tr>
<td>Reddit: EL15 [12]</td>
<td>11809</td>
</tr>
<tr>
<td>Wikipedia: openQA [40]</td>
<td>820</td>
</tr>
</tbody>
</table>

ChatGPT-RetrievalQA dataset is built based on the human ChatGPT comparison corpus (HC3) dataset [14]. HC3 is built by prompting ChatGPT with the questions of several public question-answering datasets and prompts. The goal of HC3 is to linguistically compare human and ChatGPT responses and explore the possibility of differentiating the responses generated by ChatGPT and those written by humans. HC3 contains questions (i.e., queries) from four different domains, namely, medicine (Medical Dialog [7]), finance (FiQA [23]), Wikipedia (WikiQA [40] and Wiki_csa [14]), and Reddit (EL15 [12]). 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. [11] who study if LLMs can be used for generating relevance labels, Murgia et al. [24] who compare ChatGPT’s responses with a search engine in a classroom setup, 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: (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 effective is ChatGPT-generated data on different 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 CE_{ChatGPT}² for inference (i.e., re-ranking task) on human-generated responses. Through our 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 ChatGPT-generated, 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 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 documents as training data in information retrieval tasks.

2 METHODOLOGY

Dataset and pool preparation. Our ChatGPT-RetrievalQA dataset is based on the HC3 dataset produced by Guo et al. [14], 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 [14]), we convert the dataset files to a format similar to MS MARCO [25], in both full-ranking and re-ranking setups.³ We divide the data into training, development, and test sets. To facilitate training, we provide training 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, development, and test sets for each domain.

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 words that overlap between them:

\[
s_{BM25}(q, d) = \frac{\sum_{t \in q \cap d} rs_j \cdot \frac{tf_{i,t}}{tf_{i,t} + k_1 (1-b+b \frac{ld}{wl})}}{\sum_{t \in q} tf_{i,t} + k_1 (1-b+b \frac{ld}{wl})}
\]

where \( t \) is a term, \( tf_{i,t} \) is the frequency of \( t \) in document \( d \), \( rs_j \) is the Robertson–Spärck Jones weight [29] of \( t \), and \( l \) is the average document length. \( k_1 \) and \( b \) are parameters.

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, like MonoBERT or CE_{CAT}. 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.

³This 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 a dataset among all of the cross-encoder re-rankers and BM25 on the corresponding 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>ChatGPT-RetrievalQA (Ours)</th>
<th>TREC DL’19</th>
<th>TREC DL’20</th>
<th>MS MARCO DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP</td>
<td>NDCG</td>
<td>MRR</td>
<td>MAP</td>
</tr>
<tr>
<td>MiniLM\textsubscript{human}</td>
<td>.310†</td>
<td>.384†</td>
<td>.460†</td>
<td>.326</td>
</tr>
<tr>
<td>MiniLM\textsubscript{ChatGPT}</td>
<td>.294</td>
<td>.362</td>
<td>.444</td>
<td>.342†</td>
</tr>
<tr>
<td>TinyBERT\textsubscript{human}</td>
<td>.244</td>
<td>.310</td>
<td>.367</td>
<td>.294</td>
</tr>
<tr>
<td>TinyBERT\textsubscript{ChatGPT}</td>
<td>.231</td>
<td>.291</td>
<td>.358</td>
<td>.328</td>
</tr>
</tbody>
</table>

by cross-encoder (CE) using a single linear layer $W_s$:

$$CE_{CAT} (q, m, P(t:x)) = CE ([CLS] q [SEP] p [SEP]) + W_s (2)$$

We use $CE_{CAT}$ as our cross-encoder re-ranker with a re-ranking depth of 1000. In our experiments, both $CE_{ChatGPT}$ 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 The MS MARCO-passage dataset contains about 8.8 million passages and about 1 million natural language queries and has been extensively used to train deep language models for ranking. Following prior work on MS MARCO [18, 20, 22, 43, 44], we only use the development set (~7k queries) for our empirical evaluation. We measure the same metrics in the in-domain setting on the test set of ChatGPT-RetrievalQA. In ChatGPT-RetrievalQA, the average length of human and ChatGPT responses are 142.5 and 198.1 words, respectively.

Training configuration. We use the Huggingface library [39], and PyTorch [27] for the cross-encoder re-ranking training and inference. Following prior work [15], we use the Adam [19] optimizer with a learning rate of $7 \times 10^{-6}$ for all cross-encoder layers, regardless of the number of layers trained.

4 RESULTS

4.1 Main results (RQ1)

Table 2 shows a comparison of the effectiveness of $CE_{human}$ and $CE_{ChatGPT}$. 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) [17] to assess the generalizability of our results. In the in-domain setting where we evaluate the test set queries with human-generated documents, MiniLM\textsubscript{human} significantly outperforms all other cross-encoder re-rankers. Although performing worse than the human-trained models, MiniLM\textsubscript{ChatGPT} and TinyBERT\textsubscript{ChatGPT} still outperform the strong baseline [3], BM25 [29], statistically significantly by a large margin in this setting. In the out-of-domain setting, the MiniLM\textsubscript{ChatGPT} consistently outperforms the other cross-encoder re-rankers including MiniLM\textsubscript{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, MiniLM\textsubscript{ChatGPT} for NDCG@10, and TinyBERT\textsubscript{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 MiniLM\textsubscript{human} and MiniLM\textsubscript{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 MiniLM\textsubscript{human} achieves higher effectiveness than MiniLM\textsubscript{ChatGPT} for all domains except Reddit. However, the difference in effectiveness is small, and MiniLM\textsubscript{ChatGPT} still achieves a reasonable level of effectiveness. In the finance domain, both MiniLM\textsubscript{human} and MiniLM\textsubscript{ChatGPT} achieve relatively low effectiveness compared to other domains. In the Wikipedia domain, MiniLM\textsubscript{human} and MiniLM\textsubscript{ChatGPT} achieve relatively similar levels of effectiveness. In the medicine domain, the $CE_{human}$ shows the highest effectiveness. Overall, these results suggest that MiniLM\textsubscript{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 MS MARCO [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 is worth exploring a different setup in the future.
where 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 observe the same pattern for the development 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.

**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 document 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 MiniLM\( _{ChatGPT} \) 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 humans, 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 are 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 ChatGPT-generated, 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 ones. Our results show that \( CE_{ChatGPT} \) is more effective than \( CE_{human} \) in the out-of-domain setting while MiniLM\( _{human} \) 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 ones, indicating that human-generated responses are more challenging to term-matching models.

Overall, our findings based on our dataset and experiments suggest that ChatGPT-generated responses are more useful than human-generated ones for training effective re-rankers in out-of-domain retrieval, and highlight the potential of using generative LLMs for generating effective and useful responses for creating training datasets in NLP and IR 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 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
