Toward Best Practices for Training Multilingual Dense Retrieval Models

XINYU ZHANG, KELECHI OGUEJI, XUEGUANG MA, and JIMMY LIN, University of Waterloo, Canada

Dense retrieval models using a transformer-based bi-encoder architecture have emerged as an active area of research. In this article, we focus on the task of monolingual retrieval in a variety of typologically diverse languages using such an architecture. Although recent work with multilingual transformers demonstrates that they exhibit strong cross-lingual generalization capabilities, there remain many open research questions, which we tackle here. Our study is organized as a “best practices” guide for training multilingual dense retrieval models, broken down into three main scenarios: when a multilingual transformer is available, but training data in the form of relevance judgments are not available in the language and domain of interest (“have model, no data”); when both models and training data are available (“have model and data”); and when training data are available but not models (“have data, no model”). In considering these scenarios, we gain a better understanding of the role of multi-stage fine-tuning, the strength of cross-lingual transfer under various conditions, the usefulness of out-of-language data, and the advantages of multilingual vs. monolingual transformers. Our recommendations offer a guide for practitioners building search applications, particularly for low-resource languages, and while our work leaves open a number of research questions, we provide a solid foundation for future work.

CCS Concepts: • Information systems → Structure and multilingual text search; Top-k retrieval in databases; Information retrieval diversity;

Additional Key Words and Phrases: Dense retrieval, multilingual retrieval

ACM Reference format:
Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, and Jimmy Lin. 2023. Toward Best Practices for Training Multilingual Dense Retrieval Models. ACM Trans. Inf. Syst. 42, 2, Article 39 (September 2023), 33 pages. https://doi.org/10.1145/3613447

1 INTRODUCTION

Retrieval based on dense vector representations derived from pretrained transformers such as BERT [12] represents an active area of research. By adopting a so-called bi-encoder (or dual-encoder) architecture, document representations can be computed offline and document ranking can be recast as a nearest-neighbor search problem given the query vector representation, for which existing open-source toolkits such as Faiss [24] or nmslib [8] provide efficient and scalable solutions. A bi-encoder architecture provides an attractive alternative to the so-called...
cross-encoder architecture, whereby queries and documents are concatenated and fed into a transformer directly. Cross-encoders are only practical in a reranking setup, processing candidates generated by a first-stage retrieval system, whereas bi-encoders directly support single-stage retrieval (but, of course, can be further reranked by cross-encoders if desired).

Our work explores dense retrieval models based on bi-encoders in a multilingual context. To be clear, we are concerned with monolingual retrieval, particularly in non-English languages, where both queries and documents are in the same language $L$ (which we call the target language). While many of our techniques are also applicable to the cross-lingual case, where queries and documents are in different languages, we do not explicitly consider this application. Recent work has shown that retrieval models (bi-encoders and cross-encoders) based on multilingual transformers exhibit strong cross-lingual generalization capabilities. That is, we can train the model using relevance judgments in one language and apply inference in another language for ranking [4, 7, 38, 53, 65]. It appears that models are able to "transfer" relevance matching capabilities across languages. This is an exciting finding, because it becomes possible to leverage relevance judgments in one language (for example, a high-resource language such as English) to build retrieval models for low-resource languages (e.g., Telugu).

Of course, the reality is quite complex, and there remain many open questions. Suppose we are interested in building a dense retrieval model for target language $L$. What should we do if we do not have any in-domain training data in the target language (by "in-domain" we mean training data that are drawn from the same distribution as the evaluation data)? Should we rely entirely on cross-lingual transfer from another out-of-domain dataset such as MS MARCO in English (by “out-of-domain” we mean training data that are not drawn from the same distribution as the evaluation data)? How can we exploit versions of MS MARCO that have been automatically translated into different languages, including the target language? Alternatively, if translations are not available, then can we exploit unlabeled data in the target language, and how does that compare to using machine translated training data? What if we do have (perhaps, limited) in-domain training data in the target language? Is a large out-of-domain dataset such as MS MARCO (in English or automatically translated into other languages) still beneficial?

More open questions are as follows: Transformers can be fine-tuned in different "stages" [18, 59, 67], so it is certainly possible to exploit datasets in different languages at different stages when building from a multilingual transformer. Should we? Does the answer change if the datasets draw from linguistically unrelated families or are written in entirely different scripts? Are the cross-lingual transfer effects more prominent at one stage compared to the others? What about using different pretrained transformers as starting points? And to consider an entirely different approach, should we prefer monolingual transformers if one is available?

The goal of our work is to answer these myriad questions and to begin to develop a set of best practices for training multilingual dense retrieval models. Our exploration is organized into three main scenarios as follows:

1. **Have model, no data** — scenario (1): The target language $L$ is covered by a pretrained multilingual transformer, but unfortunately we do not have any in-domain training data in the target language.

2. **Have model and data** — scenario (2): The target language $L$ is covered by a pretrained multilingual transformer and we have in-domain training data (i.e., manually labeled relevance judgments). This breaks down into two sub-cases, one where we have in-domain training data in the target language $L$ and the other where we have in-domain training data in some other (non-target) language $K$. We also consider the case where a monolingual transformer for language $L$ is available.
(3) *Have data, no model* — scenario (3): No pretrained transformer is available for the target language $L$, but we do have in-domain training data in the target language.

This work builds on **Dense Passage Retriever (DPR)** [26], one of the earliest and most foundational dense retrieval models in the literature. Our experiments adopt the original DPR design as a starting point, but we also explore better pretrained models as alternative starting points (known as “backbones”) and exploit more sophisticated training regimes based on adapters, knowledge distillation, and hard negative mining.

We view our contributions as follows:

1. We provide concrete guidance and recommend best practices, supported by empirical results, on how to train multilingual DPR models for monolingual retrieval under different scenarios, as outlined above.

2. Our study answers a number of scientific questions about multilingual dense retrieval models that until now have not been clearly addressed, such as those enumerated above. For example, we find that “pre–fine-tuning” with the MS MARCO passage ranking dataset (in English) rarely hurts effectiveness, even if the target language is unrelated to English. However, while pre–fine-tuning in multiple languages is helpful in a zero-shot setting, it can hurt effectiveness when in-domain training data are available.

3. Many of our results are surprising, such as the fact that multilingual BERT, fine-tuned on *Thai* relevance judgments, is reasonably effective for retrieval in *Arabic*, and that monolingual *English* BERT, fine-tuned on *Russian* relevance judgments, is effective for retrieval in *Russian*. We begin to untangle why in terms of cross-lingual anchors and token overlap across corpora in different languages.

In summary, we provide practitioners with a “how to” guide for building multilingual search applications and researchers with a solid foundation for future work in multilingual dense retrieval.

## 2 BACKGROUND AND RELATED WORK

### 2.1 Multilingual Pretrained Transformers

Following the success of pretrained transformers in English NLP tasks [12, 35, 48], researchers began to investigate if pretrained models can incorporate information from multiple languages at the same time. These exploration started with mBERT [12], where the model was pretrained on Wikipedia from 104 languages, following the same model configuration and pretraining paradigm as BERT [12]. Later, XLM-R was proposed, which is based on RoBERTa [35] and pretrained on the CommonCrawl dataset processed by CCNet. The XLM-R team discovered that trying to condense too many languages into a pretrained transformer model can degrade overall effectiveness across all languages, which they then proposed to alleviate by increasing the model size. Recently, mContriever [23] exploited additional pretraining using contrastive learning on synthetic training data built using a technique known as **independent cropping (IC)**. The researchers found that the technique augments general-purpose pretrained transformers in terms of zero-shot transfer capabilities and in-domain effectiveness in both English and multilingual IR tasks.

The multilingual models discussed above are all based on encoder-only transformer architectures. While there are additional studies examining multilingual pretrained models based on the encoder–decoder transformer architecture [40, 61, 62], we do not discuss them here as the models are not used in this work.

### 2.2 Multilingual Dense Passage Retriever

We base our retrieval models on DPR [26], a foundational and one of the earliest dense retrieval models. Given a question $q$ and a passage $p$, DPR generates representations, $E_Q(q)$ and $E_P(p)$, independently using question and passage encoders based on BERT by taking the representation...
of the [CLS] token in the final layer. Similarity between a query and a passage is measured by the inner product of their representations as follows:

\[ s_{q,p} \overset{\Delta}{=} \text{sim}(q,p) = E_Q(q)^T E_P(p), \]

which is optimized according to the NCE loss,

\[ L(q_i, p^+_i, p^-_{i,1}, \ldots, p^-_{i,n}) = -\log \left( \frac{\exp(s_{q_i, p^+_i})}{\exp(s_{q_i, p^+_i}) + \sum_{j=1}^{n} \exp(s_{q_i, p^-_{i,j}})} \right), \]

where \( q_i \) is the question representation, \( p^+_i \) is the representation of a positive (relevant) passage, and \( p^-_{i,j} \) is the representation of a negative (non-relevant) passage. The similarity scores \( s_{q,p} \) are used to generate a ranking for query \( q \) using a nearest-neighbor vector search library (in our experiments, Faiss). Although we select DPR for our experiments, any alternative dense retrieval model such as ANCE [60], TCT-ColBERT [33], or TAS-B [22] can be substituted; the exact choice is unlikely to affect our overall findings.

DPR encoders were originally designed to be initialized with English BERT [26]. Following previous work [4, 65], for retrieval in multiple languages and to exploit cross-lingual transfer, we instead initialize the model with multilingual pretrained transformers (e.g., mBERT, XLM-R Large, mContriever), keeping all other aspects of the training identical. We call this "mDPR."

Note that throughout this article we refer to mDPR generically as a shorthand for multilingual dense retrieval models based on a bi-encoder architecture, even though we explore additional techniques that were not present in the original paper, e.g., hard negative mining. Following standard parlance, we refer to the model used to initialize the training as the "backbone" (typically from a publicly available model checkpoint). For example, mDPR can be built on an mBERT backbone, but other alternatives include XLM-R Large and mContriever. Indeed, we experiment with a number of different backbones in this work to explore their impact on effectiveness.

**mDDR and mDDR**. In both NLP and IR, additional MLM pretraining on the target corpus has been known to be effective for adapting pretrained transformers to a different domain or language [19, 28, 43, 63]. In this work, we experiment with DDR [63], a specific technique for efficiently applying additional MLM pretraining to dense retrieval.

DDR [63] was proposed in the context of out-of-domain English retrieval based on “adapter” modules, which allow efficient parameter sharing among models for different languages and domains [6, 46, 63]. The approach disentangles dense retrieval models into a Relevance Estimation Module (REM) and Domain Adaption Modules (DAMS), where the REM is delegated to the adapter and the DAMs are delegated to the backbones. Specifically, DDR additionally pretrains a backbone from BERT on each target domain corpus, obtaining a new domain-adapted backbone. In parallel to the additional pretraining, an adapter is fine-tuned with the original backbone (e.g., BERT), where the backbone is frozen and only the adapter is trained. The trained adapter hopefully captures general-purpose relevance estimation signals. Finally, at inference time, DDR is assembled from the adapter and the domain-adapted backbone. We refer readers to the original work for more details.

We refer to our approach as mDDR in the rest of this article, which applies DDR in the multilingual scenario: The adapter is trained on MS MARCO along with mBERT, and the language-adapted backbones are additionally pretrained on the Mr. TYDI corpus (used in our experiments, see Section 3.2) in each language, starting from mBERT as well. That is, applying mDDR on Mr. TYDI yields 11 backbones (one per language) and one general-purpose adapter.

Note that the training of the backbones and the adapters proceeds in parallel in the original DDR pipeline, thus there is no guarantee that the weights of the two align after assembly. To address
Fig. 1. The pipeline of (a) mDPR init. mBERT, (b) mDPR init. mContriever, and (c) mDDR and mDDR\(^+\), where blocks with dashed outlines indicate modules that are frozen during training. (PT: pretrain; FT: fine-tune).

this issue, we further fine-tune the adapter with the domain-adapted backbone rather than the original one, referred to as mDDR\(^+\). At this stage, the 11 backbones are identical to the ones in mDDR, but now we have 11 adapters, each aligned to a different language backbone.

The pipelines of mDDR and mDDR\(^+\) are illustrated in Figure 1, along with the pipelines of mDPR with different backbones for comparison.

### 2.3 Knowledge Distillation

Other than achieving multilingual transfer capabilities via directly fine-tuning the dense retrieval models themselves, previous research has shown that dense retrieval models distilled from more powerful teacher models (usually cross-encoders) yield better in-domain and out-of-domain effectiveness with the same training data. These findings mostly lie in the context of English retrieval [21, 22], which not only benefits DPR but also applies to other bi-encoder designs [14, 15, 30, 51]. For non-English retrieval, Lin et al. [31] demonstrated that SPLADE++ [15], a sparse bi-encoder model augmented with knowledge distillation, achieves competitive scores on retrieval in Persian and Russian.

### 2.4 Cross-lingual Retrieval

There is a large body of literature on cross-lingual retrieval dating back decades. This is not our focus, and thus we do not provide a comprehensive review. Instead, we refer readers to Nie [41] for a survey of pre-neural techniques and Galuščaková et al. [16] for a more recent review.

