UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Query Generation Using Large Language Models

A Reproducibility Study of Unsupervised Passage Reranking

Rau, D.; Kamps, J.

DOI
10.1007/978-3-031-56066-8_19

Publication date
2024

Document Version
Final published version

Published in
Advances in Information Retrieval

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care)

Link to publication

Citation for published version (APA):
Rau, D., & Kamps, J. (2024). Query Generation Using Large Language Models: A Reproducibility Study of Unsupervised Passage Reranking. In N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, & I. Ounis (Eds.), *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024 : proceedings* (Vol. IV, pp. 226-239). (Lecture Notes in Computer Science; Vol. 14611). Springer. https://doi.org/10.1007/978-3-031-56066-8_19

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.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:03 Dec 2024

Query Generation Using Large Language Models
A Reproducibility Study of Unsupervised Passage Reranking

David Rau(✉) and Jaap Kamps

University of Amsterdam, Amsterdam, The Netherlands
{d.m.rau,kamps}@uva.nl

Abstract. Existing passage retrieval techniques predominantly emphasize classification or dense matching strategies. This is in contrast with classic language modeling approaches focusing on query or question generation. Recently, Sachan et al. introduced an Unsupervised Passage Retrieval (UPR) approach that resembles this by exploiting the inherent generative capabilities of large language models. In this replicability study, we revisit the concept of zero-shot question generation for re-ranking and focus our investigation on the ranking experiments, validating the UPR findings, particularly on the widely recognized BEIR benchmark. Furthermore, we extend the original work by evaluating the proposed method additionally on the TREC Deep Learning track benchmarks of 2019 and 2020. To enhance our understanding of the technique's performance, we introduce novel experiments exploring the influence of different prompts on retrieval outcomes. Our comprehensive analysis provides valuable insights into the robustness and applicability of zero-shot question generation as a re-ranking strategy in passage retrieval.

Keywords: LLMs based Query Generation · Neural Ranking Models · Language Modeling Framework for IR

1 Introduction

Text ranking stands at the beginning of several downstream NLP tasks, such as open-domain question answering (Open-QA). In this context, the objective is to retrieve relevant passages that may contain the answer. Text ranking is a rapidly evolving research area that has recently undergone drastic performance improvements by employing neural ranking models based on large-language models (LLMs). Typically, these models require extensive fine-tuning specifically tailored to the ranking task, frequently employing large training sets [2,3] on MS-Marco [1].

A useful way to classify ranking approaches is based on whether these are dense or sparse, and whether these are supervised or unsupervised [7]. While neural models dominate the supervised approaches, up to now unsupervised approaches tend to be traditional lexical systems, either traditional IR models

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
N. Goharian et al. (Eds.): ECIR 2024, LNCS 14611, pp. 226–239, 2024.
https://doi.org/10.1007/978-3-031-56066-8_19

based on text statistics, or classic dense IR models such as LSI. Given the earlier success of the statistical modeling framework in information retrieval [9], our main motivation in this paper is to explore how effective closely related unsupervised approaches based on LLMs are.

With the recent availability of various LLMs with billions of parameters like T0 [13], BLOOM [14], OPT [18], Llama [17], there lies an opportunity for further performance enhancement by utilizing these even bigger models for text retrieval. Nevertheless, fine-tuning becomes progressively expensive with increasing model size, prompting a surge of interest in applying these models in an unsupervised way. Sachan et al. [12] were the first to utilize the generative power of these LLMs proposing an unsupervised query generation method for re-ranking, coined as Unsupervised Passage Re-ranker (UPR). UPR utilizes the log-likelihood of the query conditioned on a passage for re-ranking without requiring specific training. This approach inspired follow-up research, exploring various techniques such as pairwise and listwise approaches [4,10,11,19].

In this study aimed at replicating Sachan et al. [12] work, we reexamine the UPR concept, with a specific focus on ranking experiments on the popular BeIR benchmark [16], and validate findings presented at EMNLP 2022. Our goal is to shed light on the practicality and reproducibility of this approach, further advancing our understanding of zero-shot question generation for re-ranking using LLMs, particularly through instruction prompts.

