Retrievability bias estimation using synthetically generated queries

Abolghasemi, M.A.; Verberne, S.; Askari, A.; Azzopardi, L.

Retrievability Bias Estimation Using Synthetically Generated Queries

Amin Abolghasemi
m.a.abolghasemi@liacs.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Suzan Verberne
s.verberne@liacs.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Arian Askari
a.askari@liacs.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Leif Azzopardi
leif.azzopardi@strath.ac.uk
University of Strathclyde
Glasgow, UK

ABSTRACT

Ranking with pre-trained language models (PLMs) has shown to be highly effective for various Information Retrieval tasks. However, there is no prior work on evaluating PLM-based rankers in terms of their retrievability bias. In this work, we compare the retrievability bias in two of the most common PLM-based rankers, a Bi-Encoder BERT ranker and a Cross-Encoder BERT re-ranker against BM25, which was found to be one of the least biased models in prior work. Furthermore, we conduct a series of experiments with which we explore the plausibility of using synthetic queries generated with a generative model, docT5query, in the evaluation of retrievability bias. Our experiments show promising results on the use of synthetically generated queries for the purpose of retrievability bias estimation. Moreover, we find that the estimated bias values resulting from synthetically generated queries are lower than the ones estimated with user-generated queries on the MS MARCO evaluation benchmark. This indicates that synthetically generated queries might cause less bias than user-generated queries and therefore, by using such queries in training PLM-based rankers, we might be able to reduce the retrievability bias in these models.

CCS CONCEPTS

• **Information systems** → **Evaluation of retrieval results**.

KEYWORDS

Bias, Retrievability, Evaluation, Query Generation

ACM Reference Format:
Amin Abolghasemi, Suzan Verberne, Arian Askari, and Leif Azzopardi. 2023. Retrievability Bias Estimation Using Synthetically Generated Queries . In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615221

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

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615221

1 INTRODUCTION

Retrievability measures how likely a document is to be retrieved and exposed to the user in rankings [3]. A more biased retrieval system will overly favor certain documents over other ones. Previous work has studied the bias of traditional, lexical retrieval models including BM25, and traditional probabilistic language models [3, 26, 29, 30]. In various retrieval tasks, pre-trained language models (PLMs) have been shown to be more effective rankers than lexical models [17, 20]. However, the retrievability bias that these rankers exhibit over the collection of documents, has not yet been investigated. Moreover, BM25 as one of the most widely-used baselines in the study of PLM-based ranking models, was previously shown to be one of the models with the least bias. While BM25 has been compared to the latest PLM-based ranking models in terms of efficiency and effectiveness [1, 15, 17], to the best of our knowledge there is no prior work on comparing these models in terms of their bias. In this work, we analyze the retrievability bias in PLM-based ranking models using both real user queries and synthetically generated queries. To this aim we address the following research questions:

RQ1. What is the retrievability bias of BM25 (as one of the least biased traditional lexical ranking models) in comparison to that of PLM-based rankers?

To address this question, we investigate the retrievability bias in two of the most widely-used PLM-based rankers, a Bi-Encoder BERT ranker, and a Cross-Encoder BERT re-ranker. To evaluate the retrievability bias, we initially use human-generated queries in the MS MARCO Passage Ranking benchmark. Large scale relevance assessments for these queries enable the possibility to evaluate the effectiveness on the same query set that is used for retrievability bias estimation. However, large scale human-generated queries are not always available in all retrieval settings and domains. Therefore, as generative models have shown promising results in generating queries from a given document [11, 22], we explore if we can use these models to generate synthetic queries for estimating the retrievability bias. Thus, we address the following research question:

RQ2. Can we use synthetic query generation to create query sets for the estimation of retrievability bias?

Our experiments with synthetically generated queries indicate the promise of using these queries in the estimation of retrievability bias.

In summary, our contributions in this work are three-fold:

- We show that in addition to being more effective, the two widely-used PLM-based models (BERT Bi-Encoder ranker and BERT Cross-Encoder Re-ranker) are less biased than BM25 as a common lexical ranker baseline.
- We show the promise of using synthetic queries generated with transformer-based generative models in estimating retrievability bias.
- We find that synthetically generated queries might cause less bias and suggest that training with these queries can be considered as a future research direction for reducing retrievability bias in ranking models.

