Recovering Gold from Black Sand: Multilingual Dense Passage Retrieval with Hard and False Negative Samples

Tianhao Shen[1*], **Mingtong Liu**[2], **Ming Zhou**[2], **Deyi Xiong**[1†]
[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Beijing Lanzhou Technology Co., Ltd., Beijing, China
`{thshen,dyxiong}@tju.edu.cn`,
`{liumingtong,zhouming}@langboat.com`

Abstract

Negative samples have not been efficiently explored in multilingual dense passage retrieval. In this paper, we propose a novel multilingual dense passage retrieval framework, mHFN, to recover and utilize hard and false negative samples. mHFN consists of three key components: 1) a multilingual hard negative sample augmentation module that allows knowledge of indistinguishable passages to be shared across multiple languages and synthesizes new hard negative samples by interpolating representations of queries and existing hard negative samples, 2) a multilingual negative sample cache queue that stores negative samples from previous batches in each language to increase the number of multilingual negative samples used in training beyond the batch size limit, and 3) a lightweight adaptive false negative sample filter that uses generated pseudo labels to separate unlabeled false negative samples and converts them into positive passages in training. We evaluate mHFN on Mr. TyDi, a high-quality multilingual dense passage retrieval dataset covering eleven typologically diverse languages, and experimental results show that mHFN outperforms strong sparse, dense and hybrid baselines and achieves new state-of-the-art performance on all languages. Our source code is available at `https://github.com/Magnetic2014/mHFN`.

1 Introduction

Passage retrieval, which matches relevant passages to queries, is an important task in Information Retrieval (IR). It can be also integrated as a core component to solve many Natural Language Processing (NLP) problems, e.g., open domain question answering (Chen et al., 2017), fact checking (Thorne et al., 2018), etc. Powered by large scale pretrained language models (e.g. BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and T5 (Raffel et al., 2020)), dense passage retrieval, which explores dense vector representations to match relevant passages, has attracted growing interest (Gao et al., 2020; Khattab and Zaharia, 2020; Karpukhin et al., 2020; MacAvaney et al., 2020a; Qu et al., 2021; Zhan et al., 2021; Xu et al., 2022; Gao et al., 2021a). Dense passage retrieval models usually adopt a bi-encoder (also known as dual-encoder) architecture, one encoder for encoding queries, the other for learning passage representations, which can be computed offline. With dense representations of a query and document, passage ranking is recast as a nearest neighbor search problem that can be efficiently solved by similarity search toolkits tailored for dense vectors, such as Faiss (Johnson et al., 2019).

Dense passage retrieval models are usually trained via Contrastive Learning (CL), which encourages query representations to be close to positive (i.e., relevant) passages and away from negative (i.e., irrelevant) passages in the learned semantic space. Many previous studies demonstrate the effectiveness of CL in dense passage retrieval (Karpukhin et al., 2020; Qu et al., 2021; Xu et al., 2022). Under this CL-based dense passage retrieval setting, using more negative samples has proved beneficial to models (Wu et al., 2020; Chen et al., 2020; He et al., 2020; Giorgi et al., 2021; Gao et al., 2021c). Specifically, hard negative samples (i.e., negative samples which are similar to positive samples) are more desirable than ordinary negative samples for tuning the dense representations of queries and passages (Qu et al., 2021; Zhan et al., 2021).

As multilingual pretrained language models (e.g., mBERT (Devlin et al., 2019)) have exhibited cross-lingual generalization and knowledge transfer from high-resource to low-resource languages, it is natural to extend the monolingual DPR model (Karpukhin et al., 2020) to multilingual DPR

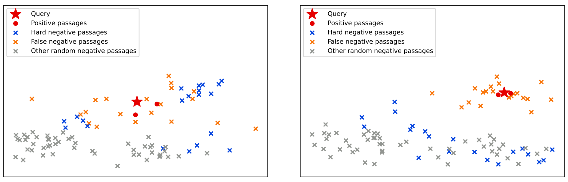
[*] Work done as an intern at Beijing Lanzhou Technology Co., Ltd., Beijing, China.
[†] Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10659 - 10670
December 7-11, 2022 ©2022 Association for Computational Linguistics

(e.g., mDPR (Zhang et al., 2021)) by replacing the monolingual backbone model with a multilingual pretrained language model. Such a multilingual setting is promising for languages without sufficient training data.

Unfortunately, negative samples are not efficiently explored in existing multilingual dense passage retrieval models. First, current multilingual dense passage retrieval models develop hard negative samples for each language in a separate and independent way, which is unable to share the common features of indistinguishable passages across languages. Second, in order to increase the number of negative samples, the in-batch and cross-batch negative technique are widely used in dense passage retrieval (Karpukhin et al., 2020; Qu et al., 2021), where negative samples are sampled from the same batch as positive samples and shared across all GPUs. However, this will quickly exhaust GPU memory with growing negative samples, hence making it difficult to increase the number of negative samples further. Third, as the number of negative samples increases, false negative samples (i.e., unlabeled relevant passages) will also grow in number, which would make it harder for training to converge and mislead the direction of optimization since false negative samples are actually positive. As shown in the Figure 1, it is desirable to make representations of unlabeled false negative samples close to positive samples and keep hard negative samples as distant as possible.

In this paper, our key interest lies in a multilingual dense passage retrieval task, where a unified retrieval model is used to retrieve in-language (i.e., in the same language as the given query) relevant passages for multiple languages. To efficiently utilize negative samples (especially hard and false negative samples) and solve the aforementioned issues, we propose mHFN, a new **m**ultilingual dense passage retrieval framework with **H**ard and **F**alse **N**egative samples. To model hard negative samples, we propose a multilingual hard negative sample augmentation module that shares hard negative samples across languages and generates synthetic negative sample representations. In this way, we force the model to learn efficient features to distinguish between positive and hard negative samples that are similar to each other. To handle false negative samples, we introduce a multilingual negative sample cache queue that stores negative samples from previous batches in each language as candidates. An adaptive false negative sample filter is then adopted to filter out the unlabeled false negative samples from in-batch and cached negative samples. The filtered false negative samples are further used as positive samples to boost training. mHFN achieves a balance between the quantity and quality of negative samples by recovering the most valuable negative samples (i.e., hard and false negative samples) from all negative samples.