In this work, we formulate and address the following research questions:

– **RQ1**: *Are the results of UPR on BeIR reproducible?*

We partially reproduce (Different team, different experimental setup) the core re-ranking experiments of the original work. We find the approach to be computationally very expensive limiting our reproducibility results to only a subset of the datasets. Next, we reduce the number of passages to be re-ranked and repeat the first experiment.

– **RQ2**: *Does shallower re-ranking impact performance negatively?*

While investigated in the original paper on a single dataset we extend their experiment to the entire BeIR and come, in contrast to the original work, to the conclusion that re-ranking a smaller set can improve performance.

After, we investigate whether unifying both retrieval stages using a query likelihood model (QL) instead of BM25 as an initial retriever leads to a positive interaction between the two QL models.

– **RQ3**: *Does initial retrieval using QL improve re-ranking performance?*

Our results suggest the opposite, interactions between BM25 and UPR seem to be more beneficial for performance.

We then extend to work of Sachan et al. [12] by evaluating UPR two new test sets of TREC DL containing more fine-grained NIST judgments.

– **RQ4**: *Does UPR achieve similar performance gains over BM25 on TREC DL?*

Our results answer this positively. Although the original paper is focusing on question answering tasks, experiments on TREC DL confirm the viability of the model for ranking.

Finally, we investigate the robustness of the performance by experimenting with different instruction prompts for the LLM.

– **RQ5**: *What is the impact of the prompt on the re-ranking performance?*

We find the model overall to be robust in many cases to the prompts, however, also find that small changes can have a negative impact on performance.

Main Contributions. Our main contributions are: (i) We conduct a replicability study of zero-shot question generation (UPR) focusing on re-ranking. (ii) We extend an ablation study of the impact of the passage candidate size to the entire BeIR (iii) We complement the original work by evaluating UPR on two new testsets, TREC Deep Learning 2019 and 2020, which present the golden standard to evaluate re-ranking with fine-grained relevance judgments. (iii) We investigate the impact of different prompts on the retrieval outcomes. And finally, (iv) we provide our codebase to facilitate the reproducibility of all results presented.

The code to reproduce the results of this paper can be found under: https://github.com/davidmrau/upr_reproducibility_ecir24.

This paper is structured as follows: First, in §2 we give an overview of the reproduced method UPR. §3 we detail our experimental setup. Next, in §4 we present our experiments and results. Finally, in §5 we conclude this reproducibility study.

2 Method

In this section, we detail the approach proposed by Sachan et al. [12]. The approach follows the classical re-ranking pipeline. First, a retriever retrieves a set of passages, which is then re-ranked by a more complex model. We will detail the pipeline in the following subsections, following the original notation.

2.1 Retriever

The retriever provides an initial set of candidate passages that can later be re-ranked. For this, the retriever ranks all passages $\mathcal{D} = \{d_1, \cdots, d_M\}$ according to the given question q. This yields a sub-set of relevant passages $\mathcal{Z} \subset \mathcal{D}$ containing possible answers to the question q. The retriever provides the most K relevant passages $\mathcal{Z} = \{z_1, \cdots, z_K\}$.

2.2 Unsupervised Passage Re-ranking (UPR)

It is the task of the re-ranker to re-score the set $\mathcal{Z}$ candidate passages ranking the most relevant passages at the top. While retrievers are required to be efficient in potentially reranking millions of passages the focus for re-rankers is to provide more fine-grained ranking scores. For each passage $z_i \in \mathcal{Z}$ a relevance score is computed as $p(z_i|q)$. The approach proposed by [12] introduced a straightforward yet insightful ranking approach that leverages the inherent generative capabilities of large language models. They utilize a pre-trained large language model (LLM) to estimate the conditional probability of generating the query q given the text passage z. In this specific case, the LLM is only trained on the next token prediction task learning to model language. The model is therefore not trained for the specific task of query generation and is therefore *unsupervised*. In more detail, applying the Bayes' rule to the conditional probability $p(z_i|q)$ yields:

$$\log p(z_i|q) = \log p\left(q \mid z_i\right) + \log p\left(z_i\right) + c, \tag{1}$$

where $p(z_i)$ denotes the prior of the passage and a common constant for all z_i. The authors make the simplifying assumption of a uniform assumption for all passages z_i. Therefore we obtain:

$$\log p(z_i|q) \propto \log p\left(q \mid z_i\right), \forall z_i \in \mathcal{Z} \tag{2}$$