2 BACKGROUND

2.1 Retrievability and retrievability bias

Retrievability [3] is broadly defined as the amount of attention a document receives from the ranking model (also called exposure [7]). Put another way, retrievability is proportional to how often and how highly ranked a document d is returned over a sample of queries Q. More formally, given a collection of documents and a retrieval system, the retrievability score of document d can be defined as [3]:

$$\mathrm{R(d)}\ @k \propto \sum_{q \in Q} p(q).\,\mathrm{f}(q, d, k) \tag{1}$$

Here, $p(q)$ is the probability of query q being issued by user (which is mostly considered equal to 1 [3]), and $\mathrm{f}(q, d, k)$ is the utility function which determines the amount of attention user pays into document d in a ranked list of k documents retrieved for query q. Previous studies [3, 29] explore two kinds of utility function: (i) Cumulative utility and (ii) Gravity-based utility. In cumulative utility, f does not discriminate between the documents at different ranks, and in gravity-based utility, the retrievability score of documents is proportional to the expected attention that user pays to a document at a specific rank in the ranked list. These two functions can be formulated as:

$$\mathrm{R}(d)@k \propto \sum_{q \in Q} \frac{\mathbb{1}\,[r_{dq} \leq k]}{(r_{dq})^b} \tag{2}$$

where r_{dq} is the rank at which d is retrieved for a given query q, b represents the discount factor, and k is the cut-off [29]. $\mathbb{1}\,[r_{dq} \leq k]$ is equal to 1 if $r_{dq} \leq k$, otherwise, 0. Intuitively, k represents how far users are willing to go down the ranked list, while b specifies how much attention they pay to document at r_{dq}. Setting $b = 0$ forms Eq. 2 as a cumulative utility function and $b \neq 0$ results in a gravity-based utility function. Given the retrievability score R(d) $@k$ for each document, following prior work [3, 6, 27, 29] we assume that all documents should be treated equally, and thus have a similar level of retrievability (i.e. equality of retrievability). To this end, we use the Gini Coefficient (G) to evaluate the retrievability bias of models [10], as was used in prior work [3, 5, 6, 31]:

$$\mathrm{G} = \frac{\sum_{i=1}^{N} (2 * i - N - 1) \cdot \mathrm{R}(d_i)@k}{N \sum_{j=1}^{N} \mathrm{R}(d_j)@k} \tag{3}$$

Here N stands for the number of documents. A Gini of 0 indicates that all documents have the same retrievability (total equality), while a Gini of 1 indicates that one document has all the retrievability and all the other documents have a retrievability of zero (total inequality). Thus, a lower Gini coefficient corresponds to a less biased model [27].

2.2 Ranking Models

There are various PLM-based ranking paradigms. We use two of the most widely-used ones, namely a Bi-Encoder BERT ranker, as the representative of dense retrieval models, and a Cross-Encoder BERT re-ranker, as the model which has shown to provide the highest effectiveness in many information retrieval tasks [17]. Additionally, we will use BM25 as the most frequently-used traditional lexical ranker and the model which was shown to be one of the least biased models in prior work on retrievability [19].

Bi-Encoder BERT ranker. In ranking with the Bi-Encoder BERT ranker, following prior work [9], we use the final hidden state of the BERT $[CLS]$ token to encode the query q and the document d into their vector representations r_q, and r_d:

$$r_q = BERT(\ [CLS]\ q\ [SEP]\ )_{[CLS]} \tag{4}$$

$$r_d = BERT(\ [CLS]\ d\ [SEP]\ )_{[CLS]} \tag{5}$$

The similarity between the representation vectors r_q and r_d is then utilized to measure the relevance score between the query q and document d:

$$\mathrm{s}(q,\ d) = \mathrm{similarity}(r_q,\ r_d) \tag{6}$$

Cross-Encoder BERT ranker. Initially proposed by Nogueira and Cho [20], a Cross-Encoder BERT ranker [20] (also called MonoBERT) is a BERT model with a fully-connected layer W_p on top of its $[CLS]$ token final hidden states [20]. In this ranker, the concatenation of a query q and a candidate document d is used as the input and the output of the model s is leveraged as the relevance score of the document d for the input query q:

$$\mathrm{s}(q,\ d) = BERT\ (\,[CLS]\ q\ [SEP]\ d\ [SEP]\,)_{[CLS]} * W_p \tag{7}$$

BM25. BM25 follows a term-based exact matching paradigm in retrieving the relevant documents for a given query [24]. We use the implementation by Pyserini[1] [16] with the tuned parameters $k_1 = 0.82$, and $b = 0.68$.