(a) mDPR (b) mHFN

Figure 1: Compared to mDPR (Zhang et al., 2021), mHFN learns better passage representations, especially for hard and false negative samples.

Our contributions can be summarized as follows:

- We propose a multilingual hard negative sample augmentation module to share hard negative samples across languages and generate diverse augmented negative samples, which improves both the quantity and quality of hard negative samples.
- In order to add more candidate negative samples beyond the batch size limit, we present a multilingual negative cache queue to store negative samples in each language. This queue is dynamically updated to keep the encoded representations in the similar hidden space and consistent with the encoder in the current training step.
- To adaptively separate the unlabeled false negative samples from all candidate negative samples, we propose a lightweight adaptive false negative sample filter, which uses generated pseudo labels to filter out false negative samples. The recovered false negative samples are used as positive instances to speed up the convergence of training.
- Experimental results show that mHFN significantly outperforms state-of-the-art baselines, by 7.9% and 6.5% in average MRR@100 and Recall@100 over all languages respectively on Mr. TyDi (Zhang et al., 2021), a high-quality multilingual passage retrieval benchmark dataset.

2 Related Work

Monolingual Dense Passage Retrieval The past few years have witnessed growing interest in monolingual dense passage retrieval. DPR (Karpukhin et al., 2020) is built on a bi-encoder architecture, which is initialized with BERT (Devlin et al., 2019) and outperforms early dense retrieval methods. RocketQA (Qu et al., 2021) first mines hard negative samples with a trained retrieval model and then uses the mined negative samples to re-train the model. However, RocketQA requires a pre-trained cross-encoder to filter out false negative samples, which is not efficient and must be trained in advance. TAS-B (Hofstätter et al., 2021) is also a retrieval model using the bi-encoder architecture. It utilizes topic-aware sampling to improve training with in-batch negative samples, and applies a dual-teacher supervision paradigm to achieve better knowledge distillation from both a cross-encoder and a ColBERT (Khattab and Zaharia, 2020) teacher model simultaneously. Other studies further apply hard negative sample mining to train dense passage retrieval models. Gao et al. (2021b) and Karpukhin et al. (2020) use BM25 (Robertson et al., 2009) top passages as hard negative samples. ANCE (Xiong et al., 2021) enhances hard negative sampling by dynamically mining hard negative samples in the training phase. However, it requires periodically rebuilding the index and refreshing hard negative samples, which greatly increases computational cost. Zhan et al. (2021) combine static BM25 hard negative samples with dynamic hard negative samples retrieved from the entire corpus by the model at the current training step.

Multilingual and Cross-lingual Dense Passage Retrieval Researchers have been utilizing cross-lingual knowledge transfer to enhance monolingual retrieval for low-resource languages since the advent of multilingual pretrained language models. Both MacAvaney et al. (2020b) and Shi et al. (2020) investigate zero-shot transfer using a cross-encoder architecture, in which they first fine-tune mBERT on the source language, then apply the model to the target language directly. However, the cross-encoder architecture they use is slow in practice. In contrast, bi-encoders equipped with nearest neighbor vector search tools such as Faiss (Johnson et al., 2019) run much faster than cross-encoders since the dense representations can be computed and indexed in advance. Asai et al. (2021b) also utilize mBERT to perform retrieval in the many-to-many scenario. By retrieving in a multilingual pool, they train a model to answer a query in any specific languages. However, this diverges from our setting, where we solely focus on using a unified model to conduct in-language retrieval for multiple languages.

For datasets, Mr. TyDi[1] (Zhang et al., 2021) collects queries and passages in eleven typologically diverse languages from Wikipedia. To the best of our knowledge, it is the only publicly available dataset that can be used for our multilingual passage retrieval experiments. Other datasets, such as XOR-TyDi (Asai et al., 2021a) and CLIRMatrix (Sun and Duh, 2020), focus on cross-lingual retrieval instead. Based on the Mr. TyDi dataset, Zhang et al. (2022) empirically investigate the best practice of training multilingual dense passage retrieval models. However, they have not studied the impact of the number of negative samples and the side effect of false negative samples.

3 Methodology

In this section, we elaborate the proposed mHFN, as illustrated in Figure 2, which aims to utilize hard and false negative samples in multilingual dense passage retrieval. It is comprised of three essential components: a multilingual hard negative sample augmentation module that shares hard negative knowledge across languages, a multilingual negative sample cache queue that stores extra negative samples in each language beyond the limit of in-batch negative samples, and an adaptive false negative sample filter that filters out unlabeled false negative samples from candidate negative samples during training.

3.1 The Bi-Encoder Architecture

Our multilingual passage retrieval model mHFN is developed with the bi-encoder architecture, where a retrieval model uses a query encoder $E_q(\cdot)$ and a passage encoder $E_p(\cdot)$ to obtain the query and passage representations respectively. The dot product between the query $\boldsymbol{q}$ and a candidate passage $\boldsymbol{p}$ can then be computed as their similarity:

$$\text{sim}(\boldsymbol{q}, \boldsymbol{p}) = E_q(\boldsymbol{q}) \cdot E_p(\boldsymbol{p}) \tag{1}$$

For a query $\boldsymbol{q}$, the top-k most similar passages will be retrieved. In order to achieve efficient retrieval, it is preferable to separate the encoding