While multilingual pretrained transformers provide a convenient starting point for exploring cross-lingual transfer in information retrieval, the study of this phenomenon predates the emergence of mBERT. Zhang et al. [64] found that aligning the query and documents of a source and target language can improve transferability in the cross-lingual word embedding (CLWE) setting. More recently, Litschko et al. [34] compared the effectiveness of CLWE models and mBERT under an unsupervised scenario in the cross-lingual setting.

Since the emergence of multilingual pretrained transformers, researchers have been exploiting cross-lingual transfer to improve monolingual retrieval. MacAvaney et al. [38] and Shi et al. [52] both explored the effectiveness of zero-shot transfer, where mBERT is fine-tuned on a source language and directly applied to a target language. However, these papers exploited cross-encoders, as opposed to the bi-encoders that we explore here. While cross-encoders can be more effective, they are often much slower, since they operate in a retrieve-and-rerank setup. In contrast, the
bi-encoder design we explore is amenable to single-stage retrieval with nearest-neighbor vector search frameworks such as Faiss. Asai et al. [4] also made use of mBERT for retrieval in a many-to-many-language scenario. They trained a QA model that can answer a query in any language by retrieving evidence from a multilingual corpus and generating an answer in the query language. This differs from our setting where we are strictly focused on retrieving relevant documents from a monolingual corpus given a query in the same language.

2.5 Why Do Multilingual Pretrained Transformers Work?

There has been much recent work on trying to understand the impressive cross-lingual abilities of mBERT, with many conflicting results. Among the earliest demonstration was the work of Wu and Dredze [58], who evaluated mBERT in a zero-shot cross-lingual transfer scenario on five NLP tasks across 39 languages and found that mBERT achieves impressive effectiveness. They argued that shared subwords aid cross-lingual transfer based on a strong correlation between the percentage of overlapping subwords and transfer effectiveness. Pires et al. [47] also reported strong cross-lingual transfer abilities using mBERT, highlighting the model’s effectiveness in transferring across language scripts and code-switched text. The authors hypothesized that common subwords such as numbers and URLs are responsible for the model’s ability to generalize cross-linguistically, since these are identical across languages and thus are “forced” to share representations. Dufter and Schütze [13] echoed this finding, claiming that shared position embeddings and shared special tokens are necessary ingredients for cross-lingual capabilities.

However, there have been contradictory results as well. K et al. [25] explored the contribution of different components in mBERT to its cross-lingual capabilities and concluded that token overlap between languages does not play a significant role. They argued that other parts of the architecture, such as the depth of the network, are more critical for creating a multilingual model. Artetxe et al. [2] also showed that neither having a shared vocabulary nor joint pretraining is needed to obtain cross-lingual transfer. In fact, the authors were able to induce multilinguality from monolingual language models. Conneau et al. [11] demonstrated that transfer is possible even when there is no shared vocabulary and that representations from monolingual BERT models in different languages can be aligned post hoc quite effectively.

While these papers provide interesting but inconclusive and contradictory findings, they have all focused on NLP tasks. To our knowledge, we are the first to systematically study and categorize the cross-lingual transfer abilities of mBERT and to attempt an explanation for the (often counterintuitive) observed behaviors, specifically for a retrieval task.

3 EXPERIMENTAL DESIGN

3.1 Design Space

The application of transformers to tackle modern NLP problems follows an approach that begins with pretraining followed by fine-tuning. In many cases, the pretraining step can be omitted altogether by using an existing model checkpoint as a starting point; such models are widely shared, for example, on the HuggingFace model hub [57]. IR researchers have adopted the same high-level approach for tackling retrieval tasks [32], and training a DPR model is no exception. Typically, researchers begin with a pretrained model checkpoint as the backbone and further fine-tune the model with relevance judgments (e.g., from the MS MARCO passage ranking dataset) in a bi-encoder architecture as discussed above.

However, beyond this general “pretrain then fine-tune” recipe, there are several more detailed steps that deserve consideration. Together, they circumscribe the design space of our experiments. A more detailed articulation of the steps is shown in Figure 2 and discussed below:
Fig. 2. An illustration of our design space, following the typical four steps in applying transformers to NLP and IR applications: pretrain (PT) a randomly initialized transformer model on a large amount of unlabeled text in a self-supervised manner; additional pretrain (PT) on unsupervised data in the target domain or using a target task objective, starting from a pretrained checkpoint; pre–fine-tune (pFT) on out-of-domain but manually labeled data; and, finally, fine-tune (FT) on in-domain, manually labeled data. For the first step, we use publicly available checkpoints. For the second step, we use publicly available checkpoints as well as perform adaptation ourselves. For the final two steps, we perform all fine-tuning ourselves.

- **Pretrain (PT).** In the pretraining step, a transformer model with randomly initialized weights is trained “from scratch” on a large amount of unlabeled text in a self-supervised manner. The goal of pretraining is to imbue the model with “general-purpose” knowledge of language and the world that can be shared across downstream tasks. This is the most computationally expensive step, involving a massive amount of resources. The outputs of this step are usually shared with the community, for example, mBERT and XLM-R. In our work, we do not pretrain any models from scratch and instead start from these existing model checkpoints.

- **Additional pretrain (Additional PT).** Researchers have found it helpful to include additional self-supervised pretraining on top of a pretrained checkpoint, typically using a “general purpose” objective such as MLM [49] or some IR-specific pretraining regime [23, 36, 37]. In this work, mContriever is selected as a representative model that exploits IR-specific additional pretraining; our experiments build on publicly available model checkpoints. Separately, we perform additional pretraining ourselves in the context of DDR [63] and our improvements to that model.

- **Pre–fine-tune (pFT).** Similarly to additional pretraining, pre–fine-tuning is usually adopted to establish a better starting point for fine-tuning, or alternatively, the model can be used in a zero-shot setting. This step typically uses manually labeled (i.e., supervised) datasets, which lead to better task performance than the self-supervised approaches described in the previous two steps. However, pre–fine-tuning data usually comes from a different distribution (e.g., language or domain) than the final evaluation data. In a retrieval task, an example is using the MS MARCO passage ranking dataset for pre–fine-tuning, which has been shown to be effective [67]. In this work, we pre–fine-tune mDPR models ourselves based on the model checkpoints (i.e., backbones) described above.

- **Fine-tune (FT).** In the final step of building a transformer-based application, the model is fine-tuned on manually labeled data that draw from the same distribution as the final evaluation data. That is, fine-tuning typically takes advantage of “in-domain” data, as opposed to pre–fine-tuning, which usually refers to the use of “out-of-domain” data. In this work, we perform all model fine-tuning ourselves.
Table 1. A High-level Summary of Our Recommended Best Practices for Retrieval in Target Language $\mathcal{L}$, Grouped According to Three Scenarios in Terms of the Availability of Models and Data

| Scenario (1), “have model, no data” | Section 4 | ⊢ pFT on mMARCO in $\mathcal{L}$ (if available); otherwise on all mMARCO ⊢ additional PT with IC or MLM (mContreiver > mDDR+ > mBERT) ⊢ distill from cross-encoders when possible |
|------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scenario (2), “have model and data” | Section 5 | — multilingual BERT, data in target language $\mathcal{L}$ ⊢ pFT on English MS MARCO, FT on data in $\mathcal{L}$ ⊢ augmenting FT data with non-target language data can help ⊢ use hard negatives if possible |
|                                    | Section 6 | — multilingual BERT, data in non-target language $\mathcal{K}$ ⊢ pFT on English MS MARCO, FT on $\mathcal{K}$ data |
|                                    | Section 7 | — mono vs. multilingual BERT ⊢ mBERT “safer” than monolingual BERT in $\mathcal{L}$ |
| Scenario (3), “have data, no model” | Section 8 | ⊢ train DPR from English backbone ⊢ combine DPR + bag-of-words results |

(PT: pretrain; pFT: pre–fine-tune; FT: fine-tune).

As mentioned in the Introduction, our explorations are grouped into three scenarios: (1) have model, no data; (2) have model and data; and (3) have data, no model. Our recommendations under each scenario are further organized according to the above design space, shown in Table 1.

The majority of our experiments are based on mBERT for multiple reasons: First, keeping the backbone and training regime identical allows us to compare results across the different scenarios. Second, while there are indeed many multilingual (and monolingual) backbones that can serve as starting points for training dense retrieval models, all of them follow essentially the same training steps. As it is impractical to run all experimental conditions with all backbones, we argue that it suffices to experiment with a representative example. We selected mBERT as the representative, because it takes “reasonable” time and resources to train, allowing us to run many experimental conditions. In contrast, the training time of an XLM-R Large backbone is around 5× that of mBERT when consuming the same amount of data. In Sections 4 and 5, we additionally study the effects of different backbones and additional pretraining techniques, but it is impractical to exhaustively explore these approaches under all settings.

### 3.2 Datasets

All experiments and recommendations in this work are based on Mr. TYDI [65], a multilingual retrieval benchmark built on the TYDI [9] dataset (originally developed for question answering). Mr. TYDI covers 11 languages from different language families that are typologically diverse. Among the languages, English, Finnish, Indonesian, and Swahili are written in Latin script, whereas no other language pair shares the same script. The corpus for each language is drawn from Wikipedia, and the query and relevance judgments are prepared by native speakers of that language. Table 2 presents statistics of the Mr. TYDI dataset, copied from the original paper [65]. At the time we began our work, this was the only available dataset for evaluating monolingual retrieval in diverse languages; other datasets, for example, XOR-TYDI [3] and CLIRMatrix [54], focus on cross-lingual retrieval. We have since built MIRACL [66], a new multilingual retrieval dataset that spans 18 different languages, but this work does not take advantage of MIRACL.
Table 2. Statistics for Mr. TYDI: Number of Questions (# Q), Judgments (# J), and Number of Passages (Corpus Size) in Each Language

| Language     | ISO | Train |   |   | Dev |   |   | Test |   |   | Corpus Size |
|--------------|-----|-------|---|---|-----|---|---|------|---|---|-------------|
| Arabic       | (ar)| 12,377| 3,115| 1,081| 2,106,586 |
| Bengali      | (bn)| 1,713 | 440  | 111  | 304,059   |
| English ‡    | (en)| 3,547 | 878  | 744  | 32,907,100|
| Finnish ‡    | (fi)| 6,561 | 1,738 | 1,254 | 1,908,757 |
| Indonesian ‡ | (id)| 4,902 | 1,224 | 829  | 1,469,399 |
| Japanese     | (ja)| 3,697 | 928  | 720  | 7,000,027 |
| Korean       | (ko)| 1,295 | 303  | 421  | 1,496,126 |
| Russian      | (ru)| 5,366 | 1,375 | 995  | 9,597,504 |
| Swahili ‡    | (sw)| 2,072 | 526  | 646  | 548,224   |
| Telugu       | (te)| 3,880 | 983  | 646  | 568,855   |
| Thai         | (th)| 3,319 | 807  | 1,190 | 58,043,326|
| Total        |     | 48,729| 12,317| 8,661 | 58,043,326|

‡ Latin-script languages.

The test set of Mr. TYDI is used for the evaluation of our dense retrieval models, and thus the training and development sets of Mr. TYDI serve as the in-domain data for fine-tuning. In this work, another three datasets serve as pre–fine-tuning data, as they provide query–passage relevance labels but not exactly in the same domain (and language) as Mr. TYDI:

- **Natural Questions (NQ).** NQ is a question answering dataset released by Kwiatkowski et al. [27], which provides over 300k questions for training. The questions are taken from queries submitted to Google, and the passages are curated from Wikipedia. This dataset is widely used by QA researchers.

- **MS MARCO.** The MS MARCO passage ranking dataset grew out of an earlier QA dataset [5], where the queries draw from Bing and the passages comprise snippets of webpages. This dataset provides over 500k manually labeled query–passage pairs and is widely used by IR researchers.

- **mMARCO.** The mMARCO dataset [7] is a multilingual passage ranking dataset automatically translated from the MS MARCO passage ranking dataset above. We use the version that is translated with the Google Translate API.¹ The dataset contains 14 languages in total (including the original English), five of which overlap with languages in Mr. TYDI (ar, en, id, ja, and ru).

These pre–fine-tuning datasets allow us to explore the design space described in Section 3.1, particularly for the scenario where we do not have in-domain data. Specifically, pre–fine-tuned models can be applied in a zero-shot setting.

### 3.3 Data Preparation

For the three pre–fine-tuning datasets, we use the Tevatron [17] versions for NQ² and MS MARCO,³ and the official release for mMARCO.⁴ We use version 1.1 of the Mr. TYDI dataset

¹ [https://cloud.google.com/translate](https://cloud.google.com/translate)
² [https://huggingface.co/datasets/Tevatron/wikipedia-nq](https://huggingface.co/datasets/Tevatron/wikipedia-nq)
³ [https://huggingface.co/datasets/Tevatron/msmarco-passage](https://huggingface.co/datasets/Tevatron/msmarco-passage)
⁴ [https://huggingface.co/datasets/unicamp-dl/mmarco](https://huggingface.co/datasets/unicamp-dl/mmarco)
released on HuggingFace. In the training sets used to train our dense retrieval models, positive examples are from the labeled positive passages provided by Mr. TYDI. Unless otherwise specified, the negative examples are prepared from the top-30 results retrieved using the tuned BM25 baseline described in Zhang et al. [65], following standard practice [26]. In Section 4, some experiments take advantage of hard negative mining [60]. In these experiments, the negative examples are mined from the top-30 results retrieved by an initial mDPR model that is initialized from mContriever and then pre–fine-tuned on the entire mMARCO dataset, i.e., the model from row (5) in Table 5.