3 EXPERIMENTAL METHODOLOGY

In this work, we employ the same methodology as used in prior studies [3, 25, 27] for estimating retrievability. However, we further extend the existing approach by drawing upon the ability of generative models to produce a synthetic set of queries to further provide a novel point of comparison with prior work.

3.1 Query Set Construction: OTQ

To create the query set Q in Eq. 2, prior work has investigated two categories of methods: (i) Sampled Query Set: in this approach, a large set of n-grams (mostly bi-grams) are sampled as queries from documents of the collection [3, 27]. (ii) Actual Query Logs: in this method, actual user-generated queries from the log of a retrieval system are employed for the purpose of retrievability bias estimation [25]. In this work, we use Actual Query Logs as they are directly representative for human-generated queries; to this end, we employ the queries from the MS MARCO Passage Ranking

[1] https://github.com/castorini/pyserini/

benchmark[2] [4] which contains ~500k train queries [4]. In our experiments we use 250K queries out of these training queries for estimating the retrievability bias and refer to this set of original training queries as OTQ. The other 250k half of training queries are then used for training the Bi-Encoder BERT ranker and the Cross-Encoder BERT re-ranker with various size of training query sets as described in Section 3.3.

3.2 Query Generation: GeQ

Recently, transformer-based generative models have shown promising results in text generation [2, 8], particularly in generating queries from a given document [21, 22]. In addition to OTQ, we investigate the query generation with transformer-based generative model, docT5query which is specifically trained on the task of generating queries out of a given document [21, 22] for the purpose of document expansion. We use docT5query to generate and sample 250k synthetic queries (same size as OTQ). To this end, first we uniformly sample 250k passages out of more than 8 Million passages in the MS MARCO Passage Ranking collection. We then feed each of 250k passages to docT5query, and we generate one query for each passage. We refer to this set of 250k generated queries as GeQ. Table 1 shows the statistics of OTQ and GeQ, indicating that the mean and variance for both query sets are comparable; however, GeQ has a lower kurtosis than OTQ.

Table 1: Statistics of query sets OTQ and GeQ.

Query Set	Mean	Variance	Kurtosis
OTQ	5.98	5.91	3.55
GeQ	6.38	6.05	2.87

3.3 Experimental Details

We train each of the Bi-Encoder and Cross-Encoder rankers with various training query set sizes starting from 50k up to 250k with steps of 50k. These 250k queries come from the MS MARCO training queries (excluding queries in OTQ) as described in Section 3.1. We do not utilize specific training approaches for the bi-encoder rankers such as data augmentation, knowledge distillation, or different negative sampling techniques [18, 33] as exploring the effect of such methods is beyond the purpose of this paper. In these experiments, we set 32 as the training batch size for the Cross-Encoder BERT ranker and 64 for our Bi-Encoder BERT ranker limited by the computational capacity. As such, for each of the training queries we took 50 triplets each of which includes a positive passage and a BM25-sampled negative passage [13, 14]. For the implementation of the cross-encoders and bi-encoder ranker we use the Sentence-Transformer[3], PyTorch and Huggingface Transformers [32] models. Additionally, we use Pyserini [16] for indexing and retrieval with Bi-Encoders. Finally, it should be noted that we use BM25 as our initial ranker for all cross-encoder re-rankers, with a re-ranking depth of 100. All code used in this paper is publically available.[4]

3.4 Evaluation

To evaluate retrievability bias with Gini as described in Section 2.1, following prior work [3, 27, 29], we explored the Gini parameter space b=[0, 1] and k={10, 20, ..., 100}; however as we saw no substantial difference, and since we use 100 as re-ranking depth, we only report results for b={0, 0.5, 1} and k={10, 50}. We evaluate bias for both OTQ and GeQ, but we only report effectiveness on OTQ as queries generated with docT5query, i.e., GeQ, are susceptible to hallucination [11] and therefore, are not guaranteed to be relevant to the passages used to generate those queries. Following prior work [12, 13, 20], we evaluate the effectiveness using nDCG@10 and MRR@10 as the official evaluation metric for this set.