[1] https://github.com/castorini/Mr.TyDi

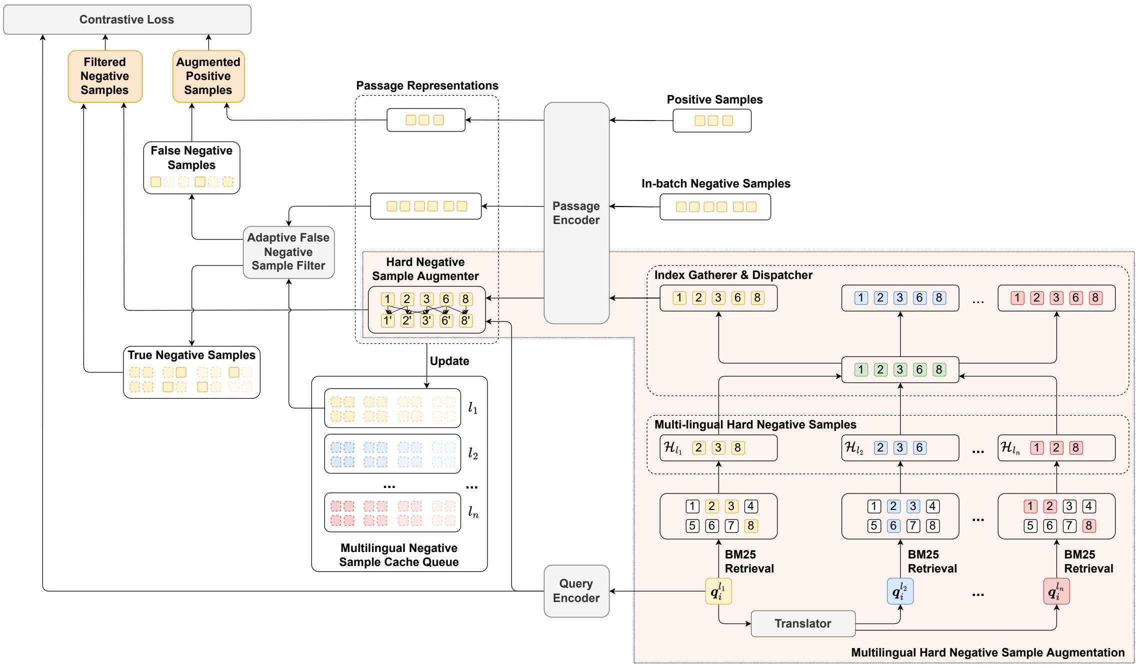

Contrastive Loss
Filtered Negative Samples
Augmented Positive Samples
Passage Representations
Positive Samples
False Negative Samples
In-batch Negative Samples
Passage Encoder
Adaptive False Negative Sample Filter
Hard Negative Sample Augmenter
Index Gatherer & Dispatcher
True Negative Samples
Update
Multi-lingual Hard Negative Samples
l_1
l_2
l_n
Multilingual Negative Sample Cache Queue
BM25 Retrieval
Query Encoder
$q_i^{l_1}$
$q_i^{l_2}$
$q_i^{l_n}$
Translator
Multilingual Hard Negative Sample Augmentation

Figure 2: The overall architecture of mHFN. Numbers, e.g., 1,2,...,8 in the diagram, indicate the index of a passage. The same indices in different colors (except the index gatherer) are the pseudo parallel passages. $\mathcal{H}_{l_1}, \mathcal{H}_{l_2}, \cdots, \mathcal{H}_{l_n}$ denote hard negative samples from language $l_1, l_2, \cdots, l_n$, respectively. For simplicity, we only show one single language in the left part of the diagram (i.e., except the translator, BM25 retrieval model, and index gatherer & dispatcher). The "Update" operation is to update the multilingual negative sample cache queue for the subsequent batch, so it is performed after calculating the current batch.

process of queries and that of passages, and the passage representations are precomputed in practice.

3.2 Multilingual Hard Negative Sample Augmentation

In order to incorporate hard negative samples into multilingual retrieval models, a simple solution is to treat each language independently. In this case, hard negative samples are computed and maintained in each language separately like a monolingual retrieval model. This method is used in mDPR (Zhang et al., 2021). However, if we consider hard negative samples as a knowledge source that indicates the most confused answers for the retrieval model, they shall be shared across multiple languages, which will make passage representations of low-resource languages more discriminative to differentiate between positive and negative samples with the help from high-resource languages.

To achieve this goal, we propose a multilingual hard negative sample augmentation module. This module has three components: **translator**, **index gatherer & dispatcher**, and **hard negative sample augmenter**, which collectively enable multiple languages to share hard negative "knowledge" so that low-resource languages can benefit from high-resource language data to improve the overall modeling capability.

As shown in Figure 2, we first use the translator (based on MarianNMT (Junczys-Dowmunt et al., 2018)) to translate the entire dataset from corresponding source language l_1 to other languages $l_2, \cdots, l_n$ to construct pseudo parallel corpora. Then, as a standard practice (Karpukhin et al., 2020; Zhang et al., 2021, 2022), we use BM25 (Robertson et al., 2009) to retrieve top-30 non-positive results as hard negative samples (denoted as $\mathcal{H}_l$) for each language. With these hard negative samples, for each query in the source language $\boldsymbol{q}^{l_1}$,[2] we combine all corresponding indices of negative samples for those pseudo parallel queries generated from the original query (index gatherer). Then we dispatch these indices to each language (index dispatcher). This will make the embedded hard negative knowledge shared across languages, especially for low-resource ones:

$$\mathcal{I}_s = \cup_{s=1}^{n} \{ j \mid \boldsymbol{p}_j^{l_s} \in \mathcal{H}_{l_s} \}, \tag{2}$$

$$\mathcal{H}'_{l_s} = \{ \boldsymbol{p}_j^{l_s} \mid j \in \mathcal{I}_s \} \tag{3}$$

[2] We omit l if the language l is not used in context for simplicity.

where $s = 1, 2, \cdots, n$. $\mathcal{H}'_{l_s}$ is the combined hard negative set, and $\boldsymbol{p}_j^{l_s}$ is the j-th negative passage in the hard negative sample set of language l_s.