In our experiments, we train models with data from a single language and with data combined from multiple languages. In the multiple languages case, data in the same batch are always drawn from the same language for both the pre–fine-tuning and fine-tuning steps. Pilot experiments indicate that this practice is helpful, because it prevents in-batch negatives from degenerating into a language detection task, where the model tries to distinguish positive passages (in one language) from negative passages (in another language).

For experiments in Section 6, where we fine-tune in language \( K \) and evaluate on language \( L \) in a large matrix experiment, we wish to alleviate the effects of training dataset size, since different languages have different numbers of relevance judgments. However, based on pilot experiments, simply downsampling all datasets to the smallest language results in under-fitting for most of the languages. Instead, we randomly sample 3300 training queries from each language, which is close to the size of training queries in Thai. For Bengali, Korean, and Swahili, which have fewer relevance judgments, we retain all available data. It is worth emphasizing that downsampling is applied only for the matrix experiments in Section 6; all other experiments in this work use all available data.

### 3.4 Experimental Setup

**Dense retrieval model training.** Our models are trained using Tevatron [17], a more efficient and flexible implementation of the original DPR code open sourced by Karpukhin et al. [26]. Following the standard practices of the community today, we train different dense retrieval models with shared parameters between the query and passage encoders, whereas in the original Mr. TYDI work [65], separate query and passage encoders are trained. In practice, a shared encoder yields comparable effectiveness but has the advantage of simplified model management. We initialize the model with different pretrained model checkpoints, all of which are available on HuggingFace [57] (see Table 3).

Unfortunately, not all languages have publicly available monolingual pretrained BERT models. Furthermore, we only considered monolingual models that were trained from scratch (not initialized with existing models). In the end, we experimented with monolingual BERT models for five languages in Mr. TYDI: Arabic, English, Finnish, Indonesian, and Korean. These were the only models that we were aware of when we first began our work.

In all experiments, models are trained 40 epochs with the corresponding training data (single or multiple languages) with 128 batch size. As mentioned above, when the model is trained on data in multiple languages, data in the same batch are always drawn from the same language in both pre–fine-tuning and fine-tuning steps. We use the Adam optimizer, setting the learning rate to 4e-5 for experiments trained on a single language and 1e-5 for the others. The maximum lengths of queries and passages are set to 64 and 256, respectively. We tuned the learning rate and the number of epochs on the Mr. TYDI development sets, considering {5e-6, 1e-5, 2e-5, 4e-5, 8e-5} and {10, 20, 30, 40, 50}, respectively, to arrive at settings that appear to work well across all languages.
Table 3. HuggingFace Models Used in Our Experiments

| Language   | HuggingFace Model                             |
|------------|-----------------------------------------------|
| multilingual pretrained transformers         |                                              |
| mBERT      | bert-base-multilingual-cased                 |
| XLM-R_Large| xlm-roberta-large                             |
| mContriever | facebook/mcontriever                         |
| AfriBERTa  | castorini/afriberta_large                    |
|            |                                              |
| monolingual pretrained transformers          |                                              |
| Arabic     | asafaya/bert-base-arabic                     |
| English    | bert-base-uncased                            |
| Finnish    | TurkuNLP/bert-base-finnish-cased-v1          |
| Indonesian | cahya/bert-base-indonesian-522M              |
| Korean     | kykim/bert-kor-base                          |

Additional pretraining with MLM and adapters. The implementation of mDDR and mDDR+ are based on Tevatron [17] and AdapterHub [45]. Adapter fine-tuning shares the same configuration as dense retrieval training above. In MLM pretraining, the models are trained for 50k steps, with batch size 2,048 and learning rate 1e-4. We set linear warm-up in the first 10k steps. The additional pretraining is conducted on a TPU v3-8 using a gradient accumulation of 16.

Cross-encoder knowledge distillation. The implementation of cross-encoder knowledge distillation is extended from Tevatron, sharing the same data pipeline as the dense retrieval model experiments. To train the cross-encoder based on XLM-R_large for knowledge distillation, we use 32 batch size with learning rate 5e-6. We distill the cross-encoder into mDPR using KL-divergence [20] with batch size 512. These reported settings are based on pilot experiments on a combination of a cross-encoder selected from \{XLM-R_large, mT5\}, a distillation method selected from \{KL-divergence Loss [20], MarginMSE [21], TASB [22]\}, and tested on eight languages (all but \{en, ja, ru\}, due to computation costs). While results on each language vary, we find that KL-divergence loss achieves the highest average MRR@100 over the tested languages.

Hybrid fusion. For experiments that involve combining results from BM25 and mDPR, we follow exactly the sparse–dense hybrid approach in Zhang et al. [65]. The final retrieval score $s_{\text{hybrid}}$ is computed as a linear combination of the BM25 score $s_{\text{sparse}}$ and the mDPR score $s_{\text{dense}}$, with $\alpha$ as a weighting parameter tuned on the development set.

Evaluation. Following Zhang et al. [65], we report MRR@100 and Recall@100 on the test sets of each language in Mr. TYDI. In our analyses, we report statistically significant differences detected using paired t tests ($p < 0.01$). Wary of the dangers of multiple hypothesis testing, we do not apply tests indiscriminately but rather only to answer specific research questions.

4 HAVE MODEL, NO DATA

Definition. In Scenario (1), the target language $L \in \text{mBERT}$, but we have no in-domain data in the target language $L$. Note that here, as well as in the following sections, we use mBERT as a convenient shorthand to refer to a multilingual pretrained transformer. Later in this section, we describe experiments that replace mBERT with enhanced backbones; while effectiveness improves, our core findings and recommendations do not.

Recommendations. Pre–fine-tune mDPR starting with a multilingual backbone (from a public checkpoint) on the mMARCO dataset. Use training data only in the target language $L$ if they are
Table 4. “Have Model, No Data”: Choice of Pre–fine-tuning Data

|   | ar | bn | en | fi | id | ja | ko | ru | sw | te | th | Avg |
|---|----|----|----|----|----|----|----|----|----|----|----|-----|
| (1) 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 |
| (2) BM25 (tuned) | 0.367 | 0.413 | 0.151 | 0.288 | 0.382 | 0.217 | 0.281 | 0.329 | 0.329 | 0.396 | 0.424 | 0.417 | 0.333 |
| (3) mDPR (NQ pFT) | 0.184 | 0.279 | 0.315 | 0.240 | 0.273 | 0.194 | 0.219 | 0.292 | 0.143 | 0.075 | 0.147 | 0.215 |
| (4) mDPR (MS pFT) | 0.441 | 0.397 | 0.327 | 0.275 | 0.352 | 0.311 | 0.282 | 0.356 | 0.342 | 0.310 | 0.269 | 0.333 |
| (5) mDPR (ar-mMARCO pFT) | 0.499 | – | 0.285 | – | 0.388 | 0.297 | – | 0.297 | – | – | – | – |
| (6) mDPR (id-mMARCO pFT) | 0.466 | – | 0.300 | – | 0.500 | 0.334 | – | 0.334 | – | – | – | – |
| (7) mDPR (ja-mMARCO pFT) | 0.422 | – | 0.290 | – | 0.341 | 0.365 | – | 0.365 | – | – | – | – |
| (8) mDPR (ru-mMARCO pFT) | 0.415 | – | 0.265 | – | 0.350 | 0.280 | – | 0.367 | – | – | – | – |
| (9) mDPR (all-mMARCO pFT) | 0.466 | 0.401 | 0.282 | 0.346 | 0.458 | 0.323 | 0.327 | 0.386 | 0.381 | 0.360 | 0.371 | 0.389 |

(a) MRR@100

|   | ar | bn | en | fi | id | ja | ko | ru | sw | te | th | Avg |
|---|----|----|----|----|----|----|----|----|----|----|----|-----|
| (1) 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 |
| (2) 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 |
| (3) mDPR (NQ pFT) | 0.493 | 0.703 | 0.757 | 0.601 | 0.653 | 0.546 | 0.545 | 0.667 | 0.414 | 0.190 | 0.409 | 0.548 |
| (4) mDPR (MS pFT) | 0.797 | 0.784 | 0.754 | 0.647 | 0.736 | 0.732 | 0.617 | 0.743 | 0.634 | 0.782 | 0.595 | 0.711 |
| (5) mDPR (ar-mMARCO pFT) | 0.856 | – | 0.718 | – | 0.798 | 0.710 | – | 0.710 | – | – | – | – |
| (6) mDPR (id-mMARCO pFT) | 0.822 | – | 0.752 | – | 0.867 | 0.751 | – | 0.751 | – | – | – | – |
| (7) mDPR (ja-mMARCO pFT) | 0.800 | – | 0.737 | – | 0.770 | 0.765 | – | 0.765 | – | – | – | – |
| (8) mDPR (ru-mMARCO pFT) | 0.768 | – | 0.681 | – | 0.715 | 0.684 | – | 0.744 | – | – | – | – |
| (9) mDPR (all-mMARCO pFT) | 0.803 | 0.806 | 0.677 | 0.722 | 0.803 | 0.688 | 0.616 | 0.722 | 0.665 | 0.893 | 0.740 | 0.740 |

(b) Recall@100

MRR@100 and Recall@100 across all 11 languages and an overall average on the Mr. TyDi test set. All conditions are zero-shot; the models have not seen any training data from Mr. TyDi. The first two rows are baselines using BM25. The remaining rows are zero-shot results based on mDPR (DPR initialized with mBERT) pre–fine-tuned (pFT) on different datasets. NQ: NaturalQuestions; MS: the original MS MARCO dataset in English; L-mMARCO: MS MARCO automatically translated to language $L$, where $L \in \{ar, id, ja, ru\}$; all-mMARCO: the entire mMARCO dataset, which contains 14 languages (including English). We underline the entries where the pre–fine-tuning dataset is in (and only in) the target language, which are also the best scores in the corresponding columns (with the exception of ru in terms of Recall@100).

Explanation. In this section, we provide supporting experimental evidence for the above recommendations, drawn from Tables 4–6. In all these tables, each row represents an experimental condition and each column shows the effectiveness (MRR@100 on top, Recall@100 on bottom) for a particular language. At the top of Table 4, rows (1) and (2) capture BM25 with default and tuned parameters, copied directly from the original Mr. TyDi paper [65], which are used as baselines. We present explorations from three perspectives: choice of pre–fine-tuning data (Sections 4.1 and 4.2), enhanced backbones (Section 4.3), and knowledge distillation from cross-encoders (Section 4.5).

4.1 English Pre–Fine-Tuning Data

In Table 4, row (3) follows the experimental setup of mDPR in Zhang et al. [65], where the model is fine-tuned on NQ [27]. However, as mentioned in Section 3.4, we share the parameters of the query and passage encoders rather than train two separate encoders, as in the original DPR paper [26]. Row (3) represents a zero-shot baseline using a multilingual dense retrieval model that has not seen any annotated examples from Mr. TyDi.
Building on this, the obvious improvement is to follow the same procedure as row (3) but using a different dataset. In row (4), we fine-tune mDPR on the MS MARCO passage ranking dataset [5], which is larger than NQ, and then apply inference for all languages in a zero-shot manner. As discussed in Section 3.1, we refer to the approach in rows (3) and (4) as pre–fine-tuning (or pFT for short) [67] to distinguish additional fine-tuning that we can further perform downstream using labeled data drawn from the same distribution as the evaluation data (discussed later).

We clearly see that pre–fine-tuning on MS MARCO, row (4), consistently beats NQ, row (3). These differences are statistically significant across all languages and support our recommendation that if one does not have data in the target language \( L \), then pre–fine-tuning on a large existing “out-of-domain” dataset is the best course of action. This strategy takes advantage of cross-lingual transfer effects and is consistent with previous work. The dataset used for pre–fine-tuning does matter, and here we find that MS MARCO is better than NQ.

These results also show that the domain of a dataset may not be the only or even the most important factor in its transfer ability: While results with NQ pre–fine-tuning are worse than with MS MARCO, its domain is “closer” to Mr. TYDI—both contain relatively well-formed questions posed against Wikipedia—whereas MS MARCO contains noisier and (sometimes) ill-formed questions, annotated against snippets extracted from webpages. Other factors, for example, the dataset size, the query length, the document length, and so on, may also play important roles.

4.2 Multilingual Pre–Fine-Tuning Data

While we observe clear transfer effects from English pre–fine-tuning to other languages in downstream retrieval tasks, it is unclear whether using machine-translated training data in the target language achieves better zero-shot effectiveness. We explore this question using mMARCO [7], where 5 of the 14 languages overlap with Mr. TYDI (ar, en, id, ja, and ru). Obviously, it would be ideal to have large manually annotated datasets like MS MARCO in multiple non-English languages, but this is obviously impractical due to the substantial costs of assembling such a dataset. The point of these experiments is to determine whether automatically machine-translated versions of MS MARCO can nevertheless be beneficial.