[2]https://github.com/microsoft/msmarco/blob/master/Datasets.md
[3]https://github.com/UKPLab/sentence-transformers
[4]https://github.com/aminvenv/retbias

4 RESULTS AND ANALYSIS

RQ1: BM25 versus PLM-based Rankers. Table 2 shows the results for the retrievability bias of the Bi-Encoder BERT ranker and the Cross-Encoder BERT ranker trained with all training samples, i.e., 250k queries. We can see that both BERT rankers show lower retrievability bias than BM25. As it can be seen, bias of PLM-based models is consistently lower than BM25 for all values of the discounting factor b and the ranking cut-off k. In other words, PLM-based rankers not only show a higher effectiveness than BM25, they also provide a lower bias than this lexical ranker which was previously shown to be one of the least biased models [3, 29].

Additionally, Figure 1 shows the retrievability bias of BERT rankers trained with various training query set sizes and Figure 2 shows the increasing effectiveness for increasing training data size. By comparing Figure 2 and 1 we can see that while using more training queries results in higher effectiveness for both models, it does not necessarily lead to lower retrievability bias. For instance, in Figure 1, *Bias(150k)* $\approx$ *Bias(100k)* on both OTQ and GeQ for the Bi-Encoder ranker with $k \in \{10, 50\}$, and *Bias(100k)* > *Bias(50k)* on both OTQ and GeQ for Cross-Encoder with $k = 50$.

This indicates that *which* queries we use for training is also of importance besides *how many* queries the model is trained on as to the retrievability bias of the model during inference. Moreover, we can see that retrievability bias scores corresponding to Cross-Encoders are not showing much variation with different training set sizes. This might suggest that Cross-Encoder re-rankers are less susceptible to retrieval biases caused by the training data. However, we should note that re-ranking with deeper re-ranking depths could be investigated in future work.

RQ2: Synthetic Query Generation. Apart from the impact of training query set size on retrievability bias estimation, Figure 1 also shows the bias estimation using queries generated with docT5query model, i.e., GeQ. It is intuitively clear in these figures that there exists a strong correlation between the bias values estimated with OTQ and the ones estimated with GeQ. More precisely, there is an almost perfect correlation (Pearson's $r \geq 0.9$ with p-value< 0.05) for all settings. Here, it should be noted that in the estimation of retrievability bias, as Eq 2 suggests, what is important is the relative bias of a model with respect to other models [28]. As such, the correlation between the results on OTQ and on GeQ could show that the generated queries of docT5query (GeQ) are promising in being used for comparing the estimated retrievability bias between models: the *relative* relationship between models with various training samples roughly follow the same pattern for both GeQ and OTQ.

Table 2: The effectiveness and retrievability bias results using OTQ. The retrievability bias is estimated using both cumulative ($b = 0$) and gravity-based ($b \in \{0.5, 1\}$) utility functions at rank cut-off values $k = \{10, 50\}$. Lower Gini corresponds to lower bias. † indicates statistically significant improvement over BM25 according to t-tests with ($p < 0.05$) and Bonferroni correction.

Method	Effectiveness ↑		Retrievability Bias (Gini) ↓					
			b=0		b=0.5		b=1	
	MRR@10	nDCG@10	k=10	k=50	k=10	k=50	k=10	k=50
BM25	0.1319	0.1658	0.3060	0.4676	0.3945	0.5243	0.5187	0.6578
Bi-Encoder	0.2232†	0.2690†	0.2857†	0.4490†	0.3689†	0.4974†	0.4942†	0.6298†
Cross-Encoder	**0.2537**†	**0.2966**†	**0.2396**†	**0.4373**†	**0.3362**†	**0.4759**†	**0.4754**†	**0.5976**†