Inspired by Kalantidis et al. (2020), we further linearly interpolate representations of queries and randomly chosen hard negative samples to provide synthetic hard negative samples (hard negative sample augmenter):

$$\tilde{\mathbf{h}}_{i,j} = \alpha E_q(\boldsymbol{q}) + \beta E_p(\boldsymbol{p}_i) + \gamma E_p(\boldsymbol{p}_j), \quad (4)$$

$$\mathbf{h}_{i,j} = \frac{\tilde{\mathbf{h}}_{i,j}}{\|\tilde{\mathbf{h}}_{i,j}\|_2} \quad (5)$$

where $\boldsymbol{q}$ is a query, and $\boldsymbol{p}_i, \boldsymbol{p}_j$ are two random hard negative samples in $\mathcal{H}'_l$. E_q and E_p denote query encoder and passage encoder respectively. $\mathbf{h}_{i,j}$ is the synthetic hard negative sample representation and will be added to the combined hard negative sample set $\mathcal{H}'_l$. $\alpha \in (0, 0.5), \beta \in (0, 1), \gamma \in (0, 1)$ are interpolation coefficients which satisfy $\alpha + \beta + \gamma = 1$. Note that following (Kalantidis et al., 2020), we set $\alpha \in (0, 0.5)$ to guarantee that the query's contribution is always smaller than those of hard negative samples.

3.3 Multilingual Negative Sample Cache Queue

Existing in-batch negative sampling method technique enables retrieval models to use other samples in the current mini-batch as negative samples to make full use of training data with little training cost (Karpukhin et al., 2020). In this method, the number of negative samples depends on the mini-batch size, which is bounded by GPU memory.

To further increase the number of negative samples from multiple languages under the multilingual setting, we propose a multilingual negative sample cache queue to help improve the encoded representations, as shown in Figure 2.

First, we build multiple batches where data in the same batch are always in the same language, which will be helpful to training because it prevents in-batch negative sampling from being degenerated into a language detection task, where the model tries to distinguish positive passages (in one language) from negative samples (in another language), as pointed by Zhang et al. (2022).

Then, we maintain multiple cache queues as external negative sample pools, where each language l has its own queue $\mathcal{Q}_l$. When we train the retrieval model with a batch $\mathcal{B}$ in language l, for each query $\boldsymbol{q}_i^l$ in $\mathcal{B}$, we filter out the in-batch negative samples along with the passage representations stored in $\mathcal{Q}_l$ via the false negative sample filter introduced in section 3.4 to obtain true negative samples. These true negative samples are combined with the original and synthetic hard negative samples for training. When the model completes training with the current batch $\mathcal{B}$ for language l, all passage representations (including both positive and negative) introduced will be added into $\mathcal{Q}_l$ for subsequent training with gradient disconnected. Considering the number of languages may be large, in order to keep multilingual scalability, we maintain these queues in RAM's pinned memory instead of GPU memory, which allows GPU devices to directly fetch them without CPU call. In our pilot experiment, this practice makes little compromise on the training speed since CUDA uses Direct Memory Access (DMA) to transfer pinned memory to GPU, and only one queue is loaded into GPU memory each time. This multilingual negative sample cache queue significantly enlarges the number of negative samples for multiple languages. The size of the cache queue for each language is only limited to a capacity threshold $\mathcal{C}$ which depends on the memory size and can be much larger than the batch size. If a cache queue is full, the earliest representations in the cache queue will be dequeued to achieve rolling updates.

3.4 Adaptive False Negative Sample Filter

In passage retrieval datasets, annotated positive passages only occupy a small portion, and a large number of actually positive passages are unlabeled and thus treated as negative ones (Qu et al., 2021). This becomes a problem for CL-based dense passage retrieval models because the models will falsely push apart the queries from these false negative samples, making training hard to converge. Current approaches in monolingual retrieval, such as RocketQA (Qu et al., 2021), still require a cross-encoder-based false negative sample filter to generate pseudo labels for unlabeled data and then filter out the false negative samples. However, cross-encoders are quite inefficient in both training and inference, especially for multilingual retrieval datasets at a potentially larger scale.

In order to efficiently filter out false negative samples, we propose an adaptive false negative sample filter. Particularly, for each query $\boldsymbol{q}_i$ in batch $\mathcal{B}$, we combine positive passages, in-batch negative samples and cached negative samples into a passage set $\mathcal{P}$, and use K-means to group these passages into K clusters. To avoid manual tuning for K, we use gap statistic (Tibshirani et al., 2001) to automatically determine the optimal K for data. We then use Gumbel-Softmax (Jang et al., 2017) to randomly assign pseudo labels, where the probability of each label $\mathcal{L}_t$ is the normalized similarity with each cluster center c_t:

$$P_{\boldsymbol{p}\in\mathcal{L}_t} = \frac{\mathrm{sim}(\boldsymbol{p},c_t)}{\sum_{j=1}^{K}\mathrm{sim}(\boldsymbol{p},c_j)}, \text{where } \boldsymbol{p} \in \mathcal{P}, \quad (6)$$

$$y_t = \frac{e^{(\log(P_{\boldsymbol{p}\in\mathcal{L}_t})+g_t)/\tau}}{\sum_{j=1}^{K} e^{(\log(P_{\boldsymbol{p}\in\mathcal{L}_t})+g_j)/\tau}}, \text{for } t=1,\ldots,K \quad (7)$$

where $g_1, \cdots, g_K \sim \mathrm{Gumbel}(0,1)$, and τ denotes the temperature hyper-parameter which controls the closeness between Gumbel-Softmax distribution and the categorical distribution. The probability of each pseudo label tends to be uniform as τ increases and become one-hot otherwise. We choose the index of the maximum y_t as the pseudo label for each passage. After assigning the pseudo labels, we treat the negative samples which have the same pseudo label as the positive passages as false negative passages, and exclude them from the negative passage set.