**Single-language pre–fine-tuning data.** In Table 4, rows (5–8), we train the model using mMARCO in the languages \{ar, id, ja, ru\} with exactly the same pre–fine-tuning regime as row (4). That is, rows (4–8) can be viewed as a group where they only differ in the pre–fine-tuning language. These languages are chosen because they are the overlapping languages between Mr. TYDI and mMARCO. We then evaluate the models on the same five languages, focusing on the effectiveness differences when the pre–fine-tuning data are in the target language versus when it is not.

Comparing the results across rows (4–8), we see that on all five languages, the zero-shot MRR@100 scores are the highest when the model is pre–fine-tuned on the target language (the underlined entries in Table 4). Additionally, the scores vary to a large extent when pre–fine-tuning in different languages, showing that the zero-shot effectiveness is sensitive to the dataset language. These results affirm the findings of Bonifacio et al. [7] for dense retrieval, demonstrating that pre–fine-tuning a dense retrieval model on target-language training data can improve zero-shot effectiveness even when the data are automatically translated, despite the fact that the data may be of lower quality due to translation errors (holding size and domain constant).

**Multi-language pre–fine-tuning data.** Knowing that pre–fine-tuning on target-language training data is consistently better (even if automatically translated), the next question is as follows: How would additional non-target languages in the pre–fine-tuning data affect the results?

This is investigated in row (9), where we pre–fine-tune mDPR on all 14 languages provided by mMARCO (including English). Interestingly, based on observations on the five languages where
Mr. TYDI and mMARCO overlap (ar, en, id, ja, ru), pre–fine-tuning in only the target language itself shows the best results. That is, pre–fine-tuning in more than just the target language hurts zero-shot effectiveness, which echoes the “curse of multilinguality” [10] even though we are only using 14 languages. This is shown by comparing the underlined scores in rows (4)–(8) to the score in the same column in row (9). Take Arabic as an example: While mDPR pre–fine-tuned on the ar–mMARCO training data achieves MRR@100 0.499, shown in row (5), column ar, it only achieves MRR@100 0.466 when pre–fine-tuned on all languages, shown in row (9), column ar. The difference is significant for 4 of 5 languages (ar, en, id, ja) and the scores are only on par for ru (0.367 vs. 0.366).

However, experimental results suggest that it is still better to use multiple languages for pre–fine-tuning, compared to just using a single non-target language (including pre–fine-tuning only on the original English version of MS MARCO). This conclusion is based on the observation that each entry in row (9) is higher than most of the non-underlined corresponding entries in rows (4–8). For example, the MRR@100 of ar in row (9) is 0.466, which is higher than or similar to the corresponding scores in rows (4), (6), (7), and (8). The MRR@100 of fi in row (9) is 0.346, which is higher than the score in row (4).

Synthesizing these results, we arrive at the following recommendations: For searching in target language $L$, pre–fine-tune only on MS MARCO translated into $L$ if it exists. Otherwise, pre–fine-tuning on all available translated versions of MS MARCO (including the original English version) appears to beat just pre–fine-tuning on MS MARCO translated into a single non-target language.

### 4.3 Enhanced Backbones

Another approach to achieve better generalization capacity is to start from better backbones. These results are shown in Table 5, where models are pre–fine-tuned under two conditions: on English MS MARCO only in the first block (MS pFT) and on all languages in mMARCO in the second block (all-mMARCO pFT). We compare three approaches to backbone enhancement.

**General multilingual backbone.** This is represented by XLM-R$_\text{Large}$ [10], with results reported in rows (1) and (4). Comparing row (1) to row (T4–4), XLM-R$_\text{Large}$ shows better overall zero-shot transfer ability than mBERT (average MRR@100 0.354 vs. 0.333), although at the cost of twice the model size and 5× training time. Additionally, we find its advantage lost when pre–fine-tuning using multilingual data: When pre–fine-tuning the model on all-mMARCO rather than MS MARCO, although the average score improves, XLM-R$_\text{Large}$ is now surpassed by mBERT, as shown in rows (4) and (T4–9), average MRR@100 0.368 vs. 0.389. When comparing improvements per language, row (1) vs. row (4), we see that 5 (bn, id, ja, ko, ru) of the 10 languages (excluding en) show worse or similar zero-shot MRR@100. This can be explained by the fact that XLM-R$_\text{Large}$ has already “absorbed” a large amount of multilingual knowledge (more so than mBERT) and thus benefits less from additional machine-translated data in the pre–fine-tuning step.

**Additional pretraining on the target task.** This is represented by mContriever [23], which started from mBERT and was then additionally pretrained on unsupervised synthetic data for information retrieval tasks using contrastive learning. Results with mContriever are reported in rows (2) and (5). Under both MS pFT and all-mMARCO pFT conditions, mContriever outperforms mBERT and XLM-R$_\text{Large}$ by a large margin, which affirms its zero-shot transfer ability as reported in the original paper [23]. Similarly to observations with XLM-R$_\text{Large}$, its improvement over mBERT is more prominent when fine-tuned on MS MARCO (0.409 vs. 0.333, improvement of 23%) compared to all-mMARCO (0.421 vs. 0.389, improvement of only 8%).

Based on these experiments, our recommendation that pre–fine-tuning in multiple languages is better than a single non-target language holds as the backbone changes. This is supported by
Table 5. "Have Model, No Data": Enhanced Backbones

|        | ar  | bn  | en  | fi   | id   | ja   | ko   | ru   | sw   | te   | th   | Avg  |
|--------|-----|-----|-----|------|------|------|------|------|------|------|------|------|
| **MS pFT** |     |     |     |      |      |      |      |      |      |      |      |      |
| (1) mDPR init. mBERT | 0.441 | 0.397 | **0.327** | 0.275 | 0.352 | 0.311 | 0.282 | 0.356 | 0.342 | 0.310 | 0.269 | **0.333** |
| (2) mDPR init. XLM-R<sub>large</sub> | 0.393 | 0.429 | 0.303 | 0.298 | 0.413 | 0.336 | 0.375 | 0.333 | 0.151 | 0.401 | 0.460 | 0.354 |
| (3) mDDR | 0.500 | 0.461 | 0.369 | 0.461 | 0.310 | 0.349 | 0.333 | 0.286 | 0.280 | 0.333 | 0.282 | **0.342** |
| (3') mDDR<sup>T</sup> | 0.451 | 0.401 | 0.302 | 0.386 | **0.463** | 0.319 | 0.360 | 0.346 | **0.533** | 0.251 | 0.394 | 0.382 |
| **all-mMARCO pFT** |     |     |     |      |      |      |      |      |      |      |      |      |
| (1) mDPR init. XLM-R<sub>large</sub> | 0.466 | 0.401 | 0.282 | 0.346 | 0.458 | 0.323 | 0.376 | 0.366 | 0.560 | 0.371 | 0.389 |      |
| (2) mDPR init. mContriever | 0.435 | 0.391 | 0.243 | 0.338 | 0.413 | 0.308 | 0.298 | 0.325 | 0.346 | 0.151 | 0.401 | 0.368 |
| (3) mDDR | 0.505 | 0.467 | 0.275 | 0.393 | 0.448 | 0.349 | 0.369 | 0.369 | 0.484 | 0.186 | 0.361 | 0.337 |
| (3') mDDR<sup>T</sup> | 0.483 | 0.451 | 0.302 | 0.386 | 0.463 | 0.319 | 0.360 | 0.346 | 0.533 | 0.251 | 0.394 | 0.382 |

(a) MRR@100

|        | ar  | bn  | en  | fi   | id   | ja   | ko   | ru   | sw   | te   | th   | Avg  |
|--------|-----|-----|-----|------|------|------|------|------|------|------|------|------|
| **MS pFT** |     |     |     |      |      |      |      |      |      |      |      |      |
| (1) mDPR init. mBERT | 0.797 | 0.784 | **0.734** | 0.647 | 0.736 | 0.732 | 0.617 | 0.743 | 0.634 | 0.782 | 0.595 | 0.711 |
| (2) mDPR init. XLM-R<sub>large</sub> | 0.758 | 0.824 | 0.720 | 0.670 | 0.812 | **0.758** | 0.676 | 0.761 | 0.405 | 0.676 | **0.875** | 0.721 |
| (3) mDDR | 0.836 | 0.878 | 0.727 | 0.785 | 0.821 | 0.713 | 0.706 | 0.745 | 0.766 | 0.868 | 0.859 | 0.791 |
| (3') mDDR<sup>T</sup> | 0.826 | 0.847 | 0.745 | 0.804 | **0.842** | 0.696 | **0.708** | 0.738 | 0.875 | 0.601 | 0.784 | 0.769 |
| **all-mMARCO pFT** |     |     |     |      |      |      |      |      |      |      |      |      |
| (1) mDPR init. XLM-R<sub>large</sub> | 0.803 | 0.806 | 0.677 | 0.722 | 0.803 | 0.688 | 0.616 | 0.722 | 0.665 | 0.893 | 0.740 | 0.803 |
| (2) mDPR init. mContriever | 0.757 | 0.806 | 0.616 | 0.721 | 0.777 | 0.673 | 0.590 | 0.682 | 0.706 | 0.846 | 0.862 | 0.731 |
| (3) mDDR | 0.814 | 0.874 | 0.686 | 0.801 | 0.837 | 0.715 | 0.687 | **0.764** | 0.855 | **0.928** | 0.874 | **0.803** |

(b) Recall@100

The observation that MRR@100 in rows (4) and (5) are generally higher than the corresponding conditions in rows (1) and (2) for the non-English entries. We set aside en as a special case, since rows (1) and (2) are pre–fine-tuned on English MS MARCO. That is, the models are pre–fine-tuned in the target language when evaluated on English.

Additional pretraining in the target language. This is represented by mDDR and mDDR<sup>T</sup> [63], where the models also start from mBERT but with additional pretraining on target language corpora with MLM loss (see Section 2.2 for details). As mDDR adapts mBERT to each language, we do not further pre–fine-tune the model with mMARCO, which “dilutes” the model for non-target languages. Results of mDDR are reported in row (3), where we observe that the strategy of continuing to pretrain on target corpora does not consistently improve the out-of-language generalization capacity: Compared to row (T4–4), we observe improvements with mDDR, row (3), on some of the languages (fi, id, ko, sw, th), but effectiveness drops for the other languages, some by a large margin. Overall, mDDR achieves similar effectiveness as mDPR initialized from mBERT, differing only by 0.004 in average MRR@100.

However, effectiveness is greatly improved after additional pre–fine-tuning on the adapter. Comparing row (3’) to row (3), we see that MRR@100 improves for all languages and the average increases from 0.337 to 0.382. These results affirm our hypothesis about the misaligned weights between the language-adapted backbone and the adapter. Additionally, average MRR@100 is now on par with mDPR pre–fine-tuned on all mMARCO data, initialized from mBERT, i.e., row (3’) vs. row (4); the scores are close for most of the languages, except sw and te. These results suggest that additional MLM pretraining and pre–fine-tuning on data in multiple languages serve similar functions; see discussions in the next section.
4.4 Discussion: Relevance Transfer and Language Adaptation

Taking a step back and reviewing the above zero-shot approaches, we can understand them as attempts to “approximate” in-language, in-domain fine-tuning data in a cheap manner from two perspectives: (1) query–document relevance, e.g., from human-labeled datasets such as MS MARCO and from synthetic relevance data in mContriever, and (2) target-language knowledge, e.g., from machine translated data and from additional MLM pretraining on the target-language corpus. Based on this perspective, the above empirical results suggest that

(1) regarding query–document relevance: IC-style additional pretraining on synthetic relevance data provides gains beyond English MS MARCO, since we observe that mContriever + MS MARCO > mBERT + MS MARCO.

(2) regarding target-language knowledge: additional MLM pretraining yields similar results as pre–fine-tuning on a human-labeled retrieval dataset that has been translated into multiple languages, but both appear to be worse than target-language translations of the same dataset (in all cases, we consider imperfect machine translations). This is seen in mBERT + target-L-mMARCO > mBERT + all-mMARCO = mDDR+. In other words, machine translation should be deployed in a targeted manner.

While this summary provides a better understanding of the comparative effectiveness under each experimental configuration, it is also worth discussing the cost of each approach, especially with respect to point (2) above: While mMARCO provides out-of-the-box translations for 13 languages, it is an existing resource from our perspective and hence “cheap” for us to exploit. However, in the general case, for a new target language L, performing machine translation on a similar scale remains costly and challenging (not to mention that a good translation system for L might not even be available). Similarly, the pretraining of mContriever depends on CCNet [56] and requires careful preprocessing of web data. Additionally, based on our experience, effective IC-based pretraining is non-trivial in terms of properly setting hyperparameters and other experimental configurations to achieve the desired gains. On the contrary, while additional pretraining with MLM does not yield the best results, it requires only an unlabeled corpus in the target language. Thus, we suggest that additional MLM pretraining in the target language may be an effective substitute when the translation of pre–fine-tuning data is not practical.

