WSRR: Weighted Rank-Relevance Sampling for Dense Text Retrieval

Kailash Hambarde[1] and Hugo Proença[1]

IT: Instituto de Telecomunicações, University of Beira Interior, Portugal

Abstract. As in many other domains based in the *contrastive learning* paradigm, *negative sampling* is seen as a particular sensitive problem for appropriately training dense text retrieval models. For most cases, it is accepted that the existing techniques often suffer from the problem of uninformative or false negatives, which reduces the computational effectiveness of the learning phase and even reduces the probability of convergence of the whole process. Upon these limitations, in this paper we present a new approach for dense text retrieval (termed WRRS: Weighted Rank-Relevance Sampling) that addresses the limitations of current negative sampling strategies. WRRS assigns probabilities to negative samples based on their relevance scores and ranks, which consistently leads to improvements in retrieval performance. Under this perspective, WRRS offers a solution to uninformative or false negatives in traditional negative sampling techniques, which is seen as a valuable contribution to the field. Our empirical evaluation was carried out against the AR2 baseline on two well known datasets (NQ and MS Doc), pointing for consistent improvements over the SOTA performance.

Keywords: Dense text retrieval, Relevance score, Negative sampling

1 Introduction

The field of information retrieval has undergone significant advances with the growing amount of textual data, making the task of retrieving relevant information from large text sources a crucial problem. In this context, queries and documents are typically represented by low-dimensional vectors, from where similarity metrics are used to perceive the relevance of a document with respect to a query [1][2][3][4][5]. Despite the widespread use of this kind of methods, the main challenge in training dense text retrieval models lies in selecting appropriate negatives from a large pool of documents during negative sampling [1].

The most commonly used negative sampling methods, such as random negative sampling [6][1] and top-k hard negatives sampling [9][8], have obvious limitations: random negative sampling tends to select uninformative negatives, while top-k sampling may include false negatives [9][4]. To address these limitations, this paper proposes a novel approach, termed WRRS: Weighted Rank-Relevance Sampling, for the dense text retrieval task. When compared to the SOTA, the key point is that WRRS assigns a probability to each negative sample based on its relevance score *and* rank, and subsequently selects negative samples based on the joint probability from both factors, which enables to select *better* sets of negative instances, allowing for consistent improvements in the training phase of dense text retrieval methods.

Preliminary

The dense text retrieval task (DTR) refers to retrieve the k most relevant documents from a large candidate pool, given a query, q. Upon its efficiency, the dual-encoder architecture is the most widely used in DTR, consisting of a query encoder, E_q, and a document encoder, E_d. The two encoders map the query and the document into k-dimensional dense vectors, h_q and h_d, respectively. The semantic relevance score between q and d can be computed as follows:

$$s(q, d) = h_q \cdot h_d. \tag{1}$$

Recently, pre-trained language models (PLMs) have also adopted the dual-encoders in DTR and the representation of the [CLS] tokens done by mean of dense vectors [10].

Overall, according to a contrastive learning paradigm, the objective of the dense text retrieval task is to maximize the semantic relevance between the query q and the most relevant documents D^+, while minimizing the semantic relevance between q and any other irrelevant documents, $D^- = D\ D^+$. As in many other contrastive learning-based applications, training this kind of models is seen as a highly sensitive task, being negative sampling commonly used to speed up the training process and increase the probabilities of convergence. This step involves either randomly sampling negatives or selecting the top-k hard negatives ranked by BM25 or the dense retrieval model itself [1][4]. The optimization objective can be formulated as follows:

$$\theta^* = \arg\min_{\theta} \sum_{q} \sum_{d^+ \in D^+} \sum_{d^- \in D^-} e^{-L(s(q,d^+), s(q,d^-))}, \tag{2}$$

where $L(.)$ is the loss function.

The remainder of this paper is organized as follows: Section 2, overviews the commonly used negative sampling methods in dense text retrieval, including random negative sampling and top-k hard negatives sampling and other. Section 3 presents the proposed WRRS approach in detail. Section 4 reports our experiments, comparing WRRS performance to baseline negative sampling methods. Finally, Section 5 summarizes the contributions of this paper and highlights the significance of the proposed WRRS approach for dense text retrieval.

2 Related Work

Recent years have seen significant advancements in dense retrieval methods for text retrieval tasks [19][20][21][22]. Unlike traditional sparse retrieval methods such as TF-IDF and BM25, dense retrieval methods convert queries and documents into low-dimensional dense vectors, which are then compared using vector distance metrics (e.g., cosine similarity) for retrieval. To learn an effective dense retrieval model, it is critical to sample high-quality negatives that are paired with the given query and positive samples.