We further use τ to control the confidence of label assignment. Since passage representations are not optimized at the beginning of training, pseudo labels are relatively unreliable. So the τ should be higher to achieve more random label assignments. As training continues, passage representations are gradually optimized, so pseudo labels should be assigned with higher confidence (i.e., with a lower τ) at this moment. Thus, we apply a decaying schedule on τ to control the confidence by training steps:

$$\tau = \frac{\tau_0}{\text{\# Training steps}} \quad (8)$$

where τ_0 is relatively large to make Gumbel-Softmax distribution close to uniform distribution at early training steps.

	Train		Dev		Test		Corpus Size
	# Q	# J	# Q	# J	# Q	# J	
Arabic (Ar)	12,377	12,377	3,115	3,115	1,081	1,257	2,106,586
Bengali (Bn)	1,713	1,719	440	443	111	130	304,059
English (En)	3,547	3,547	878	878	744	935	32,907,100
Finnish (Fi)	6,561	6,561	1,738	1,738	1,254	1,451	1,908,757
Indonesian (Id)	4,902	4,902	1,224	1,224	829	961	1,469,399
Japanese (Ja)	3,697	3,697	928	928	720	923	7,000,027
Korean (Ko)	1,295	1,317	303	307	421	492	1,496,126
Russian (Ru)	5,366	5,366	1,375	1,375	995	1,168	9,597,504
Swahili (Sw)	2,072	2,401	526	623	670	743	136,689
Telugu (Te)	3,880	3,880	983	983	646	664	548,224
Thai (Th)	3,319	3,360	807	817	1,190	1,368	568,855
Total	48,729	49,127	12,317	12,431	8,661	10,092	58,043,326

Table 1: The detailed statistics for the Mr. TyDi dataset: the number of questions (# Q), labeled positive passages (# J), and the total number of passages (Corpus Size) in each language.

3.5 Loss Function

We use the NCE loss to optimize the mHFN model. Since false negative samples are actually positive, we can treat them as positive passages when calculating the loss function to achieve better utilization:

$$\begin{aligned} L = &-\sum_{\boldsymbol{p}_i^+\in\mathcal{L}_i} \log \frac{e^{\mathrm{sim}(\boldsymbol{q}_i,\boldsymbol{p}_i^+)}}{e^{\mathrm{sim}(\boldsymbol{q}_i,\boldsymbol{p}_i^+)} + \sum_{\boldsymbol{p}_i^-\notin\mathcal{L}_i} e^{\mathrm{sim}(\boldsymbol{q}_i,\boldsymbol{p}_i^-)}} \\ &-\sum_{\boldsymbol{p}_i^-\in\mathcal{L}_i} \log \frac{e^{\mathrm{sim}(\boldsymbol{q}_i,\boldsymbol{p}_i^-)}}{e^{\mathrm{sim}(\boldsymbol{q}_i,\boldsymbol{p}_i^-)} + \sum_{\boldsymbol{p}_i^-\notin\mathcal{L}_i} e^{\mathrm{sim}(\boldsymbol{q}_i,\boldsymbol{p}_i^-)}} \end{aligned} \quad (9)$$

where $\boldsymbol{q}_i$, $\boldsymbol{p}_i^+$ and $\boldsymbol{p}_i^-$ indicate the query, corresponding positive and negative passages respectively. $\mathcal{L}_i$ is the pseudo label of the corresponding positive passages for $\boldsymbol{q}_i$.

The confidence is relatively low at the beginning of training. Hence, we only use false negative samples as positive passages in the loss function (i.e., the second term in the loss function) when the confidence becomes relatively higher after 20% of training steps.

4 Experiments

4.1 Dataset and Evaluation Metrics

We chose Mr. TyDi (Zhang et al., 2021) to evaluate our proposed model. Mr. TyDi (Zhang et al., 2021) is a multilingual retrieval benchmark dataset constructed from TyDi QA (Clark et al., 2020), a question answering dataset covering eleven typologically diverse languages. Given a query, the goal of Mr. TyDi is to find relevant passages in a pool of Wikipedia passages in the same language. The detailed statistics of the dataset are shown in Table 1, which are copied from the original Mr. TyDi paper. Following the original setting in Mr. TyDi, we report MRR@100 and Recall@100 on the test set of each language.

	Ar	**Bn**	**En**	**Fi**	**Id**	**Ja**	**Ko**	**Ru**	**Sw**	**Te**	**Th**	**Avg.**
BM25 (default)	0.368	0.418	0.140	0.284	0.376	0.211	0.285	0.313	0.389	0.343	0.401	0.321
BM25 (tuned)	0.367	0.413	0.151	0.288	0.382	0.217	0.281	0.329	0.396	0.424	0.417	0.333
mDPR	0.291	0.291	0.291	0.206	0.271	0.213	0.235	0.283	0.189	0.111	0.172	0.226
mDPR (hybrid)	0.491	0.535	0.284	0.365	0.455	0.355	0.362	0.427	0.405	0.420	0.492	0.417
mDPR (FT)	0.695	0.659	0.476	0.550	0.565	0.496	0.453	0.515	0.633	0.891	0.607	0.594
mHFN	0.746	0.739	0.572	0.643	0.657	0.564	0.568	0.589	0.715	**0.919**	0.689	0.673
- index gatherer & dispatcher	0.735	0.705	0.551	0.625	0.633	0.545	0.531	0.571	0.689	0.901	0.667	0.650
- hard negative sample augmenter	0.723	0.717	0.535	0.630	0.640	0.532	0.542	0.573	0.701	0.902	0.679	0.652
- (index gatherer & dispatcher + hard negative sample augmenter)	0.715	0.701	0.519	0.618	0.625	0.510	0.517	0.554	0.681	0.895	0.660	0.636
- multilingual negative sample cache queue	0.718	0.705	0.525	0.622	0.640	0.515	0.538	0.569	0.705	0.907	0.671	0.647
- adaptive false negative sample filter	0.725	0.709	0.514	0.619	0.621	0.526	0.532	0.565	0.699	0.892	0.669	0.643
mHFN (hybrid)	**0.802**	**0.771**	**0.624**	**0.715**	**0.701**	**0.574**	**0.605**	**0.615**	**0.734**	0.897	**0.811**	**0.714**