4.5 Knowledge Distillation

The above experiments compare different approaches to improving out-of-language generalization using only the mDPR model (including different backbones). We now discuss the case where we have access to cross-encoders, which have been shown to be more robust in out-of-domain tasks [21].

Results are shown in Table 6. The top row presents the scores of monoXLM-R\textsubscript{Large}, a variant of the monoBERT cross-encoder [42] initialized from XLM-R\textsubscript{Large}. The model reranks the top-100 candidates returned from BM25, i.e., row (2) in Table 4. The second block, rows (T4–4) and (T5–2), shows the baselines of mDPR initialized from different backbones (mBERT and mContriever) and then pre–fine-tuned on English MS MARCO, copied from row (4) in Table 4 and row (2) in Table 5, respectively. Comparing row (A) to the two baselines, we see that monoXLM-R\textsubscript{Large} MRR@100 exceeds mDPR by a large margin, which confirms the effectiveness of cross-encoders in the out-of-language scenario.

The third block of the table presents our knowledge distillation results: rows (1) and (2) report mDPR initialized from mBERT or mContriever, distilled from the cross-encoder in row (A). Compared to the two baselines above, these models consume the same amount of resources during
Table 6. “Have Model, No Data”: Knowledge Distillation

|                  | ar | bn | en | fi | id | ja | ko | ru | sw | te | th | Avg |
|------------------|----|----|----|----|----|----|----|----|----|----|----|-----|
| (A) monoXLM-R_{Large} | 0.651 | 0.595 | 0.366 | 0.536 | 0.635 | 0.476 | 0.451 | 0.532 | 0.632 | 0.787 | 0.686 | 0.577 |
| (T4–4) mDPR init. mBERT | 0.441 | 0.397 | 0.327 | 0.275 | 0.352 | 0.311 | 0.282 | 0.356 | 0.342 | 0.310 | 0.269 | 0.333 |
| (T5–2) mDPR init. mContriever | 0.500 | 0.461 | 0.301 | 0.369 | 0.461 | 0.310 | 0.349 | 0.357 | 0.461 | 0.450 | 0.483 | 0.409 |
| (1) Distilled mDPR init. mBERT | 0.503 | 0.432 | 0.304 | 0.399 | 0.463 | 0.358 | 0.359 | 0.398 | 0.456 | 0.389 | 0.340 | 0.395 |
| (2) Distilled mDPR init. mContriever | 0.554 | 0.488 | 0.292 | 0.413 | 0.357 | 0.275 | 0.405 | 0.333 | 0.506 | 0.639 | 0.595 | 0.448 |

(a) MRR@100

|                  | ar | bn | en | fi | id | ja | ko | ru | sw | te | th | Avg |
|------------------|----|----|----|----|----|----|----|----|----|----|----|-----|
| (A) monoXLM-R_{Large} | 0.793 | 0.870 | 0.537 | 0.720 | 0.843 | 0.643 | 0.619 | 0.654 | 0.764 | 0.897 | 0.853 | 0.745 |
| (T4–4) mDPR init. mBERT | 0.797 | 0.784 | 0.754 | 0.667 | 0.736 | 0.732 | 0.617 | 0.743 | 0.654 | 0.782 | 0.595 | 0.711 |
| (T5–2) mDPR init. mContriever | 0.836 | 0.878 | 0.727 | 0.785 | 0.821 | 0.713 | 0.706 | 0.745 | 0.766 | 0.868 | 0.859 | 0.791 |
| (1) Distilled mDPR init. mBERT | 0.830 | 0.860 | 0.655 | 0.770 | 0.815 | 0.733 | 0.687 | 0.765 | 0.757 | 0.798 | 0.700 | 0.759 |
| (2) Distilled mDPR init. mContriever | 0.864 | 0.896 | 0.705 | 0.840 | 0.864 | 0.741 | 0.695 | 0.874 | 0.961 | 0.896 | 0.818 |

(b) Recall@100

MRR@100 and Recall@100 of the cross-encoder along with mDPR that is pre–fine-tuned or distilled from the cross-encoder, evaluated across all 11 languages and an overall average on the Mr. TyDI test set. Models are pre–fine-tuned on English MS MARCO in all experiments. The first row denotes the monoXLM-R_{Large} cross-encoder and the second block captures baselines without knowledge distillation, copied from previous tables.

The results are positive. When initialized from either backbone, knowledge distillation yields big improvements: average MRR@100 from 0.333 to 0.395 with mBERT, row (T4–4) vs. row (1), and average MRR@100 from 0.409 to 0.448 with mContriever, row (T5–2) vs. row (2). Furthermore, a better initialization checkpoint boosts the knowledge distillation results: Starting from mContriever achieves 0.053 points higher average MRR@100 compared to mBERT, shown in row (2) vs. row (1), yielding the best zero-shot average MRR@100 on Mr. TyDI among the settings we have explored so far. Nonetheless, a large disparity in effectiveness persists between the distilled mDPR models and their cross-encoder teachers, suggesting the need for additional explorations in knowledge distillation techniques under multilingual scenarios.

5 HAVE MODEL AND DATA IN TARGET LANGUAGE: MULTILINGUAL BERT

Definition. In Scenario (2a), the target language $L \in$ mBERT and we have in-domain training data (manually labeled relevance judgments) in $L$.

Recommendations. Start with pre–fine-tuning mDPR using English MS MARCO data, then continue fine-tuning on relevance judgments in $L$. Augmenting the fine-tuning data with relevance judgments from other languages generally helps as well, although in some cases the results are mixed and the overall differences are small. Somewhat surprisingly, given in-domain training data in the target language, pre–fine-tuning on (only) English MS MARCO beats pre–fine-tuning on all languages in mMARCO. Additionally, training the model with in-language hard negative examples provides consistent gains.

5.1 English/Multilingual Pre–Fine-Tuning Data

Table 7 focuses on scenario (2a) specifically: Rows (T4–4) and (T4–9) are zero-shot results copied from rows (4) and (9) in Table 4. Rows (1)–(4) start with the raw mBERT checkpoint without any pre–fine-tuning and then fine-tune on different groups of Mr. TyDI training data. This is contrasted against rows (5)–(8), which start from the checkpoint pre–fine-tuned on English MS MARCO, row (T4–4). Two groups, rows (1) and (2) and rows (5) and (6), are additionally contrasted against rows (9) and (10), which start from the checkpoint pre–fine-tuned on all-mMARCO,
Table 7. “Have Model and Data”: Multilingual Pre–fine-tuning and Fine-tuning

| Model and Language | MRR@100 | Recall@100 |
|--------------------|--------|-----------|
| (14) mDPR (MS pFT) | 0.441  | 0.397  |
| (14) mDPR (all-mMARCO pFT) | 0.466  | 0.401  |
| (1) mDPR (in-lang FT) | 0.678  | 0.638  |
| (2) mDPR (all FT) | 0.695  | 0.659  |
| (3) mDPR (in-script FT) | 0.644  | 0.535  |
| (4) mDPR (out-script FT) | 0.457  | 0.353  |
| (5) mDPR (MS pFT + in-lang FT) | 0.691  | 0.651  |
| (6) mDPR (MS pFT + all FT) | 0.695  | 0.623  |
| (7) mDPR (MS pFT + in-script FT) | 0.473  | 0.355  |
| (8) mDPR (MS pFT + out-script FT) | 0.476  | 0.585  |
| (9) mDPR (mMARCO pFT + in-lang FT) | 0.669  | 0.648  |
| (10) mDPR (mMARCO pFT + all FT) | 0.660  | 0.651  |
| (11) mDDR (in-lang FT) | 0.677  | 0.585  |
| (12) BM25 + row (6) | 0.714  | 0.702  |

(a) MRR@100

| Model and Language | MRR@100 | Recall@100 |
|--------------------|--------|-----------|
| (14) mDPR (MS pFT) | 0.797  | 0.784  |
| (14) mDPR (all-mMARCO pFT) | 0.803  | 0.806  |
| (1) mDPR (in-lang FT) | 0.889  | 0.896  |
| (2) mDPR (all FT) | 0.894  | 0.937  |
| (3) mDPR (in-script FT) | 0.812  | 0.842  |
| (4) mDPR (out-script FT) | 0.817  | 0.543  |
| (5) mDPR (MS pFT + in-lang FT) | 0.891  | 0.914  |
| (6) mDPR (MS pFT + all FT) | 0.900  | 0.955  |
| (7) mDPR (MS pFT + in-script FT) | 0.847  | 0.865  |
| (8) mDPR (MS pFT + out-script FT) | 0.884  | 0.857  |
| (9) mDPR (mMARCO pFT + in-lang FT) | 0.890  | 0.941  |
| (10) mDPR (mMARCO pFT + all FT) | 0.878  | 0.914  |
| (11) mDDR (in-lang FT) | 0.904  | 0.937  |
| (12) BM25 + row (6) | 0.932  | 0.946  |

(b) Recall@100

MRR@100 and Recall@100 of mDPR evaluated across all 11 languages and an overall average on the Mr. TyDi test set. “MS pFT”: pre–fine-tuning on English MS MARCO; “all-mMARCO pFT”: pre–fine-tuning on all of mMARCO including en; “in-lang FT”: fine-tuning on the Mr. TyDi training set from only the target language; “all FT”: fine-tuning on Mr. TyDi training data from all languages; “in/out-script”: fine-tuning on training data from the target language and languages that are in the same/different scripts; for example, if the target language is en, the “in-script” languages would be the target language (en) and all the other Latin-script languages (f1, id, sw) and the “out-script” languages would be the target language (en) and all the non-Latin-script languages (ar, bn, ja, ko, ru, te, th).

row (T4–9). Specifically, rows (1), (5), and (9), tagged with “in-lang,” represent models fine-tuned using the Mr. TyDi training data only in the target language L. These results answer the following questions:

(i) Do we still need pre–fine-tuning if we have training data in the target language L?
(ii) If so, then does the model still benefit from additional multilingual pretraining and pre–fine-tuning as in the “have model, no data” scenario?

The answer to the first question appears to be, unequivocally, yes, as row (5) beats row (1) across all languages; these differences are statistically significant in 6 of the 11 languages. This means that no matter the target language, models benefit from cross-lingual transfer from English in the pre–fine-tuning step. Retrieval in Arabic, Bengali, and so on, still benefits from cross-lingual transfer with English data—despite the fact that the languages are so different (down to the written script).

However, the answer to the second question appears to be no. Comparing row (9) to row (5), pre–fine-tuning on all of mMARCO produces worse results than pre–fine-tuning on (only) English MS MARCO across all languages. These results seem to indicate that the multilingual information learned from the pre–fine-tuning step is secondary when downstream in-language data are
available and that multilingual pre–fine-tuning does not provide a better initialization point for subsequent fine-tuning, even though it shows better zero-shot effectiveness. Another way to explain this finding is that if “high quality” in-domain, in-language training data are available, adding lower quality data (e.g., automatically translated, out-of-domain) might actually reduce effectiveness.

Similar results are observed when the model is additionally pretrained using MLM on the target language corpus. As presented in row (11), starting from the mDDR+ checkpoint, row (3′) in Table 5, we continue to fine-tune the adapters on the in-language Mr. TYDI training data. We see that the advantage of language adaptation from mDDR+ appears to completely vanish after in-language fine-tuning: In row (11), starting from mDDR+ yields average MRR@100 0.542, which is lower than the score of directly fine-tuning mDPR from mBERT, average MRR@100 0.555, shown in row (1).

We thus conclude that knowledge gained from multiple languages prior to the fine-tuning stage, under the framework in Section 4.4, does not appear to provide additional benefits when in-language, in-domain fine-tuning data are available.

5.2 Multilingual Fine-Tuning Data

A closely related question: Can we further leverage cross-lingual transfer effects at the fine-tuning step? In Mr. TYDI, we have data for all languages, and so we can experiment with MS MARCO or mMARCO pFT and further fine-tune on all available data, in all languages. This answers the following research question: If we have in-domain relevance judgments in languages other than the target language, then are such data helpful?

The outcome of these experiments is presented in the remaining rows of Table 7. Results starting from the MS MARCO pFT checkpoint are shown in row (6). We see that using data from all languages beats or performs similarly to using data only from the target language, row (5), in all cases except Bengali; the gains are only statistically significant for Russian, though. Results starting from mMARCO pFT are similar. We observe that row (10), where the model is fine-tuned on all languages, outperforms row (9) on 8 of 11 languages. These results suggest that we can benefit from cross-lingual transfer in the fine-tuning step as well as the pre–fine-tuning step. However, the conclusions are equivocal: Augmenting in-domain fine-tuning data in target language L with in-domain data from other non-target languages generally helps, but not always. Nevertheless, the magnitudes of the observed differences are generally small.