The LLM parameterized by Θ is then used to calculate the average log-likelihood over query terms. as follows:

$$\log p(q|z_i) = \frac{1}{|q|} \sum_t \log p\left(q_t \mid q_{<t}, z_i; \Theta\right) \tag{3}$$

To make sure the generated query stays close to the actual query teacher forcing is applied through the entire input. For re-ranking the negative log-likelihood is then used as a relevance score to rank passages. An example of the model input is given in the following:

"*Passage:* {}. *Please write a question based on this passage. Question:* {}",

where {} is replaced with the content of the passage/query respectively.

3 Experimental Setup

In this section, we detail our experimental setup consisting of datasets, retrievers, and the pre-trained LLM.

3.1 Datasets

In this section we detail the used datasets. The original work evaluates UPR on the BeIR Benchmark. We extend the evaluation to two TREC DL testsets.

BeIR Benchmark. BeIR Benchmark is a popular test suite for zero-shot evaluation it consists of various datasets of which only a subset is publicly available. Each dataset contains relevance judgments, queries, and evidence passages. The datasets span different retrieval tasks such as question answering, fact-checking, etc., and stem from different domains. We rely on the publicly available datasets provided on the HuggingFace Hub. For two datasets we find different numbers of queries than reported in the original. For CQA-Dupstack 12,569 instead 13,145 and for Nfcorpus 308 instead 323.

TREC Deep Learning Testsets. The TREC Deep Learning Track testsets are the golden standard for evaluating re-ranking models. In contrast to the publicly available BeIR datasets TREC DL testsets contain fine-grained relevance judgements provided by NIST accessors and are of high quality. We use the passage retrieval task testsets of TREC DL 2019/2020 [2,3]. The queries are based on real user search queries from the Bing search engine.

3.2 Retriever

BM25. BM25 is a lexical-based retriever that is based on TF-IDF. We follow the original work and use the publicly available Pyserini Python toolkit [8] for initial retrieval with BM25 using the default parameters.

Query Likelihood (QL). The query likelihood retrieval model is a probabilistic retriever that assesses the relevance of passage to a user's query based on the likelihood that the passages would generate the query. It calculates the probability of observing the query terms in a passage and ranks passages by their estimated likelihood of being relevant to the query. Again, we use the publicly available Pyserini Python toolkit [8] for initial retrieval with QL with the default parameters.

3.3 Pre-trained Large-Language Model

T0_3B. Following the original work, we employ a T5-based model with 3 billion parameters. T5 employs an encoder-decoder architecture, where the encoder processes input text and the decoder generates output text, allowing it to excel in tasks such as text generation, translation, and summarization. We utilize the publicly available model available on HuggingFace hub[1] under "bigscience/T0_3B". To accelerate inference we use the model in half-precision.

4 Experiments and Results

In this section, we answer the research questions posed in Sect. 1. We organized the experiments in subsections of which each will answer one research question.

[1] https://huggingface.co/.

First, in §4.1 we focus on the reproducibility of the re-ranking results [12] using UPR on the BeIR Benchmark. Second, in §4.2, we repeat an ablation study changing the experimental setup by (i) UPR re-ranking a smaller number of passages, and (ii) evaluating the ablation instead of on only one dataset in the original paper on the entire BeIR Benchmark. Then, in §4.3 we investigate the impact of the performance when using a query likelihood model for the initial retrieval as well. In §4.4 we test UPR on two new testsets, namely TREC DL 2019 and 2020. Finally, in §4.5 we test the robustness of the model to the instruction prompt.

4.1 Reproducibility: Unsupervised Passage Re-ranking

We are interested in whether the main re-ranking results on BeIR in the original work can be reproduced. In this section, we aim to reproduce the passage re-ranking results on the BeIR Benchmark. **RQ1**: *Are the results of UPR on BeIR reproducible?*

To address RQ1, we reproduce the best-performing method presented in Table 6 of the original paper [12], namely re-ranking the top-1,000 passages retrieved by BM25 with *UPR* leveraging the T0 3B Language model. A more detailed report on the BeIR Benchmark can be found in Appendix A.4 in the original paper, which will serve as a reference for this reproducibility study. Following the original paper we report NDCG@10 and Recall@100 averaged over queries.