(a) MRR@100 on the test set

	Ar	**Bn**	**En**	**Fi**	**Id**	**Ja**	**Ko**	**Ru**	**Sw**	**Te**	**Th**	**Avg.**
BM25 (default)	0.793	0.869	0.537	0.719	0.843	0.645	0.619	0.648	0.764	0.758	0.853	0.732
BM25 (tuned)	0.800	0.874	0.551	0.725	0.846	0.656	0.797	0.660	0.764	0.813	0.853	0.758
mDPR	0.650	0.779	0.678	0.568	0.685	0.584	0.533	0.647	0.528	0.366	0.515	0.594
mDPR (hybrid)	0.863	0.937	0.696	0.788	0.887	0.778	0.706	0.760	0.786	0.827	0.875	0.809
mDPR (FT)	0.894	0.937	0.839	0.846	0.867	0.811	0.771	0.819	0.893	0.969	0.866	0.865
mHFN	0.941	**0.953**	0.879	0.912	0.952	0.895	0.909	0.913	0.935	**0.981**	0.955	0.930
- index gatherer & dispatcher	0.932	0.905	0.850	0.882	0.920	0.874	0.841	0.891	0.897	0.963	0.929	0.899
- hard negative sample augmenter	0.912	0.924	0.825	0.901	0.923	0.842	0.868	0.890	0.921	0.968	0.935	0.901
- (index gatherer & dispatcher + hard negative sample augmenter)	0.902	0.901	0.803	0.871	0.882	0.812	0.830	0.854	0.879	0.949	0.906	0.872
- multilingual negative sample cache queue	0.910	0.905	0.807	0.888	0.931	0.809	0.855	0.876	0.922	0.970	0.931	0.891
- adaptive false negative sample filter	0.915	0.909	0.801	0.896	0.903	0.850	0.867	0.872	0.908	0.941	0.925	0.890
mHFN (hybrid)	**0.952**	0.951	**0.914**	**0.923**	**0.962**	**0.911**	**0.915**	**0.927**	**0.936**	**0.981**	**0.956**	**0.939**

(b) Recall@100 on the test set

Table 2: Multilingual retrieval results across all 11 languages on the Mr. TyDi dataset.

4.2 Baselines

In the original Mr. TyDi benchmark (Zhang et al., 2021), there are three types of baselines: sparse, dense, and hybrid. The sparse baselines are implemented with Pyserini (Lin et al., 2021) using BM25 (Robertson et al., 2009), along with a "tuned" BM25 baseline which optimizes MRR@100 by tuning the default BM25 parameters k_1 and b on the development set. We denote these two baselines as **BM25 (default)** and **BM25 (tuned)** respectively. The dense baseline is **mDPR** (Zhang et al., 2021), which is a neural model fine-tuned on the NaturalQuestions dataset (Kwiatkowski et al., 2019) using the DPR (Karpukhin et al., 2020) pipeline and mBERT encoder. The hybrid baseline is a combination of the dense baseline and the tuned BM25 sparse baseline, which is denoted by **mDPR (hybrid)**. The final fusion score of the hybrid baseline is calculated by $s_{sparse} + \alpha \cdot s_{dense}$, where s_{sparse} and s_{dense} represent the normalized scores from the sparse and dense retrieval baseline respectively. The hyper-parameter α is also tuned on the development set by optimizing MRR@100. Following the above setting, we also report the performance of our hybrid model, which uses the same fusion strategy as the hybrid baseline on our model and the tuned BM25 baseline.

Considering that the dense baseline is under zero-shot setting (i.e., not trained on Mr. TyDi), for a fair comparison, we report the result of mDPR fine-tuned on Mr. TyDi training set from Zhang et al. (2022) as an additional baseline. We denote this model as **mDPR (FT)**. The implementation details are shown in Appendix 4.3.

4.3 Implementation Details

Our models were built upon Tevatron (Gao et al., 2022), a lightweight and efficient dense passage retrieval toolkit. Following Zhang et al. (2022), we use the same built-in procedure in Tevatron to conduct preprocessing on the Mr. TyDi dataset. The query encoder and passage encoder were initialized using mBERT-base (110M parameters) with shared parameters. We used AdamW optimizer for optimization, and the learning rate was set to 5e-5. The initial temperature hyper-parameter τ_0 was set to 100. The multilingual negative sample cache queue had a default maximum capacity M_l of 20k for each language l.

In all experiments, we trained the model with 2k steps on 4 NVIDIA A6000 48GB GPUs. The batch size was set to 256 for each GPU. All experimental results reported were averaged over 5 runs with different random seeds.

4.4 Main Results

We show the main results of mHFN on the Mr. TyDi dataset in Table 2. As can be seen, mHFN surpasses all baseline models by a large margin and achieves state-of-the-art performance for all languages on the Mr. TyDi dataset. Specifically, the sole mHFN model outperforms over tuned BM25, mDPR, mDPR (hybrid) and mDPR (FT) by 38.1%/18.1%, 44.7%/33.6%, 25.6%/12.1% and 7.9%/6.5% in average MRR@100/Recall@100, respectively. The hybrid model of mHFN and BM25 takes a step further to improve the average MRR@100 and Recall@100 over all languages by 4.1% and 0.9% compared to the sole mHFN model.