Next, we attempt to better understand cross-lingual transfer effects from different non-target languages. For example, would models benefit more from languages sharing the same script with the target language? Rows (3), (4), (7), and (8) answer this question. Row (3) reports the results of the variant where we only fine-tune in languages written in the Latin script (there are no other language pairs that share the same script), and row (4) reports results that only fine-tune on the (Latin-script) target language itself and all other non-Latin-script languages (for example, the fine-tuning languages are all but fi, id, sw when the target language is en). For rows (7) and (8), we apply the same fine-tuning treatment as rows (3) and (4) but start from the checkpoint with pre–fine-tuning rather than the raw mBERT checkpoint.

We observe that scores in rows (3) and (4) are always higher than in row (1), which indicates that additional training data in both in-script and out-script languages are generally beneficial when not pre–fine-tuning. Comparing those two rows to row (2), we observe that using all available data additionally helps. However, this trend does not hold when pre–fine-tuning is added: The scores in rows (7) and (8) are quite close to the scores in row (5).

The summary here is that the models all appear to benefit from fine-tuning on all available data, although the effects are stronger without pre–fine-tuning. This does make sense, since with pre–fine-tuning, we are already obtaining maximal cross-language transfer effects from a large
MRR@100 and Recall@100 of mDPR initialized from mBERT and mContriever, evaluated across all 11 languages and an overall average on the Mr. TyDi test set. All experiments are first pre–fine-tuned on the entire mMARCO dataset, then on all Mr. TyDi training data.Rows marked with HN use the hard negative training data on Mr. TyDi prepared from the model of row (5) in Table 5 (mDPR init. mContriever + all-mMARCO pFT).

5.3 Enhanced Backbones and Hard Negatives

Results in Section 4 show that mDPR initialized from mContriever performs the best in the zero-shot condition. We thus take it as a starting point and compare the approach with mBERT in the current scenario, where in-language, in-domain data are available. Results are presented in Table 8. All these experiments start from the corresponding all-mMARCO pFT checkpoint in Table 5 and then are fine-tuned on training data in all Mr. TyDi languages.

We additionally investigate the effects of fine-tuning with hard negatives, which has been found to be helpful on English datasets but has not been explored in multilingual retrieval. In rows (2) and (4), where the experiments are marked with HN, the negative training examples are prepared from the top-30 results using the model in row (5) in Table 5 (DPR init. mContriever + all-mMARCO pFT) rather than from BM25 (the rest of the experiments in this article).

In terms of the average MRR@100 scores reported in rows (1) and (3), which do not use hard negatives, mContriever continues to outperform mBERT. However, looking at the effectiveness in each language, mContriever is only noticeably better on 4 of 11 languages (ko, sw, te, th), and this is only significant on sw and th, whereas mBERT appears to outperform mContriever on 5 (bn, en, id, ja, ru). The two models are on par with each other for the other two languages (ar, fi). Nevertheless, when fine-tuned on the hard negatives, as shown in rows (2) and (4), mContriever equals or outperforms mBERT on 9 languages (all but en and te), among which 3 are significant (ko, sw, th). These results suggest that mContriever can better take advantage of the hard negative training data compared to mBERT. Hard negatives increase MRR@100 by 6% with mContriever, comparing row (3) to row (4), while only 3% with mBERT, comparing row (1) to row (2). Additionally, hard negatives improve MRR@100 significantly on 5 languages (ar, en, ja, ru, th) with mContriever, while the improvement is only significant on 2 languages (en, ru) with mBERT.
Table 9. Matrix Experiments Training mDPR on the Language Denoted in the Row, Evaluating on the Language in the Column, Reporting MRR@100

|     | ar  | bn  | en  | fi  | id  | ja  | ko  | ru  | sw  | te  | th  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ar  | 0.629 | 0.560 | 0.373 | 0.455 | 0.483 | 0.367 | 0.384 | 0.451 | 0.460 | 0.506 | 0.387 |
| bn  | 0.448 | 0.638 | 0.309 | 0.372 | 0.385 | 0.314 | 0.326 | 0.325 | 0.433 | 0.532 | 0.341 |
| en  | 0.439 | 0.389 | 0.436 | 0.381 | 0.402 | 0.296 | 0.314 | 0.363 | 0.397 | 0.289 | 0.218 |
| fi  | 0.524 | 0.436 | 0.369 | 0.484 | 0.471 | 0.325 | 0.358 | 0.420 | 0.470 | 0.619 | 0.295 |
| id  | 0.486 | 0.490 | 0.375 | 0.415 | 0.524 | 0.326 | 0.353 | 0.372 | 0.442 | 0.366 | 0.221 |
| ja  | 0.480 | 0.573 | 0.328 | 0.391 | 0.395 | 0.471 | 0.382 | 0.401 | 0.441 | 0.615 | 0.345 |
| ko  | 0.421 | 0.464 | 0.315 | 0.373 | 0.377 | 0.309 | 0.383 | 0.330 | 0.421 | 0.500 | 0.181 |
| ru  | 0.476 | 0.524 | 0.337 | 0.410 | 0.400 | 0.364 | 0.311 | 0.419 | 0.424 | 0.531 | 0.243 |
| sw  | 0.422 | 0.397 | 0.251 | 0.324 | 0.367 | 0.258 | 0.304 | 0.317 | 0.580 | 0.276 | 0.228 |
| te  | 0.512 | 0.587 | 0.331 | 0.399 | 0.404 | 0.398 | 0.368 | 0.380 | 0.421 | 0.561 | 0.422 |
| th  | 0.505 | 0.554 | 0.319 | 0.389 | 0.410 | 0.361 | 0.323 | 0.416 | 0.411 | 0.648 | 0.588 |

(a) mDPR initialized from mBERT

|     | ar  | bn  | en  | fi  | id  | ja  | ko  | ru  | sw  | te  | th  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ar  | 0.663 | 0.608 | 0.450 | 0.450 | 0.508 | 0.432 | 0.392 | 0.509 | 0.425 | 0.613 | 0.380 |
| bn  | 0.549 | 0.645 | 0.432 | 0.427 | 0.468 | 0.432 | 0.401 | 0.435 | 0.438 | 0.644 | 0.417 |
| en  | 0.469 | 0.412 | 0.480 | 0.292 | 0.319 | 0.368 | 0.344 | 0.376 | 0.302 | 0.283 | 0.245 |
| fi  | 0.542 | 0.501 | 0.472 | 0.524 | 0.495 | 0.396 | 0.409 | 0.477 | 0.441 | 0.463 | 0.355 |
| id  | 0.551 | 0.538 | 0.432 | 0.441 | 0.561 | 0.385 | 0.396 | 0.443 | 0.430 | 0.496 | 0.349 |
| ja  | 0.531 | 0.500 | 0.374 | 0.367 | 0.405 | 0.483 | 0.396 | 0.453 | 0.385 | 0.429 | 0.375 |
| ko  | 0.515 | 0.450 | 0.430 | 0.379 | 0.428 | 0.426 | 0.455 | 0.499 | 0.413 | 0.404 | 0.356 |
| ru  | 0.570 | 0.521 | 0.448 | 0.437 | 0.488 | 0.427 | 0.400 | 0.499 | 0.421 | 0.510 | 0.351 |
| sw  | 0.498 | 0.470 | 0.388 | 0.409 | 0.440 | 0.365 | 0.353 | 0.428 | 0.635 | 0.494 | 0.331 |
| te  | 0.575 | 0.583 | 0.426 | 0.390 | 0.433 | 0.449 | 0.413 | 0.480 | 0.432 | 0.867 | 0.436 |
| th  | 0.587 | 0.530 | 0.446 | 0.410 | 0.490 | 0.427 | 0.411 | 0.480 | 0.428 | 0.605 | 0.611 |

(b) mDPR pre–fine-tuned on MS MARCO

Fine-tuning is performed on the mBERT backbone directly (top) and after pre–fine-tuning on MS MARCO (bottom). For en, ko, and sw, we use all available training data; for the remaining languages, we downsample to 3,300 training queries. Cell values are highlighted columnwise: the color gradient ranges from blue (high values) to orange (low values). Note that due to data downsampling, the diagonal results are not identical to results in Table 7.

To conclude, we find that the advantage of mContriever over mBERT still persists when in-language, in-domain fine-tuning data are available and that the advantage is more prominent when hard negative examples are exploited. This leads to our recommendation to use hard negative mining techniques if possible.

6 HAVE MODEL AND DATA IN NON-TARGET LANGUAGE: MULTILINGUAL BERT

Definition. In Scenario (2b), the target language \( L \in \text{mBERT} \), but we have no data in language \( L \). However, we have data in another language \( K \) and \( K \in \text{mBERT} \). Language \( K \) may not be linguistically related to language \( L \) and may not even be in the same script.

Recommendations. Start with pre–fine-tuning mDPR using English MS MARCO data, and then continue fine-tuning on relevance judgments in language \( K \). That is, models may benefit from cross-lingual transfer, surprisingly, across different language families and even different scripts.

Explanation. The results of a matrix experiment where we consider all combinations of \( L \) and \( K \) for all languages in Mr. TYDI are shown in Table 9. On the top, we fine-tune mDPR directly from the mBERT checkpoint, and on the bottom, we fine-tune an mDPR model that has already been fine-tuned on English MS MARCO.
Table 10. Element-wise Difference between MS pFT + FT and MS pFT (Zero-shot), Answering the Question "Should We Always Fine-tune on Non-target Language Data?"

|     | ar  | bn  | en  | fi  | id  | ja  | ko  | ru  | sw  | te  | th  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ar  | 0.221 | 0.211 | 0.123 | 0.175 | 0.156 | 0.122 | 0.110 | 0.152 | 0.085 | 0.303 | 0.111 |
| bn  | 0.108 | 0.248 | 0.105 | 0.152 | 0.116 | 0.121 | 0.119 | 0.079 | 0.096 | 0.334 | 0.147 |
| en  | 0.027 | 0.015 | 0.153 | 0.017 | 0.033 | 0.058 | 0.062 | 0.020 | 0.040 | 0.027 | 0.025 |
| fi  | 0.101 | 0.105 | 0.145 | 0.139 | 0.143 | 0.085 | 0.127 | 0.121 | 0.099 | 0.153 | 0.086 |
| id  | 0.110 | 0.141 | 0.105 | 0.166 | 0.209 | 0.074 | 0.114 | 0.086 | 0.088 | 0.186 | 0.080 |
| ja  | 0.090 | 0.103 | 0.047 | 0.092 | 0.053 | 0.172 | 0.114 | 0.097 | 0.043 | 0.119 | 0.105 |
| ko  | 0.074 | 0.053 | 0.103 | 0.104 | 0.076 | 0.115 | 0.171 | 0.143 | 0.071 | 0.094 | 0.067 |
| ru  | 0.128 | 0.124 | 0.121 | 0.162 | 0.136 | 0.117 | 0.118 | 0.143 | 0.079 | 0.200 | 0.062 |
| sw  | 0.057 | 0.073 | 0.061 | 0.134 | 0.088 | 0.054 | 0.071 | 0.072 | 0.293 | 0.183 | 0.061 |
| te  | 0.133 | 0.187 | 0.099 | 0.115 | 0.081 | 0.139 | 0.131 | 0.124 | 0.090 | 0.537 | 0.167 |
| th  | 0.145 | 0.133 | 0.119 | 0.135 | 0.138 | 0.116 | 0.129 | 0.124 | 0.086 | 0.295 | 0.342 |

The answer is yes, with the exception that further fine-tuning on English data for retrieval in non-English languages might result in overfitting. Positive values are highlighted in blue and negative values in orange, with saturation proportional to magnitude.

Let us consider the starting point of mDPR with MS MARCO pre–fine-tuning. The research question is as follows: We have training data in language $K$, but it is not in our target language $L$. Should we fine-tune on it anyway? That answer, surprisingly, is yes. These results are presented in Table 10, which shows the element-wise difference between MS pFT + FT vs. MS pFT (zero-shot). For example, mDPR (MS pFT) obtains 0.269 on Thai in a zero-shot setting, as seen in Table 4, row (4). However, if we fine-tune further on Telugu data, then we can achieve 0.436 on Thai (seen in Table 9, bottom, te row, th column). This represents an improvement of 0.167 (seen in Table 10, te row, th column). We obtain this gain even though Telugu and Thai have very little in common and are not even written in the same script.

Looking at Table 10, we see that all values are positive with the exception of a few in the en row; that is, when further fine-tuning on English data and evaluating on other languages. This tells us that, starting with the mDPR model pre–fine-tuned on MS MARCO, additional fine-tuning on language $K$ improves effectiveness, even if the target language is $L$, where $L \neq K$. The exception appears to be further fine-tuning in English, where we interpret the results to be a sign of overfitting, since the pre–fine-tuning data are already in English.

The above analysis uses pre–fine-tuning on MS MARCO as the starting point and then further fine-tuning on language-specific data. However, does pre–fine-tuning always improve effectiveness? That is, could it be the case that for target language $L$, if we have data in language $K$, then is it better to start from the raw mBERT checkpoint and skip pre–fine-tuning? This question is answered by Table 11, where we show the element-wise difference between MS pFT + FT vs. FT directly; this corresponds to the element-wise difference between the two matrices in Table 9. Along the diagonals, the answer is that pre–fine-tuning always helps, and this is consistent with the results in Section 5.1.