Early works in the field [1][23] mostly utilized in-batch random negatives and hard negatives sampled by BM25. Later, a series of studies [4][9] found that using top-k ranked examples as hard negatives, selected by the dense retriever, is more effective in improving the performance of the retriever. Some methods [9][16] adopt a dynamic sampling strategy

that actively selects top-k hard negatives once after a set interval during training. However, these top-k negative sampling strategies can easily lead to selection of higher-ranking false negatives for training.

To address this issue, previous works have incorporated techniques such as knowledge distillation [4][24][25], pre-training [26][27], and other denoising techniques [28][29] to alleviate this problem. Despite their effectiveness, these methods often rely on complicated training strategies or additional complementary models.

Negative Sampling In the field of dense text retrieval, the selection of negative instances has been commonly seen as a crucial role for appropriately training this kind of models. According to a classical *constrastive learning* paradigm, negative samples are used to help the model to discriminate between relevant and irrelevant passages. Three types of negatives are typically considered in [1]: 1) The first one regards random negatives, which are any random passages from the corpus. The authors highlight the drawback of using random negatives, as they may not be semantically or contextually relevant to the question, leading to false negatives; 2) The second type consists of BM25 negatives, which are top passages returned by the BM25 retrieval algorithm that match the majority of question tokens but do not contain the answer. The authors find that BM25 negatives may also result in irrelevant or nonsensical passages being selected as negatives; Finally, 3) the third type is gold negatives, which are positive passages paired with other questions that appear in the training set. The authors mention that gold negatives may not generalize well to new questions or represent real-world scenarios, and may also not be diverse enough to cover all possible incorrect answers. To address the issue of constructing uninformative negatives, [9] proposes a novel method called "Approximate Nearest Neighbor Noise Contrastive Estimation" (ANCE). This method samples negatives globally from the corpus and uses an asynchronously updated Approximate Nearest Neighbor (ANN) index to retrieve top documents via the Dense Retrieval (DR) model. The results show the importance of constructing negatives globally to improve learning convergence. [11] proposes a multi-stage training approach that improves negative contrast in neural passage retrieval. The approach consists of six stages, starting with random sampling of negatives from the corpus and ending with the selection of negatives based on the outputs of a fine-tuned neural retrieval model. The authors suggest that this multi-stage approach allows for the selection of negatives that are more representative of the true negatives, leading to improved learning convergence and performance. [12] proposes a new approach to negative sampling called SimANS for training dense retrieval models. The authors observe that ambiguous negatives, which are negatives ranked near the positives, are more informative and less likely to be false negatives. Therefore, SimANS incorporates a new sampling probability distribution that samples more ambiguous negatives. The experiments show that SimANS outperforms other negative sampling methods and provides a promising solution to the problem of negative sampling in dense retrieval models.

Two training algorithms for Dense Retrieval (DR) models named Stable Training Algorithm for dense Retrieval (STAR) and query-side training Algorithm for Directly Optimizing Ranking performance (ADORE) are proposed in [16]. STAR aims to improve the stability of DR training by introducing random negatives, while ADORE replaces the commonly used static hard negative sampling method with a dynamic one to directly optimize the ranking performance. [17] focuses on training sparse representation learning-based neural retrievers using hard-negative mining and Pre-trained Language Model initialization. This work is based on SPLADE, a sparse expansion-based retriever,

and aims to improve its performance and efficiency in both in-domain and zero-shot settings. The results showed that the use of hard-negative mining and Pre-trained Language Model initialization led to state-of-the-art results and demonstrated the effectiveness of these techniques for sparse representation learning-based neural retrievers. [18] a new negative sampling strategy, Cross-Batch Negative Sampling (CBNS), proposed for the training of two-tower recommender system models. This strategy takes advantage of the encoded item embeddings from recent mini-batches to improve the model training, effectively reducing the memory and computational costs associated with large batch size training. The results of both theoretical analysis and empirical evaluations demonstrate the effectiveness and efficiency of CBNS in comparison to existing strategies.

In this context, the Weighted Rank Relevance Sampling (WRRS) algorithm, as described in Algorithm 1, presents a new approach to negative sampling in dense retrieval that addresses the limitations of previous methods. The WRRS algorithm first builds an approximate nearest neighbor (ANN) index using the dense retrieval model, retrieves the top-k ranked negatives for each query with their relevance scores, computes the relevance scores of each query and its positive documents, sorts the negatives for each query based on relevance scores, and generates probabilities for each negative sample based on its rank and relevance score. The algorithm then uses these probabilities to sample negatives for each instance in the batch during optimization of the dense retrieval model. This approach aims to generate a more diverse set of negatives and reduce the likelihood of sampling higher-ranking false negatives during training.