5 Analysis

5.1 Ablation Study

In order to further analyze the effectiveness of each component, we conduct three ablation studies to quantify the contribution of various factors: the multilingual hard negative sample augmentation, multilingual negative sample cache queue, and adaptive false negative sample filter.

Effect of the multilingual hard sample negative augmentation To demonstrate the effectiveness of the multilingual hard negative sample augmentation module, we show the results of mHFN without the index gatherer & dispatcher, hard negative sample augmenter, and both. As Table 2 shows, the removal of the multilingual hard negative sample augmentation module (including both the index gatherer & dispatcher and hard negative sample augmenter) leads to a significant performance drop of average 3.7% MRR@100 and 5.8% Recall@100 on all languages. Specifically, the removal of the index gatherer & dispatcher results in a drop of average 2.3% and 3.1% in MRR@100 and Recall@100 in all languages respectively. Especially, the three languages which have the smallest amount of training data (Korean, Bengali, and Swahili) suffer from the largest performance drop of 3.7%/6.8%, 3.4%/4.8%, and 2.6%/3.8% in MRR@100/Recall@100. These results suggest that the index gatherer & dispatcher can share knowledge of hard negative samples across languages, which is beneficial to mHFN, especially for low-resource languages. We also observe a similar performance drop for the hard negative sample augmenter. This is because the hard negative sample augmenter can dynamically synthesize new hard negative samples based on existing static (i.e. generated before the actual training phase) BM25 hard negative samples during training, which enriches the diversity of negative samples and thus benefits the model.

Effect of the multilingual negative sample cache queue We also conducted an ablation study on the multilingual negative sample cache queue to demonstrate the effectiveness of incorporating more negative samples into multilingual dense passage retrieval. As can be seen in Table 2, the performance of mHFN with the multilingual negative sample cache queue is better than that without it. Specifically, we observe an average 2.6% and 3.9% drop in MRR@100 and Recall@100 respectively in all languages. These results indicate that the multilingual negative sample cache queue can increase the number of negative samples for each language, which leads to a better performance of mHFN. We further analyzed the impact of the size of the cache queue on performance. The experimental results show that the performance can be improved with a larger cache queue, especially with the adaptive false negative sample filter.

Effect of the false negative sample filter We conducted another ablation study to demonstrate the effectiveness of the adaptive false negative sample filter. As shown in Table 2, its absence leads to a significant performance drop of average 3.0% MRR@100 and 4.0% Recall@100 in all languages. We speculate that it is likely to bring noise that will falsely guide the model to distinguish unlabeled positive samples from labeled positive samples if we simply use in-batch and cached negative samples. As a comparison, we propose an adaptive negative sample filter to dynamically filter out false negative samples and use them as positive passages to further improve the multilingual dense passage retrieval, and results show that the false negative sample filter can further improve the performance based on the multilingual negative sample cache queue with quality-enhanced negative passages.

5.2 Case Study And Visualization

We provide two examples of mHFN vs. mDPR in Table 3.

In the first example, both mDPR and mHFN exclude hard negative samples from the top 100 retrieval results. The example-relevant false negative sample is excluded by mDPR from the top 100 retrieval results, while mHFN ranks the relevant false negative sample in the second place. This is because the valuable false negative samples can be discovered and filtered out by mHFN, which will

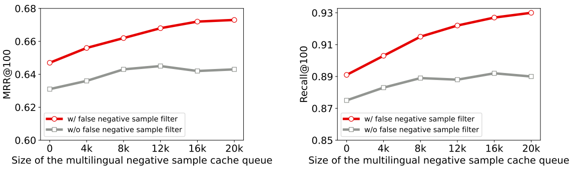
Queries	Labeled Positive Passages	Unlabeled False Negative Passages	HN Statistics	
4565: **How long does a basketball game last?**	3921#31 **(Ranking 33rd in mDPR, 2st in mHFN):** . . . Games are played in four quarters of 10 (FIBA)[35] or 12 minutes (NBA).[36] College men's games use two 20-minute halves,[37] college women's games use 10-minute quarters . . .	768856#26 **(Ranking 564th in mDPR, 1st in mHFN):** . . . NCAA men's games are divided into two halves, each 20 minutes long; NBA games are played in four quarters of 12 minutes each; and WNBA and NCAA women's games are played in 10-minute quarters . . .	**Number of HN** 0 (mDPR) 0 (mHFN)	**Average Rank of HN** N/A (mDPR) N/A (mHFN)
5093: **What was the longest dynasty in China's history?**	5760#17 **(Ranking 43rd in mDPR, 1st in mHFN):** . . . The Zhou dynasty (1046 BC to approximately 256 BC) is the longest-lasting dynasty in Chinese history . . .	43464#0 **(Ranking 3rd in mDPR, 2nd in mHFN):** . . . The Zhou dynasty lasted longer than any other dynasty in Chinese history . . .	**Number of HN** 3 (mDPR) 1 (mHFN)	**Average Rank of HN** 20.67 (mDPR) 87 (mHFN)

Table 3: Examples of mHFN vs. mDPR on the Mr. TyDi dataset (in English). HN denotes the top 30 hard negative passages generated by BM25. The average rank and number of these hard negative passages are counted from the top 100 retrieval results.

then be used as positive samples to achieve better retrieval.

In the second example, mHFN also gives a much higher ranking to the labeled positive answer. Especially, the average rank of hard negative samples (20.67) is even higher than the labeled positive answer (43). The top 100 retrieval results of mDPR contain more hard negative samples than mHFN. This is because our multilingual hard negative sample augmentation module can share the knowledge of indistinguishable passages across languages, and then synthesize various hard negative samples to further improve the discrimination between positive and negative samples.

The visualization of the earned representations of queries, hard negative samples, and false negative samples of mHFN vs. mDPR is shown in Figure 1. It is clear that our model is capable of separating hard and false negative samples from ordinary negative samples.