However, for scenario (2b), where $L \neq K$, the answer appears to be not necessarily so. For example, training on $K =$ Telugu and evaluating on $L = $ Japanese, starting with the MS pFT model pre–fine-tuned on MS MARCO. We report MRR@100: Each row represents the source language of the fine-tuning data and each column represents the target language. The diagonal corresponds to the “in-language” condition described in Section 5. However, as described in Section 3.3, we use the same amount of training data for all languages (exactly 3,300 queries), except for Bengali, Korean, and Swahili, which have fewer training queries and therefore we use all available data. Due to this difference, the results in the diagonals differ from the results in Table 7.

The above analysis uses pre–fine-tuning on MS MARCO as the starting point and then further fine-tuning on English data and evaluating on other languages. This tells us that, starting with the mDPR model pre–fine-tuned on MS MARCO, additional fine-tuning on language $K$ improves effectiveness, even if the target language is $L$, where $L \neq K$. The exception appears to be further fine-tuning in English, where we interpret the results to be a sign of overfitting, since the pre–fine-tuning data are already in English.
Table 11. Element-wise Differences between MS pFT + FT vs. FT Directly, Answering the Question "Does Pre–fine-tuning Always Improve Effectiveness?"

|     | ar  | bn  | en  | fi  | id  | ja  | ko  | ru  | sw  | te  | th  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ar  | 0.033 | 0.047 | 0.077 | −0.005 | 0.025 | 0.065 | 0.008 | 0.058 | −0.035 | 0.107 | −0.007 |
| bn  | 0.101 | 0.006 | 0.123 | 0.055 | 0.085 | 0.118 | 0.075 | 0.110 | 0.006 | 0.112 | 0.076 |
| en  | 0.029 | 0.023 | 0.044 | −0.089 | −0.083 | 0.072 | 0.030 | 0.013 | −0.095 | −0.006 | 0.026 |
| fi  | 0.018 | 0.066 | 0.103 | 0.040 | 0.024 | 0.071 | 0.052 | 0.057 | −0.030 | −0.156 | 0.060 |
| id  | 0.066 | 0.048 | 0.057 | 0.026 | 0.036 | 0.059 | 0.043 | 0.071 | −0.012 | 0.130 | 0.128 |
| ja  | 0.051 | −0.074 | 0.046 | −0.024 | 0.010 | 0.012 | 0.015 | 0.052 | −0.057 | −0.186 | 0.030 |
| ko  | 0.094 | −0.014 | 0.115 | 0.006 | 0.051 | 0.117 | 0.070 | 0.169 | −0.008 | 0.104 | 0.155 |
| ru  | 0.094 | −0.003 | 0.111 | 0.028 | 0.089 | 0.064 | 0.089 | 0.080 | −0.003 | −0.020 | 0.088 |
| sw  | 0.076 | 0.072 | 0.137 | 0.085 | 0.073 | 0.107 | 0.048 | 0.112 | 0.055 | 0.218 | 0.103 |
| te  | 0.063 | −0.003 | 0.095 | −0.009 | 0.029 | 0.052 | 0.045 | 0.100 | 0.011 | 0.006 | 0.014 |
| th  | 0.082 | −0.024 | 0.127 | 0.021 | 0.081 | 0.066 | 0.088 | 0.064 | 0.016 | −0.043 | 0.023 |

For In-language Training, the answer is yes, as the diagonal entries are always positive (in blue). For cross-lingual zero-shot transfer, not necessarily so, as can be seen by the negative (in orange) cases off-diagonal. However, we do observe more positive than negative cases, which indicates pre–fine-tuning may help in general. We thus believe that pre–fine-tuning is a “safer” choice. Color saturation is proportional to the magnitude of values.

confers an advantage of 0.052 over starting from the default mBERT checkpoint (seen in Table 11, te row, ja column); with pFT, we obtain 0.449, but without pFT, we only reach 0.398 (shown in Table 9). On the flip side, training on $K$ = Japanese and evaluating on $L$ = Telugu, starting from the raw mBERT checkpoint beats starting with MS pFT by 0.186; with pFT, we obtain 0.429, but without pFT, we reach 0.615. In other words, given target language $L$ and training data in another language $K$, it does not appear possible to determine, a priori, whether we should pre–fine-tune on MS MARCO first or directly use the raw mBERT checkpoint as the starting point.

Phrased slightly differently and summarizing our recommendations for scenario (2b), for a target language $L$, there does appear to exist a language $K$ (and in many cases, multiple languages) that provides positive cross-lingual transfer, although we are not able to generalize what characteristics lead to these beneficial effects. With pre–fine-tuning, further fine-tuning on language $K$ appears not to hurt (in general) and thus forms the basis of our recommendation. While it is true that in some cases one can achieve even better effectiveness without pre–fine-tuning (see Table 11), we struggle to provide a consistent, explainable pattern, and thus we believe that pre–fine-tuning is a “safer” choice.

7 HAVE MODEL AND DATA: MONO VS. MULTILINGUAL BERT

**Definition.** In the previous exploration of scenario (2a), we assume the starting point of mBERT (i.e., using mBERT as the backbone). Here, in scenario (2a’), we consider the case where a monolingual BERT model is available for the target language $L$. What should we do in this case?

**Recommendations.** Using a monolingual BERT backbone can yield a model that is more effective than using mBERT, but the monolingual model is not consistently better. Thus, it seems “safer” to just use mBERT as the backbone.

**Explanation.** Before presenting experimental results that support our recommendations, it makes sense to discuss the possible tradeoffs at work here. Previous work has found that monolingual BERT models can be more effective than mBERT on a variety of natural language processing tasks [1, 39, 50, 55], but the gains are not consistent. Obviously, the relative effectiveness depends on the quality of the monolingual model, the corpora used for pretraining, and a
Table 12. Rows (a)–(d): “Have Model and Data”: Mono vs. Multilingual BERT. Rows (e)–(h): “Have Model, no Data”

|                        | ar  | bn  | en  | fi  | id  | ja  | ko  | ru  | sw  | te  | th  | Avg |
|------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| (T7–5) mDPR (MS pFT + in-lang FT) | 0.691 | 0.651 | 0.489 | 0.551 | 0.562 | 0.488 | 0.453 | 0.485 | 0.450 | 0.564 | 0.706 | 0.619 | 0.591 |
| (T7–6) mDPR (MS pFT + all FT)   | 0.695 | 0.625 | 0.492 | 0.560 | 0.579 | 0.501 | 0.487 | 0.517 | 0.644 | 0.661 | 0.691 | 0.600 |
| (a) monolingual DPR (in-lang FT) | 0.678 | – | 0.426 | 0.573 | 0.545 | – | 0.476 | – | – | – | – | – |
| (b) monolingual DPR (in-script FT) | – | – | 0.412 | 0.540 | 0.488 | – | – | – | – | – | – | – |
| (c) monolingual DPR (out-script FT) | 0.682 | – | 0.426 | 0.522 | 0.540 | – | 0.454 | – | – | – | – | – |
| (d) monolingual DPR (all FT)    | 0.682 | – | 0.448 | 0.540 | 0.533 | – | 0.454 | – | – | – | – | – |
| (e) English DPR (in-lang FT)    | 0.578 | 0.261 | 0.426 | 0.385 | 0.396 | 0.084 | 0.011 | 0.291 | 0.447 | 0.001 | 0.007 | 0.262 |
| (f) English DPR (MS pFT + in-lang FT) | 0.592 | 0.318 | 0.497 | 0.423 | 0.439 | 0.218 | 0.182 | 0.298 | 0.499 | 0.001 | 0.030 | 0.318 |
| (g) AfriBERTa DPR (in-lang FT)  | 0.442 | 0.186 | 0.236 | 0.321 | 0.355 | 0.220 | 0.140 | 0.094 | 0.465 | 0.548 | 0.263 | 0.297 |
| (h) BM25 + row (f)            | 0.628 | 0.480 | 0.501 | 0.480 | 0.510 | 0.333 | 0.299 | 0.440 | 0.535 | 0.423 | 0.424 | 0.459 |

(a) MRR@100

|                        | ar  | bn  | en  | fi  | id  | ja  | ko  | ru  | sw  | te  | th  | Avg |
|------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| (T7–5) mDPR (MS pFT + in-lang FT) | 0.891 | 0.914 | 0.837 | 0.852 | 0.863 | 0.828 | 0.799 | 0.819 | 0.890 | 0.967 | 0.886 | 0.868 |
| (T7–6) mDPR (MS pFT + all FT)   | 0.900 | 0.955 | 0.841 | 0.856 | 0.860 | 0.813 | 0.785 | 0.843 | 0.876 | 0.966 | 0.883 | 0.871 |
| (a) monolingual DPR (in-lang FT) | 0.894 | – | 0.805 | 0.893 | 0.888 | – | 0.820 | – | – | – | – | – |
| (b) monolingual DPR (in-script FT) | – | – | 0.793 | 0.881 | 0.869 | – | – | – | – | – | – | – |
| (c) monolingual DPR (out-script FT) | 0.890 | – | 0.801 | 0.865 | 0.877 | – | 0.802 | – | – | – | – | – |
| (d) monolingual DPR (all FT)    | 0.890 | – | 0.814 | 0.867 | 0.890 | – | 0.802 | – | – | – | – | – |
| (e) English DPR (in-lang FT)    | 0.807 | 0.581 | 0.805 | 0.712 | 0.740 | 0.297 | 0.021 | 0.689 | 0.790 | 0.017 | 0.035 | 0.499 |
| (f) English DPR (MS pFT + in-lang FT) | 0.816 | 0.676 | 0.879 | 0.753 | 0.788 | 0.509 | 0.423 | 0.676 | 0.827 | 0.016 | 0.095 | 0.587 |
| (g) AfriBERTa DPR (in-lang FT)  | 0.738 | 0.451 | 0.565 | 0.658 | 0.692 | 0.635 | 0.416 | 0.293 | 0.808 | 0.872 | 0.657 | 0.617 |
| (h) BM25 + row (f)            | 0.874 | 0.946 | 0.867 | 0.830 | 0.908 | 0.723 | 0.640 | 0.795 | 0.872 | 0.813 | 0.831 | 0.829 |

(b) Recall@100

MRR@100 and Recall@100 of mDPR evaluated across all 11 languages and an overall average on the Mr. TyDi test set. “MS pFT”: pre–fine-tuning on MS MARCO; “in-lang FT”: fine-tuning on the Mr. TyDi training set from only the target language; “all FT”: fine-tuning on the Mr. TyDi training data from all languages; “in/out-script”: fine-tuning on training data from the target language and languages that are in the same/different scripts. Rows (T7–5) and (T7–6) are identical to rows (5) and (6) in Table 7.

number of other factors. Nevertheless, it is possible that there exist “better” monolingual BERT models for a target language, compared to mBERT. However, this must be balanced against the benefits of cross-lingual transfer obtained “for free” from using mBERT by leveraging datasets in other languages, for example, pre–fine-tuning on MS MARCO. Obviously, with a monolingual (non-English) BERT, it no longer makes sense to pre–fine-tune on MS MARCO. So, the research question boils down to this: What is more important, overall monolingual “in-language” quality or cross-lingual transfer effects from pre–fine-tuning using datasets in other languages?

The answer appears to be that monolingual models can be better, but not consistently so, and thus our recommendation is that using mBERT appears to be “safer.” These results are shown in Table 12, rows (a)–(d), where we trained DPR in five different languages (including English) starting from a monolingual backbone (described in Section 3.4). We are quick to point out that these models are pretrained by different groups and may vary in quality, which is dependent on the pretraining corpus, model size, hyperparameters, and a myriad of other issues.

Starting with an available monolingual BERT (in each of the target languages), we can fine-tune on just the in-language training data, shown in row (a); all in-script training data, shown in row (b); in-language and out-script training data, row (c); and all available data, row (d). The results show, not surprisingly, that with monolingual BERT, fine-tuning on anything other than in-language data appears to be pointless; in most cases it makes little difference, and in some cases it actually hurts. This makes sense, as a monolingual BERT model, by definition, is not designed to process input in other languages.

The monolingual BERT vs. mBERT comparison is shown by row (a) vs. row (T7–6), since with mBERT we can exploit pre–fine-tuning on MS MARCO and training data in other languages. We see that effectiveness is comparable, and Finnish is the only language where using a monolingual

ACM Transactions on Information Systems, Vol. 42, No. 2, Article 39. Publication date: September 2023.
model yields higher MRR@100. In row (f), more fully discussed in Section 8, we start with English BERT, apply pre–fine-tuning on MS MARCO and additional fine-tuning on English data. For English retrieval, this yields the highest effectiveness, since we are leveraging both a high-quality monolingual model and exploiting (in-language) pre–fine-tuning. The difference between row (f) and row (T7–5) can be attributed to mBERT vs. English BERT, and we can see that even in the best possible case, monolingual BERT is only marginally better than mBERT.

These findings support our recommendation to start with mBERT, which we view as the “safer” option and has the additional benefit of simplifying the operational aspects of model training (e.g., dealing with different tokenizers and different model sizes). Our results also show that for English retrieval, English BERT beats mBERT slightly, but this is not inconsistent with our recommendation, since our focus is on multilinguality.