3 Proposed Approach

It should be stressed that "WRRS - Weighted Rank-Relevance Sampling" can be seen as an extension of [12], by assigning a probability to each negative sample not only based on its relevance score, but also on its position (rank) in the retrieved list. The probability (π) is obtained using two methods: Using the rank of the negative sample and its relevance score. The formula is:

$$\pi = (k - rank(D_-))/(k * (k + 1))/2, \tag{3}$$

where k is the number of retrieved negatives, D_- is the negative sample, and $rank(D_-)$ is its rank in the retrieved list. Using a weighted method that combines the relevance score and rank:

$$\pi = \alpha * f(|s(q, D_-)s(q, d+)b|) + (1-) * f(r(q, D_-)), \tag{4}$$

where α is a weighting factor between 0 and 1. If α is 0, the relevance score will not be considered in the calculation, and the probability will be based solely on the rank. If α is 1, the rank will not be considered, and the probability will be based solely on the relevance score. For values of α between 0 and 1, the rank and relevance score will be considered according to the value of α, with higher values giving more weight to the relevance score and lower values giving more weight to the rank. $s(q, d+)$ is the average relevance score of the positive documents, b is a bias term, $r(q, D_-)$ is the rank of the negative document D_-. The negative samples are then selected based on the calculated probability π for each sample.

Algorithm 1 Weighted Rank Relevance Sampling

Input: Queries and their positive documents (q, D_+), document pool D, pre-learned dense retrieval model M

Output: New training data (q, D_+, D_-)

1: Build the ANN index on D using M.
2: Retrieve the top-k ranked negatives $D\_$ for each query with their relevance scores $s(q, D_-)$ from D.
3: Compute the relevance scores of each query and its positive documents $s(q, D+)$.
4: Sort negatives D_- for each query based on their relevance scores $s(q, D_-)$ in descending order
5: Generate the probability to each negative sample based on its rank and relevance score using equation Eq. 4.
6: Construct new training data (q, D_+, D_-)
7: **while** M has not converged **do**
8: Sample a batch from $(q, D+, D\_)$
9: Sample negatives for each instance from the batch according to pi
10: Optimize parameters of M using the batch and sampled negatives.
11: **end while**

WRRS is a method for obtaining negatives from a given mini-batch. The method consists of three key steps, as described below. Step 1: Selection of Top-k Ranked Negatives [9] [4] This step is similar to previous methods for selecting the top-k ranked negatives from the candidate pool $(D\backslash D+)$ using an approximate nearest neighbor (ANN) search tool such as FAISS [13]. Step 2: Computation of Sampling Probabilities, in this step, the sampling probabilities for all the top-k ranked negatives are computed using equation 4. Step 3: Sampling of Negatives, the negatives are sampled according to their computed sampling probabilities. The overall algorithm is presented in Algorithm 1.

4 Experiments and Results

To evaluate the effectiveness of WRRS, experiments were conducted on two public text retrieval datasets: MSMARCO Document Ranking (MS Doc) [14] and NQ [15]. The statistics of these datasets are presented in Table 1. Detailed information about the baselines, and implementations can be found in [12].

Table 1. Statistics of the retrieval datasets

Dataset	Training	Dev	Test	Documents
MS DOC	367,013	5,193	-	3,213,835
NQ	58,880	8,757	3,610	21,015,324

The MS Doc dataset consists of 3,213,835 documents with 367,013 instances in the training set and 5,193 instances in the development set. The NQ dataset consists of 21,015,324 documents with 58,880 instances in the training set and 8,757 instances in the

development set and 3,610 instances in the test set. As a baseline, we used the AR2 [8] retrieval method.

Table 2. Comparison between the Mean reciprocal Rank (MRR) values for the WRRS and the baselines considered, on the MS Doc development set.

Method	MRR@10	R@100
BM25	0.279	0.807
AR2	0.418	0.914
WRRS	0.425	0.917

The performance of, BM25, AR2, and WRRS, were evaluated on a Microsoft Document (MS Doc) development set, as shown in Table 2. The performance measured using two metrics, MRR@10 (Mean Reciprocal Rank at 10) and R@100 (Recall at 100). As seen in Table 2, both AR2 and WRRS outperform BM25 on both MRR@10 and R@100, with scores of 0.418 and 0.425 for MRR@10, and 0.914 and 0.917 for R@100, respectively. BM25, on the other hand, has scores of 0.279 for MRR@10 and 0.807 for R@100, indicating that it performs relatively poorly compared to AR2 and WRRS.