(a) MRR@100

(b) Recall@100

Figure 3: The effect of the size of the multilingual negative sample cache queue. Average MRR@100 (a) and Recall@100 (b) over all languages on the Mr. TyDi dataset are plotted.

5.3 Analysis on the Size of the Multilingual Negative Sample Cache Queue

An important variable in our model is the size of the multilingual negative sample cache queue M_l. We conducted experiments to investigate its effect. We show the results of mHFN with different lengths of cache queue in Figure 3. It can be seen that the average MRR@100 and Recall@100 over all languages can be improved with increasing queue size (i.e., an increasing number of negative samples). However, the overall performance basically remains unchanged when the size > 8k if the false negative sample filter is not used, whereas the performance is stably improving along the increasing queue size if the false negative sample filter is present. We conjecture that as the number of negative samples increases, more false negative cases are also introduced, which may hurt training. The false negative sample filter can be considered as a purifier that separates the adaptive false negative samples from all candidate negative samples and thus enhances the overall quality of negative samples.

6 Conclusion

In this paper, we have presented a novel multilingual dense passage retrieval model, mHFN, which efficiently explores hard and false negative samples. It can 1) efficiently share hard negative samples across languages and generate augmented high-quality hard negative samples, 2) increase the number of multilingual negative samples, and 3) adaptively filter out unlabeled false negative samples from all candidate negative samples for effective training.

Experiments and in-depth analysis validate the effectiveness of mHFN and demonstrate that it outperforms the strong sparse, dense, and hybrid baselines, setting new state-of-the-art results on the Mr. TyDi dataset.

Acknowledgements

This work was partially supported by Zhejiang Lab (No. 2022KH0AB01). We would like to thank the anonymous reviewers for their insightful comments.

Limitations

Although we propose a multilingual negative sample cache queue to mitigate the limit of GPU memory, several powerful GPUs are still necessary to speed up training. Additionally, as the corpus is much larger than the training data, it is also time-consuming to encode the entire corpus in the evaluation phase, which makes real-time evaluation much more difficult with a huge corpus for most pretrained model-based dense passage retrieval models. For example, it takes us more than 10 hours to evaluate a checkpoint on a single NVIDIA A6000 GPU for all languages in the Mr. TyDi dataset. There are some possible solutions, like sampling a random subset of the corpus for approximate evaluations, or evaluating asynchronously with a specialized validation toolkit such as Asyncval[3] (Zhuang and Zuccon, 2022). We leave this issue to our future work.

Ethics Statement

mHFN is trained on the Mr. TyDi dataset, which is originated from Wikipedia. Since Wikipedia can be edited by anyone, it may contain inappropriate content at the time of dataset construction. Therefore, we advise users to carefully examine the ethical implications of the retrieved results and to apply our retrieval model with caution in real-world scenarios. However, since there is a highly active community of volunteers to inspect the content and ensure the overall correctness, appropriateness, and quality of Wikipedia, our mHFN has lower ethical risk compared to the dense passage retrieval models trained with uncensored web-crawled data. Also, due to the characteristics of the dense passage retrieval task, the ethical risk of mHFN is also lower than generative auto-regressive language models (Bender et al., 2021). Meanwhile, mHFN achieves high-quality dense passage retrieval for eleven typologically diverse languages, which contributes to the equality and diversity of language technology.

References

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021a. XOR QA: cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online. Association for Computational Linguistics.

Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. 2021b. One question answering model for many languages with cross-lingual dense passage retrieval. *Advances in Neural Information Processing Systems*, 34:7547–7560.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: a benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2020. Modularized transfomer-based ranking framework. In *EMNLP*, pages 4180–4190. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. COIL: revisit exact lexical match in information retrieval with contextualized inverted list. In *NAACL-HLT*, pages 3030–3042. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021b. Complement lexical retrieval model with semantic residual embeddings. In *European Conference on Information Retrieval*, pages 146–160. Springer.

[3] https://github.com/ielab/asyncval

Luyu Gao, Xueguang Ma, Jimmy J. Lin, and Jamie Callan. 2022. Tevatron: an efficient and flexible toolkit for dense retrieval. *arXiv:2203.05765*.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021c. Scaling deep contrastive learning batch size under memory limited setup. In *The 6th Workshop on Representation Learning for NLP (RepL4NLP)*, pages 316–321.

John M. Giorgi, Osvald Nitski, Bo Wang, and Gary D. Bader. 2021. Declutr: deep contrastive learning for unsupervised textual representations. In *ACL-IJCNLP*, pages 879–895. Association for Computational Linguistics.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

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 *Proc. of SIGIR*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809.

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)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 39–48, New York, NY, USA. Association for Computing Machinery.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. 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)*, pages 2356–2362.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: a robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020a. Efficient document re-ranking for transformers by precomputing term representations. In *SIGIR*, pages 49–58. ACM.

Sean MacAvaney, Luca Soldaini, and Nazli Goharian. 2020b. Teaching a new dog old tricks: resurrecting multilingual retrieval using zero-shot learning. In *European Conference on Information Retrieval*, pages 246–254. Springer.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: an optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Peng Shi, He Bai, and Jimmy Lin. 2020. Cross-lingual training of neural models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2768–2773.

Shuo Sun and Kevin Duh. 2020. CLIRMatrix: a massively large collection of bilingual and multilingual datasets for cross-lingual information retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4160–4170, Online. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. LaPraDoR: unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3557–3569, Dublin, Ireland. Association for Computational Linguistics.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512, Virtual Event Canada. ACM.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. TyDi: a multi-lingual benchmark for dense retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, and Jimmy Lin. 2022. Towards best practices for training multilingual dense retrieval models. *arXiv:2204.02363*.

Shengyao Zhuang and Guido Zuccon. 2022. Asyncval: a toolkit for asynchronously validating dense retriever checkpoints during training. *arXiv:2202.12510*.