While the authors shared their code[2], running the BeIR benchmark is not supported. Previous reproducibility studies have [5,6,15] shown how different pre-processing can have grave impacts on the reported performance, therefore we believe in the importance of this reproducibility study to confirm the results independently. To this end, we independently re-implement UPR from scratch[3], including the support of running the BeIR Benchmark out-of-the-box. We further add the support of CQADupStack to the HuggingFace hub, which is currently not available. While the retriever scores are replicated (different team, same experimental setup) the UPR scores are reproduced (different team, different experimental setup).

Contrary to the original paper we report conistently non-capped recall scores across datasets (the original paper reports capped recall scores for trec-covid). The original work does not specify a max. input length for UPR or whether special tokens should be added. We choose a max. length of 512 and add special tokens.

Results. Due to the computational complexity of re-ranking the top-1,000 results with such a large LLM, we could only reproduce results for a limited subset of the datasets in BeIR. To put this into context, just for evaluating MS-Marco re-ranking $\approx$ 7M passages (6,980 queries x 1,000 passages) with max.

[2] https://github.com/DevSinghSachan/unsupervised-passage-reranking.
[3] https://github.com/davidmrau/upr_reproducibility_ecir24.

Table 1. Reproduced Unsupervised passage re-ranking results on the BEIR benchmark. #Q and # denote the number of queries and evidence passages, respectively. We show results for BM25 top 1,000 and re-ranking the same with the T0-3B language model using UPR. Δ indicates the performance difference relative to the original results [12]. Results − could not be reproduced due to the computational complexity. The average is only over the reported datasets, excluding the non-capped recall values.

Dataset	#Q	#E	NDCG@10		Recall@100	
			BM25 (Δ)	re-ranked (Δ)	BM25 (Δ)	re-ranked (Δ)
Scifact	300	5K	66.5	70.6 (+0.3)	90.8	94.8 (+0.6)
Scidocs	1000	25K	15.8	-	35.6	-
Nfcorpus	308	3.5K	34.1 (+1.6)	36.3 (+1.5)	26.2 (+1.2)	27.4 (−0.6)
FIQA-2018	648	57K	23.6	-	53.9	-
Trec-covid	50	0.2M	65.6 (+0.1)	68.0 (−0.8)	11.4	12.7 (*)
Touche-2020	49	0.4M	36.7 (−0.1)	22.3 (+1.7)	53.8	46.6 (+0.9)
NQ	3452	2.7M	32.9	-	76.0	-
MS-MARCO	6980	8.8M	22.8	-	65.8	-
HotpotQA	7405	5.2M	60.3	-	74.0	-
ArguAna	1406	8.7K	41.4 (+9.9)	-	94.3	-
CQADupStack	12569	0.5M	29.9	-	60.6	-
Quora	10000	0.5M	78.9	-	97.3	-
DBPedia	400	4.6M	31.3	36.3 (+0.9)	39.8	48.4 (−4.9)
Fever	6666	5.4M	75.3	-	93.0	-
Climate-Fever	1535	5.4M	21.3	-	43.6	-
Average			42.4 (+0.8)	46.7 (+0.7)	64.6 (+0.1)	54.3 (−1.0)

* non-capped

batch size 196 on a single A100 40 GB would take 130 h. Since the number of queries is the defining parameter for this experiment we choose all datasets with < 500 queries-namely Scifact, NFcorpus, Trec-covid, touche-2020, and DBPedia. In the next section, we will provide results for all datasets in BeIR using the top-100 passages instead.

Our results reproducing the results of the original work can be found in Table 1. The table shows NDCG@10 and Recall@100 for the initial retrieval of top-1,000 passages using BM25. Δ indicates the difference from the originally reported results.

We first investigate the performance of our BM25 retriever. Our replicated scores match the scores for most datasets reported in the original paper macro average NDCG@10 42.4 vs 43.2 and Recall@100 64.6 vs 64.7. Note that the macro average of the re-ranking presented in Table 1 are not directly comparable with the original work, as we average only over the scores present in the table (excluding non-capped recall). Only for datasets Nfcorpus, Trec-covid, Touche-2020, and ArguAna, we observe minimal score variations except for

ArguAna which deviates strongly (+20% NDCG@10). The original work reports NDCG@10 31.5 while we observe a score of 41.4. Since the experimental setup is the same it is not clear where the score differences originate (our scores match the official reported scores using Pyserini[4]).