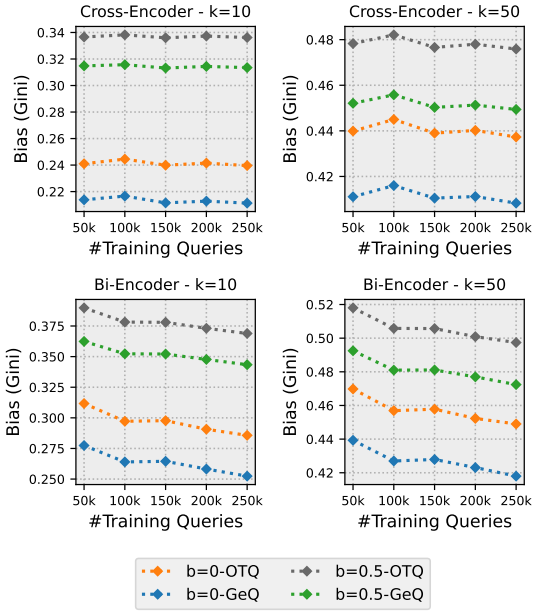

Figure 1: Retrievability Bias results with cumulative utility ($b = 0$) and gravity-based utility ($b = 0.5$) for the Cross-Encoder BERT Re-ranker and the Bi-Encoder BERT ranker using different training query set sizes.

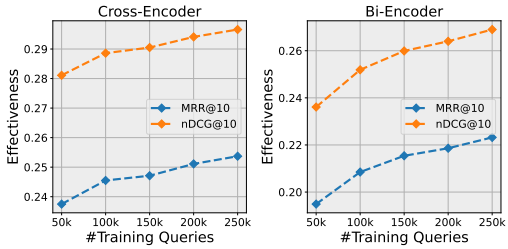

Figure 2: Effectiveness Results on OTQ for Cross-Encoder BERT re-ranker and Bi-Encoder BERT ranker using different training query set sizes.

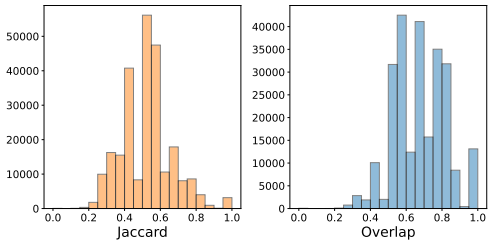

Figure 3: Distribution of maximum Jaccard similarity and overlap between each query of GeQ with queries of OTQ.

To better explore this correlation, we analyze the similarity between generated queries and user queries by computing the maximum Jaccard similarity ($\frac{|Q_1 \cap Q_2|}{|Q_1 \cup Q_2|}$) and overlap ($\frac{|Q_1 \cap Q_2|}{|Q_1|}$) between each query of GeQ with all queries of OTQ. Figure 3 shows the distribution of these measures for all queries of GeQ. These distributions indicate that there is indeed a difference in the queries of the two query sets GeQ and OTQ. However, the average of maximum Jaccard similarities and overlap scores (~0.517, ~0.669 respectively) also indicates that there is quite some overlap of query terms in the queries of GeQ over OTQ queries to which we might be able to relate the high correlation in the bias estimation with the two query sets. Moreover, Figure 1 shows that retrievability bias scores estimated with GeQ are lower than the bias scores estimated with OTQ for both the Bi-Encoder and Cross-Encoder BERT rankers with different amount of training queries. This suggests that the user-generated training queries may actually be more biased than the synthetically generated training queries – and that by using synthetic queries in training data it may be possible to further reduce the biases. This presents an interesting direction for future work where biases in human generated data could be mitigated through generative approaches.

5 CONCLUSION

In this work, we study the retrievability bias in ranking with pre-trained language models. To this aim, we compare neural ranking models including a Bi-Encoder BERT ranker and a Cross-Encoder BERT ranker to the widely-used lexical ranker, BM25. Our findings show that in addition to being more effective, PLM-based rankers are less biased than the traditional ranking model, BM25. We also perform an initial study on the plausibility of sampled queries generated with transformer-based generation model to be used in the retrievability bias estimation. We find that leveraging user-generated queries leads to greater bias values for ranking models than when synthetically generated queries are being used. This finding could suggest that the training queries themselves may be encoding biases that we were previously unaware of. In future we plan to investigate the possibility of using generated queries for training PLM-based rankers to find out if this could further reduce the retrievability bias in these rankers. Concurrent work by Penha et al. [23] explores this direction, trying to improve the retrievability of items using controlled query generation. While we have focused on retrievability bias, which encodes the notion of fairness based on equality, in future work, we plan to examine more deeply how the training and learning influences other aspects of model bias, and how such biases can be further reduced.