Table 3. Comparing Retrieval Performance on the NQ Test Set using BM25, AR2, and WRR

Method	R@5	R@20	R@100
BM25	-	59.1	73.7
AR2	77.9	86.2	90.1
WRRS	78.0	85.7	90.3

The Table 3 provides the performance of, BM25, AR2, and WRRS, on a NQ test set. As shown in the Table 3, AR2 and WRRS perform similarly and outperform BM25 on all three metrics. AR2 has R@5 score of 77.9, R@20 score of 86.2, and R@100 score of 90.1. WRRS has R@5 score of 78.0, R@20 score of 85.7, and R@100 score of 90.3. BM25, on the other hand, R@20 score of 59.1, and R@100 score of 73.7. This suggests that WRRS perform better than AR2 and BM25 in retrieving relevant items.

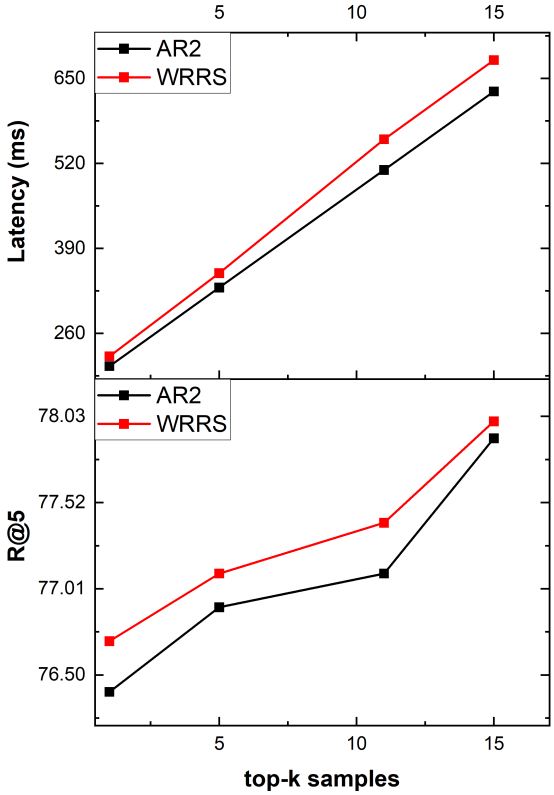

Fig. 1. Retrieval performance and training latency with regard to different k-sampled negative values on the NQ dataset.

Ablation Studies

In our ablation experiments, two algorithms (AR2 baseline and WRRS) were tested and evaluated on NQ based on their recall at 5 (R@5) and latency. The results of this evaluation are presented in Fig 1.

R@5 is a commonly used metric in the evaluation of information retrieval systems, and it represents the fraction of relevant items among the first five retrieved items. In this study, the WRRS algorithm consistently outperforms the AR2 algorithm in terms of R@5 recall, with higher scores for each rank in the test cases. This suggests that the WRRS algorithm is more effective at retrieving relevant items in an information retrieval system. However, the latency of the WRRS algorithm is higher than that of the AR2 algorithm. Latency refers to the amount of time it takes for a system to respond to a request, and lower latency is generally preferable in information retrieval systems.

5 Conclusions

In this paper, we proposed a new strategy (WRRS) for selecting negative samples to be used in the training phase of contrastive learning-based dense text retrieval methods. The proposed strategy was empirically compared to two baseline retrieval methods - BM25 and AR2 - on two public text retrieval datasets: MSMARCO Document Ranking and NQ. The obtained results show that WRRS consistently outperformed BM25 on both MRR@10 and R@100 metrics on the MS Doc development set and R@5, R@20, and R@100 metrics on the NQ test set. The performance of AR2 and WRRS was found to be similar, with WRRS providing slightly better results. These findings suggest that WRRS is a promising approach for text retrieval, that can be useful for a broad range of applications. Further research involves to perceive the actual changes that should be applied to the described WRRS, in order to extend its capabilities to signficantly different domains.

Acknowledgement

The author would like to thank to AddPath - Adaptative Designed Clinical Pathways Project (CENTRO-01-0247-FEDER-072640 LISBOA-01-0247-FEDER-072640). This work is funded by FCT/MCTES through national funds and co-funded by EU funds under the project UIDB/50008/2020.