Considering UPR, the main interest in this reproducibility experiment, we observe small score differences. The original paper reports a macro average for NDCG@10 46.7 vs 47.4 and Recall@100 54.3 vs 53.4. These performance differences are likely rooted in the score deviations for BM25.

Answer to RQ1. Our first research question was: *Are the results of UPR on BeIR reproducible?* Due to the computational complexity of the approach, only a subset of the results could be reproduced. While observing small score differences for BM25 and UPR, for the experiments that we could reproduce, we were overall able to validate the claims presented in the original work regarding the passage re-ranking of UPR on the BeIR benchmark.

4.2 Shallow Re-ranking with UPR

The computational complexity of re-ranking in the previous experiment allowed us to only reproduce results for a subset of BeIR. In this section, we seek to answer **RQ2**: *Does shallower re-ranking impact performance negatively?* We repeat the experiment of the previous section but reduce the number of passages to be re-ranked from the top-1000 to top-100. This not only enables us to assess UPR across all the originally reported datasets using our limited computational resources but also facilitates an analysis of the influence of re-ranking a smaller subset of the initially retrieved passages with UPR, significantly reducing GPU runtime. The original paper explores a similar experiment in Sect. 4.2.3, only on the NQ dataset. We complement the original paper by validating their result on the entire BeIR Benchmark using NDCG@10 as a metric, as the initial results are of most interest. We omit reporting Recall@100 as it remains constant re-ranking the top-100.

Results. We present our results in Table 2. The UPR NDCG@10 scores for re-ranking are directly comparable to Table 1 and Table 10 in the original work. We observe that re-ranking only the top-100 passages yields a performance of NDCG@10 46.7 on average over all datasets compared to 44.0 in the original work, being 2,7 points higher. These results suggest that re-ranking a smaller number of passages using UPR can improve performance, indicating that UPR is overestimating the relevance of some passages that are ranked lower than top-100.

Answer to RQ2. Our second research question was: *Does shallower re-ranking impact performance negatively?* We find that re-ranking the top-100 passages

[4] https://castorini.github.io/pyserini/2cr/beir.html.

Table 2. Reproduced Unsupervised passage re-ranking results on the BEIR benchmark. #Q and # denote the number of queries and evidence passages, respectively. We show results for BM25 top 100 and re-ranking the same with the T0-3B language model using UPR. Δ indicates the performance difference relative to the original results [12].

Dataset	#Q	#E	NDCG@10	
			BM25 (Δ)	UPR (re-ranked) (Δ)
Scifact	300	5K	66.5	70.5 (+ 0.2)
Scidocs	1000	25K	15.8	17.6 (+ 0.6)
Nfcorpus	308	3.5K	34.1 (+1.6)	36.4 (+ 1.6)
FIQA-2018	648	57K	23.6	41.5 (− 2.9)
Trec-covid	50	0.2M	65.6 (+0.1)	76.0 (+ 7.2)
Touche-2020	49	0.4M	36.7 (−0.1)	22.4 (+ 1.8)
NQ	3452	2.7M	32.9	45.6 (+ 0, 2)
MS-MARCO	6980	8.8M	22.8	29.4 (− 0.8)
HotpotQA	7405	5.2M	60.3	70.7 (− 2.6)
ArguAna	1406	8.7K	41.4 (+9.9)	50.0 (+12.8)
CQADupStack	12569	0.5M	29.9	37.8 (− 3.8)
Quora	10000	0.5M	78.9	83.5 (+ 0.4)
DBPedia	400	4.6M	31.3	35.5 (− 0.1)
Fever	6666	5.4M	75.3	67.1 (+ 0.8)
Climate-Fever	1535	5.4M	21.3	16.0 (+ 4.3)
Average			42.4 (+0.8)	46.7 (+ 1.8)

using UPR does not have a negative impact on performance. In contrast, we observe the performance to increase for Trec-covid over re-ranking a very deep pool (top-1000) of candidate passages. Reproducing the ablation study of the original work on the entire BeIR (instead of only NQ) suggests re-ranking a shallower pool of candidate passages to be sufficient if not beneficial (for getting the top results right), which is the opposite of what the original work suggests.