REFERENCES

[1] Amin Abolghasemi, Arian Askari, and Suzan Verberne. 2022. On the interpolation of contextualized term-based ranking with bm25 for query-by-example retrieval. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*. 161–170.

[2] Arian Askari, Mohammad Aliannejadi, Evangelos Kanoulas, and Suzan Verberne. 2023. A Test Collection of Synthetic Documents for Training Rankers: ChatGPT vs. Human Experts. In *The 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023)*. ACM.

[3] Leif Azzopardi and Vishwa Vinay. 2008. Retrievability: An evaluation measure for higher order information access tasks. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 561–570.

[4] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[5] Shariq Bashir and Andreas Rauber. 2009. Improving Retrievability of Patents with Cluster-Based Pseudo-Relevance Feedback Documents Selection. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (Hong Kong, China) *(CIKM '09)*. Association for Computing Machinery, New York, NY, USA, 1863–1866. https://doi.org/10.1145/1645953.1646250

[6] Shariq Bashir and Andreas Rauber. 2011. On the Relationship between Query Characteristics and IR Functions Retrieval Bias. *J. Am. Soc. Inf. Sci. Technol.* 62, 8 (Aug. 2011), 1515–1532. https://doi.org/10.1002/asi.21549

[7] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Jun 2018). https://doi.org/10.1145/3209978.3210063

[8] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. InPars: Unsupervised Dataset Generation for Information Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2387–2392. https://doi.org/10.1145/3477495.3531863

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Joseph L Gastwirth. 1972. The estimation of the Lorenz curve and Gini index. *The review of economics and statistics* (1972), 306–316.

[11] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. 2023. Doc2Query: When Less is More. *The 45th European Conference on Information Retrieval* (2023).

[12] 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).

[13] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.

[14] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6769–6781.

[15] Ruohan Li, Jianxiang Li, Bhaskar Mitra, Fernando Diaz, and Asia J Biega. 2022. Exposing Query Identification for Search Transparency. In *Proceedings of the ACM Web Conference 2022*. 3662–3672.

[16] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073* (2021).

[17] 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.

[18] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. 163–173.

[19] Colin McLellan. 2019. *The relationship between retrievability bias and retrieval performance*. Ph.D. Dissertation. University of Glasgow.

[20] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[21] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint* 6 (2019).

[22] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[23] Gustavo Penha, Enrico Palumbo, Maryam Aziz, Alice Wang, and Hugues Bouchard. 2023. Improving Content Retrievability in Search with Controllable Query Generation. In *Proceedings of the ACM Web Conference 2023*. 3182–3192.

[24] 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.

[25] Dwaipayan Roy, Zeljko Carevic, and Philipp Mayr. 2022. Studying retrievability of publications and datasets in an integrated retrieval system. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*. 1–9.

[26] Myriam C Traub, Thaer Samar, Jacco Van Ossenbruggen, Jiyin He, Arjen de Vries, and Lynda Hardman. 2016. Querylog-based assessment of retrievability bias in a large newspaper corpus. In *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*. IEEE, 7–16.

[27] Colin Wilkie and Leif Azzopardi. 2014. Best and fairest: An empirical analysis of retrieval system bias. In *European Conference on Information Retrieval*. Springer, 13–25.

[28] Colin Wilkie and Leif Azzopardi. 2014. Efficiently estimating retrievability bias. In *European Conference on Information Retrieval*. Springer, 720–726.

[29] Colin Wilkie and Leif Azzopardi. 2014. A retrievability analysis: Exploring the relationship between retrieval bias and retrieval performance. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 81–90.

[30] Colin Wilkie and Leif Azzopardi. 2016. A topical approach to retrievability bias estimation. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*. 119–122.

[31] Colin Wilkie and Leif Azzopardi. 2017. Algorithmic bias: do good systems make relevant documents more retrievable?. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2375–2378.

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 38–45.

[33] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).