References

1. Karpukhin, Vladimir, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. "Dense passage retrieval for open-domain question answering." arXiv preprint arXiv:2004.04906 (2020).
2. Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).
3. Brickley, Dan, Matthew Burgess, and Natasha Noy. "Google Dataset Search: Building a search engine for datasets in an open Web ecosystem." In The World Wide Web Conference, pp. 1365-1375. 2019.
4. Qu, Yingqi, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. "RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering." arXiv preprint arXiv:2010.08191 (2020).
5. Izacard, Gautier, and Edouard Grave. "Leveraging passage retrieval with generative models for open domain question answering." arXiv preprint arXiv:2007.01282 (2020).
6. Luan, Yi, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. "Sparse, dense, and attentional representations for text retrieval." Transactions of the Association for Computational Linguistics 9 (2021): 329-345.
7. Xiong, Lee, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. "Approximate nearest neighbor negative contrastive learning for dense text retrieval." arXiv preprint arXiv:2007.00808 (2020).
8. Zhang, Hang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. "Adversarial retriever-ranker for dense text retrieval." arXiv preprint arXiv:2110.03611 (2021).
9. Xiong, Lee, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. "Approximate nearest neighbor negative contrastive learning for dense text retrieval." arXiv preprint arXiv:2007.00808 (2020).

10. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
11. Lu, Jing, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. "Multi-stage training with improved negative contrast for neural passage retrieval." In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6091-6103. 2021.
12. Zhou, Kun, Yeyun Gong, Xiao Liu, Wayne Xin Zhao, Yelong Shen, Anlei Dong, Jingwen Lu et al. "Simans: Simple ambiguous negatives sampling for dense text retrieval." arXiv preprint arXiv:2210.11773 (2022).
13. Johnson, Jeff, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with gpus." IEEE Transactions on Big Data 7, no. 3 (2019): 535-547.
14. Nguyen, Tri, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. "MS MARCO: A human generated machine reading comprehension dataset." choice 2640 (2016): 660.
15. Kwiatkowski, Tom, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein et al. "Natural questions: a benchmark for question answering research." Transactions of the Association for Computational Linguistics 7 (2019): 453-466.
16. Zhan, Jingtao, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. "Optimizing dense retrieval model training with hard negatives." In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1503-1512. 2021.
17. Formal, Thibault, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. "From distillation to hard negative sampling: Making sparse neural ir models more effective." In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2353-2359. 2022.
18. Wang, Jinpeng, Jieming Zhu, and Xiuqiang He. "Cross-batch negative sampling for training two-tower recommenders." In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1632-1636. 2021.
19. Zhan, Jingtao, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. "Repbert: Contextualized text embeddings for first-stage retrieval." arXiv preprint arXiv:2006.15498 (2020).
20. Hong, Wu, Zhuosheng Zhang, Jinyuan Wang, and Hai Zhao. "Sentence-aware contrastive learning for open-domain passage retrieval." In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1062-1074. 2022.
21. Ram, Ori, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. "Learning to retrieve passages without supervision." arXiv preprint arXiv:2112.07708 (2021).
22. Zhou, Kun, Beichen Zhang, Wayne Xin Zhao, and Ji-Rong Wen. "Debiased contrastive learning of unsupervised sentence representations." arXiv preprint arXiv:2205.00656 (2022).
23. Min, Sewon, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. "AmbigQA: Answering ambiguous open-domain questions." arXiv preprint arXiv:2004.10645 (2020).
24. Ren, Ruiyang, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. "Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking." arXiv preprint arXiv:2110.07367 (2021).
25. Lu, Yuxiang, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian et al. "Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval." arXiv preprint arXiv:2205.09153 (2022).
26. Zhou, Jiawei, Xiaoguang Li, Lifeng Shang, Lan Luo, Ke Zhan, Enrui Hu, Xinyu Zhang et al. "Hyperlink-induced pre-training for passage retrieval in open-domain question answering." arXiv preprint arXiv:2203.06942 (2022).
27. Xu, Canwen, Daya Guo, Nan Duan, and Julian McAuley. "Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval." arXiv preprint arXiv:2203.06169 (2022).
28. Mao, Kelong, Zhicheng Dou, and Hongjin Qian. "Curriculum Contrastive Context Denoising for Few-shot Conversational Dense Retrieval." In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 176-186. 2022.

29. Hofstätter, Sebastian, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. "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, pp. 113-122. 2021.