4.3 UPR with Query Likelihood Retriever

The original paper utilizes both sparse retrievers (BM25) and dense retrievers (Contriever) for the initial retrieval process. In both cases, this creates a discrepancy between the first-stage and the second-stage rankers. This leaves unifying first- and second-stage retrieval by using query likelihood models for both stages unexplored. We fill this gap by repeating the experiment but using a Query Likelihood model as a retriever.

In this section, we seek to answer **RQ3**: *Does initial retrieval using QL improve re-ranking performance?* Employing query likelihood-based rankers for both stages might lead to positive interaction effects between them, as similar

Table 3. Unsupervised passage re-ranking results on the BEIR benchmark for the Query Likelihood retriever of top 100 and re-ranking the same with the T0-3B language model using UPR. #Q and #E denote the number of queries and evidence passages, respectively.

Dataset	#Q	#E	NDCG@10	
			QLD	UPR (re-ranked)
Scifact	300	5K	66.5	70.9
Scidocs	1000	25K	14.9	17.1
Nfcorpus	323	3.5K	33.5	36.6
FIQA-2018	648	57K	20.5	39.7
Trec-covid	50	0.2M	54.0	75.3
Touche-2020	49	0.4M	49.8	25.2
NQ	3452	2.7M	29.3	44.7
MS-MARCO	6980	8.8M	20.8	29.4
HotpotQA	7405	5.2M	58.4	67.0
ArguAna	1406	8.7K	36.1	49.1
CQADupStack	13145	0.5M	24.0	36.1
Quora	10000	0.5M	64.8	81.7
DBPedia	400	4.6M	27.6	34.3
Fever	6666	5.4M	71.3	66.8
Climate-Fever	1535	5.4M	20.7	16.2
Average			40.1	44.7

Table 4. Unsupervised passage re-ranking results for TREC DL 2019 and 2020 for BM25 and re-ranked with the T0-3B language model using UPR.

Dataset	#Q	#E	NDCG@10		MRR@10	
			BM25	UPR (re-ranked)	BM25	UPR (re-ranked)
TREC DL 2019	43	8.8M	50.6	61.1	70.2	70.9
TREC DL 2020	54	8.8M	48.0	61.7	65.3	74.4

passages might be promoted by both models. Again, limited by the computational complexity re-rank only the top-100 passages.

Results. We report the performance in Table 3 in NDCG@10. We find both the initial ranking as well as the re-ranking using a query likelihood-based model to perform worse than when using BM25 as an initial retriever (comparing results with Table 2).

Answer to RQ3. Our third research question was: *Does initial retrieval using QL improve re-ranking performance?* Our results suggest that unifying first- and

second-stage retrieval using a query likelihood-based model does not lead to an improved re-ranking performance.

4.4 UPR on TREC DL

In this section we investigate **RQ4**: *Does UPR achieve similar performance gains over BM25 on TREC DL?* While the original paper explores various different datasets mostly focussing on passage retrieval for QA, the main testsets for passage re-ranking TREC DL are not reported, and is unclear whether the performance gains translate to the fine-grained high-quality relevance labels of the NIST accessors. We first retrieve the top-100 passages using BM25 and then re-rank them using UPR. We show the most popular metrics for TREC DL NDCG@10 and MRR@10.

Results. The results for TREC DL 2019/2020 can be found in Table 4. We observe UPR comfortably outperforming BM25 for both measures, except for TREC DL 2019 MRR@10 is only marginally higher for the re-ranked passages.

Answer to RQ4. Our fourth research question was: *Does UPR achieve similar performance gains over BM25 on TREC DL?* Re-ranking using UPR is effective and able to outperform BM25 by a large margin, similar to the claims of the original work. Our results on two new datasets validate the claims of the original work that re-ranking using UPR can improve over traditional unsupervised models (over 22% on TREC DL 2019, and 18% on 2020), even though with a smaller margin 10% on BeIR.

4.5 Robustness of UPR to Instruction Prompts

While the original work briefly mentions instruction prompt selection in A.2. they do not provide ablations to different prompts or indicate the robustness of the performance to different prompts. We investigate this with **RQ5**: *What is the impact of the prompt on the re-ranking performance?*

For this experiment, we systematically evaluate the robustness UPR using a variety of different prompts changing the instruction component's order, removing instruction terms, lowercasing, and paraphrasing. Similar to the previous experiment we retrieve top-100 candidates using BM25 and re-rank with UPR. We measure the performance on the TREC DL 2020 testset in NDCG@10.

Results. The prompts alongside the performance can be found in Table 5. We observe that overall the model is robust against changes to the instruction prompts. However, we observe unexpected performance drops for exchanging the term "question" for "query" (see Table 5 *paraphrasing* 7.) as well as removing the instructions entirely (see *removing instruction terms* 3.). Shortening the instructions to a minimum (*removing instruction terms* 5.) seems to be an effective way of prompting for UPR.

Table 5. Robustness of UPR to changes in the instruction prompts reported on TREC DL 2020.

Prompt	NDCG@10
default:	
1. *Passage:* {}. *Please write a question based on this passage. Question:* {}	61.74
changing instruction component order:	
1. *Please write a question based on the passage. Passage:* {}. *Question:* {}	61.10
paraphrasing:	
1 *Passage:* {}. *Please write a question based on the previous input. Question:* {}	62.03
2 *Text:* {}. *Please write a question based on this text. Question:* {}	61.78
3 *Passage:* {}. *Please write a question that this passage could answer. Question:* {}	60.27
4 *We are using zero-shot question generation to re-rank passages of the generated question. For this based on the likelihood please write a question based on this passage. Passage:* {}. *Question:* {}	60.98
5 *Passage:* {}. *Please generate a question based on this passage. 4. Question:* {}	61.69
6 *Passage:* {}. *Please output a question based on this passage. Question:* {}	60.59
7. *Passage:* {}. *Please write a query based on this passage. Query:* {}	54.57
removing instruction terms:	
1 *Passage:* {}. ~~*Please*~~ *Write a question based on this passage. Question:* {}	60.55
2 ~~*Passage:*~~ {}. ~~*Please write a question based on this passage.*~~ *Question:* {}	53.02
3 *Passage:* {}. ~~*Please write a question based on the passage.*~~ *Question:* {}	56.97
4 ~~*Passage:*~~ {}. *Please write a question based on this passage. Question:* {}	61.49
5. *Passage:* {}. ~~*Please*~~*Write* ~~*a*~~ *question* ~~*based on this passage.*~~ *Question:* {}	61.74
lowercasing:	
1 *passage:* {}. *please write a question based on this passage. question:* {}	60.88

Answer to RQ5. Our fifth research question was: *What is the impact of the prompt on the re-ranking performance?* While the model overall seems to be fairly robust against the proposed instruction prompt manipulations we find very subtle changes in the prompt such as exchanging single instruction terms with synonyms can lead to a large performance drop.

5 Discussion and Conclusions

In this reproducibility study, we have examined Unsupervised Passage Reranking using an LLM. We investigated the *reproducibility* of the zero-shot question generation results reported on the BeIR. To this end, we re-implemented UPR, and this way allow others to reproduce and experiment with UPR on BeIR, as well as TREC DL 2019/2020. For the datasets of BeIR that could be we evaluated, we found the original results to be reproducible, even though with small score differences, likely caused by different retrieval scores by BM25. We complemented the original work with several ablation experiments. First, we examined the impact of the size of the passage candidate set, and came, in contrast to the original work, to the conclusion, that re-ranking a smaller set of initially retrieved passages can improve performance using UPR while saving

GPU inference time. Second, we investigated the impact of unifying first- and second-stage retrieval using a query likelihood-based model for both and found it to compromise performance. Third, we test UPR also on two IR datasets (TREC DL 2019/2020) validating the effectiveness claims made by the original authors. Finally, we systematically test UPR for robustness to changes in the instruction prompt. We observe the performance of UPR to be very robust to most changes in the instruction prompt, however, also discovered how minimal changes can lead to a large drop in performance. Further, we found a minimal instruction of "Write question" to be sufficient.

Overall, our findings demonstrate the viability completely unsupervised neural information retrieval models. Any modern LLM can be used simply as a "language model" and estimate query likelihood in similar ways to the classic language modeling framework [9]. This opens up a new line of research in unsupervised neural IR models that complements the dominant focus on supervised neural IR models [7]. There are several benefits of pursuing such models. First, these models are applicable to very large language models where training is prohibitively expensive or impossible. Second, as unsupervised neural models clearly outperform traditional lexical approaches, they also present more realistic baselines for supervised ranking models. Third, there is conceptual and theoretical interest in unifying the statistical language modeling framework with current LLMs. Fourth, the controllable way of the unsupervised query generation model is by design avoiding hallucination, one of the main open problems of using LLMs for IR.

Acknowledgments. This research is funded in part by the Netherlands Organization for Scientific Research (NWO CI # CISC.CC.016).Views expressed in this paper are not necessarily shared or endorsed by those funding the research.

References

1. Bajaj, P., et al.: MS MARCO: a human generated machine reading comprehension dataset. CoRR (2016)
2. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2020 deep learning track. CoRR abs/2102.07662 (2021). https://arxiv.org/abs/2102.07662
3. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2019 deep learning track. CoRR abs/2003.07820 (2020). https://arxiv.org/abs/2003.07820
4. Dai, Z., et al.: Promptagator: few-shot dense retrieval from 8 examples. arXiv preprint arXiv:2209.11755 (2022)
5. Hendriksen, M., Vakulenko, S., Kuiper, E., de Rijke, M.: Scene-centric vs. object-centric image-text cross-modal retrieval: A reproducibility study. In: Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part III, p. 68–85, Springer-Verlag, Berlin, Heidelberg (2023), ISBN 978-3-031-28240-9, https://doi.org/10.1007/978-3-031-28241-6_5
6. Lajewska, W., Balog, K.: From baseline to top performer: A reproducibility study of approaches at the trec 2021 conversational assistance track. In: Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023,

Dublin, Ireland, April 2–6, 2023, Proceedings, Part III, p. 177–191, Springer-Verlag, Berlin, Heidelberg (2023), ISBN 978-3-031-28240-9, https://doi.org/10.1007/978-3-031-28241-6_12
7. Lin, J., Ma, X.: A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. CoRR abs/2106.14807 (2021). https://arxiv.org/abs/2106.14807
8. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: a Python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), pp. 2356–2362 (2021)
9. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Croft, W.B., Moffat, A., van Rijsbergen, C.J., Wilkinson, R., Zobel, J. (eds.) SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24–28 1998, Melbourne, Australia, pp. 275–281, ACM (1998), https://doi.org/10.1145/290941.291008
10. Pradeep, R., Sharifymoghaddam, S., Lin, J.: Rankvicuna: Zero-shot listwise document reranking with open-source large language models. arXiv preprint arXiv:2309.15088 (2023)
11. Qin, Z., et al.: Large language models are effective text rankers with pairwise ranking prompting. arXiv preprint arXiv:2306.17563 (2023)
12. Sachan, D., et al..: Improving passage retrieval with zero-shot question generation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 3781–3797, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022), https://doi.org/10.18653/v1/2022.emnlp-main.249, https://aclanthology.org/2022.emnlp-main.249
13. Sanh, V., et al.: Multitask prompted training enables zero-shot task generalization. arXiv preprint arXiv:2110.08207 (2021)
14. Scao, T.L., et al.: Bloom: a 176b-parameter open-access multilingual language model (2023)
15. Schütz, M.: Disinformation detection: Knowledge infusion with transfer learning and visualizations. In: Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part III, p. 468–475, Springer-Verlag, Berlin, Heidelberg (2023), ISBN 978-3-031-28240-9, https://doi.org/10.1007/978-3-031-28241-6_54
16. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: a heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021). https://openreview.net/forum?id=wCu6T5xFjeJ
17. Touvron, H., et al.: Llama: Open and efficient foundation language models (2023)
18. Zhang, S., et al.: Opt: open pre-trained transformer language models (2022)
19. Zhuang, H., et al.: Rankt5: fine-tuning t5 for text ranking with ranking losses. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2308–2313 (2023)