8 HAVE DATA, NO MODEL

Definition. In Scenario (3), the target language $L \notin$ mBERT (i.e., we do not have a multilingual model that has been pretrained with the target language), but we do have data in the target language $L$.

Recommendations. Training a dense retrieval model with an English backbone, even for a non-English target language, can be effective. Similarly, training a dense retrieval model using a multilingual BERT, even one that does not include the target language, can also be effective. Even if these dense models are not effective on their own, they appear to provide valuable relevance signals that can improve over bag-of-words BM25 results.

Explanation. The core of our recommendations here can be characterized as “something is better than nothing,” and the results are quite surprising. Before proceeding, however, we should first defend this as a reasonable scenario: We do think that it is plausible to have training data in some low-resource language that is not covered by mBERT. In short, search existed before neural networks and pretrained transformers. It is certainly possible for some organization to build (or even have already built) a bag-of-words search engine in some low-resource language outside the set used to pretrained mBERT and then to use this search engine to gather relevance judgments (perhaps to train a simple learning-to-rank model). After all, mBERT only covers around 100 languages, and there are thousands of languages in the world we would potentially like to search in. The amount of relevance judgments used in our experiments (as low as around 1,000 examples) seems modest (see Table 2), and thus we argue that it is entirely plausible to have some training data in a target language not covered by mBERT.

What if we just start with English BERT and fine-tune on training data in the target language $L$, even though at face value this seems non-sensical, since the language might bear no similarity to English? These results are shown in Table 12, row (e), without MS MARCO pFT, and row (f), with MS MARCO pFT. We see that, for example, training English BERT with Indonesian relevance judgments yields a model that is quite “reasonable” for retrieval in Indonesian. We achieve an MRR@100 0.439, row (f), compared to mDPR (MS pFT + all FT), which achieves 0.579. Note that Indonesian and English are both written in the Latin script. However, even across scripts, we can build dense retrieval models that perform much better than random, e.g., Arabic and Bengali, but in other cases, e.g., Telugu and Thai, the results are pretty close to garbage. Interestingly, pre–fine-tuning English DPR on MS MARCO improves effectiveness across all languages except Telugu. This seems to indicate that, with a high-quality English DPR model, we can obtain at least a “reasonable” dense retrieval model for many languages.

An alternative solution for this scenario is to use a multilingual BERT that does not contain the target language $L$. Here we obviously cannot use mBERT, since all the languages in Mr. TYDI
are in mBERT’s pretraining corpus. Instead, we use AfriBERTa \cite{cite}, an mBERT-style multilingual
language model pretrained from scratch on 11 African languages. Hence, all the languages in
Mr. TYDI, except Swahili, are absent from its pretraining corpus. Results are shown in Table 12,
row (g); note that pre–fine-tuning AfriBERTa on English MS MARCO does not make sense,
because English is not one of the languages in the pretraining corpus. Surprisingly, AfriBERTa
obtains very “reasonable” results on most languages, despite never having been pretrained on
them. However, it seems like English BERT performs on par or better on all languages except
Telugu and Thai. From these results, we can see that a multilingual BERT model can be used to
bootstrap “reasonable” dense retrieval models in this scenario.

Our recommendation for scenario (3) is that if a pretrained model does not exist for the target
language, then simply using English BERT or another multilingual BERT can be helpful. However,
the astute reader might point out that in this scenario, dense retrieval models perform worse than
BM25, so what is the point of even using them? We address this potential objection by combining
the results in row (f) with bag-of-words BM25 results, shown in Table 12 as row (h). Here we see
that the hybrid dense retrieval + BM25 scores are generally better than BM25 results alone. Even in
the case of Telugu and Thai, the two cases where English BERT yields terrible low-quality output,
the hybrid results are just a tiny bit worse (te) or marginally better (th). Thus, it appears that even
if the models trained following our recommendations are not effective, they can still provide com-
plementary relevance signals to improve bag-of-words BM25 in a hybrid fusion approach. Thus,
“something is better than nothing.”

9 TOKEN OVERLAP AMONG MULTILINGUAL CORPORA

In this article, we have laid out a number of best practices for training multilingual dense retrieval
models for monolingual retrieval, particularly for low-resource languages. Our recommendations
are supported by empirical experiments with Mr. TYDI, using data from 11 typologically diverse
languages. To complement these recommendations, we attempt to explore the question of “why”
in this section.

It is usually assumed that corpora from different languages have very different vocabularies,
either in terms of naturally occurring words or algorithmically generated subwords from the to-
kenizers of pretrained multilingual transformers. We question this assumption and hypothesize
that shared vocabularies (especially after tokenization) contribute to the cross-lingual transfer ca-
pabilities of multilingual models. In this section, we refer to tokens generated by transformer to-
kenizers as \textit{subwords} and tokens that naturally exist in a corpus simply as \textit{words}. Our analysis of
overlapping vocabularies across different languages shows that the proportions of common words
and subwords are much higher than one might have expected. We speculate that this is naturally
caused by code-switching in Wikipedia articles and further enriched by the (relatively) compact
vocabulary space in current multilingual pretrained transformers.

9.1 Shared Word Vocabularies

In Wikipedia, named entities are additionally written in their “native” language when it is differ-
ent from the language of the corpus. This is most common in English articles but also appears
frequently in pairs of non-English languages. For example, the Korean article about Saladin, the
12th century sultan of the Ayyubid dynasty, contains his Arabic name in the first sentence of the
article. This means that naturally occurring corpora in one language contain tokens from other
languages, and we observe this occurring across all the Mr. TYDI languages. Trivially, all languages
share in the use of Arabic numerals as well.

We attempt to quantitatively characterize this phenomenon in the Mr. TYDI corpora. In Table 13,
we report the percentage of shared vocabularies of naturally occurring words, which illustrates
Table 13. Analysis of Unique Whitespace-delimited Tokens in Mr. TyDI

| ar  | bn  | en  | fi  | id  | ja  | ko  | ru  | sw  | te  | th  | Total |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------|
| 94.1% | 0.0%  | 2.5%  | 0.7%  | 1.0%  | 0.1%  | 0.5%  | 0.2%  | 0.9%  | 0.0%  | 0.0%  | 1,397,830 |
| 2.2% | 66.0%  | 14.2%  | 3.3%  | 5.7%  | 0.3%  | 3.8%  | 0.4%  | 3.9%  | 0.1%  | 0.1%  | 64,379 |
| 0.7% | 0.1%  | 37.4%  | 13.7%  | 21.7%  | 0.8%  | 6.0%  | 1.4%  | 17.9%  | 0.0%  | 0.3%  | 1,802,836 |
| 0.0% | 0.0%  | 3.4%  | 90.7%  | 3.6%  | 0.1%  | 0.1%  | 0.2%  | 2.0%  | 0.0%  | 0.0%  | 2,337,303 |
| 1.0% | 0.1%  | 16.6%  | 10.4%  | 51.3%  | 1.0%  | 3.9%  | 0.6%  | 15.1%  | 0.0%  | 0.2%  | 382,769 |
| –   | –   | –   | –   | –   | –   | –   | –   | –   | –   | –   | –   |
| ko  | 0.0%  | 0.0%  | 0.9%  | 0.2%  | 0.4%  | 0.3%  | 97.5%  | 0.1%  | 0.6%  | 0.0%  | 0.0%  | 4,192,486 |
| ru  | 0.1%  | 0.0%  | 2.6%  | 1.0%  | 1.2%  | 0.1%  | 0.5%  | 93.4%  | 1.1%  | 0.0%  | 0.0%  | 2,994,515 |
| sw  | 0.7%  | 0.0%  | 9.4%  | 6.6%  | 11.8%  | 0.0%  | 0.3%  | 0.3%  | 70.7%  | 0.0%  | 0.0%  | 119,526 |
| te  | 0.7%  | 0.2%  | 16.2%  | 3.5%  | 7.7%  | 0.1%  | 0.3%  | 0.3%  | 6.8%  | 64.3%  | 0.0%  | 65,950 |
| th  | 0.0%  | 0.0%  | 0.9%  | 0.2%  | 0.4%  | 0.2%  | 0.4%  | 0.4%  | 0.0%  | 0.0%  | 97.5%  | 2,470,418 |

Each row represents a different corpus; we omit Japanese here, because it is written without spaces. Each column indicates the percentage of words belonging to that language, based on an off-the-shelf language id package. The total number of unique words in each language is reported in the rightmost column. The values are highlighted row-wise: The color gradient ranges from blue (high values) to orange (low values).

The prominence of code-switching, especially surrounding English. For each language, we begin by tokenizing its corpus using whitespace and punctuation; we omit Japanese from this analysis, since it is written without spaces. We apply an off-the-shelf Python language id package called langdetect to determine the language of each unique token. These results are shown in Table 13, where each row represents the corpus in a language and each column indicates the identified language. Considering the first row, showing results for the Arabic corpus: We see that 94.1% of unique words are indeed identified as Arabic. However, there are non-trivial numbers of words that are identified as belonging to other languages as well. For example, the Arabic article on Isaac Newton contains his name in English. In total, we observe that about 2.5% of the unique words in the Arabic corpus are actually identified as English. Although the results in Table 13 are a bit noisy, because language id is imperfect and whitespace tokenization can be problematic for some cases, we have manually verified these results as “reasonable” via spot checking. Regardless, this analysis conclusively shows that code-switching naturally occurs in Wikipedia.

Looking at the English row, we see that English Wikipedia exhibits the greatest diversity in code-switching “out of English” into other languages, e.g., Saladin to Arabic, Peter the Great to Russian, and Tokugawa Ieyasu to Japanese. Looking at the English column, we see that English is the most common language that non-English corpora code-switch “into.” For example, the Wikipedia pages on Albert Einstein in non-English languages all mention his name in English. While English-centric cross-lingual anchors dominate, we see many examples that do not involve English: For example, the Arabic article about Tokugawa Ieyasu contains his name in Japanese, and the Russian article about the Chittoor district in India contains the native Telugu name. Furthermore, these pivots are, for the most part, named entities, which are often the topic of natural language questions.

We believe this provides at least an initial explanation of why pre–fine-tuning on MS MARCO data works so well, even in a zero-shot setting, e.g., Table 4, row (4). It seems reasonable to assume that mBERT has already “internalized” dictionaries of named entities across many languages as a natural by-product of MLM pretraining, given that the code-switching behavior described above

6https://github.com/Mimino666/langdetect
7For example, it can be difficult to identify the language of words in Latin-script languages without context.
8We confirmed via spot-checking that this is indeed the case by forcing mBERT to perform cross-lingual prediction of masked tokens, and indeed the model is able to generate correct named entity translations.
Table 14. Analysis of Unique Tokens Produced by Pretrained Transformer Tokenizers in Mr. TYDI

|      | ar  | bn  | en  | fi  | id  | ja  | ko  | ru  | sw  | te  | th  |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ar   | 100.0% | 92.3% | 62.2% | 82.3% | 81.3% | 78.2% | 81.9% | 66.8% | 94.2% | 92.2% | 90.5% |
| bn   | 57.3% | 100.0% | 39.1% | 54.8% | 54.7% | 52.1% | 57.2% | 43.8% | 72.0% | 80.3% | 67.8% |
| en   | 98.1% | 99.2% | 100.0% | 99.0% | 98.8% | 98.3% | 97.7% | 94.5% | 99.7% | 98.5% | 99.6% |
| fi   | 82.0% | 88.0% | 62.6% | 100.0% | 81.3% | 78.8% | 81.5% | 70.4% | 94.9% | 89.2% | 88.6% |
| id   | 87.7% | 95.1% | 67.6% | 88.0% | 100.0% | 84.9% | 88.1% | 73.6% | 97.0% | 94.1% | 95.2% |
| ja   | 88.3% | 94.8% | 70.6% | 89.3% | 88.8% | 100.0% | 92.5% | 76.5% | 94.6% | 94.1% | 95.0% |
| ko   | 79.4% | 89.4% | 60.0% | 79.2% | 79.2% | 79.4% | 100.0% | 65.9% | 91.5% | 88.3% | 91.5% |
| ru   | 92.2% | 94.9% | 80.6% | 93.0% | 91.7% | 91.1% | 91.4% | 100.0% | 97.8% | 94.2% | 92.6% |
| sw   | 57.7% | 71.1% | 38.8% | 58.4% | 55.1% | 52.3% | 57.9% | 44.6% | 100.0% | 75.2% | 67.2% |
| te   | 51.0% | 71.5% | 34.6% | 49.5% | 48.2% | 46.1% | 50.4% | 38.5% | 67.9% | 100.0% | 61.0% |
| th   | 71.6% | 86.5% | 50.0% | 70.4% | 69.9% | 67.6% | 74.7% | 56.6% | 86.8% | 87.3% | 100.0% |

(a) mBERT tokenizer

Each language cell in a column (L<sub>c</sub>) indicates the percentage of overlapping tokens between itself and the language cell in a row (L<sub>r</sub>) over the total number of unique tokens in L<sub>c</sub>. For example, the bn–ar entry in Table 14(a) is 57.3%, indicating that the number of overlapping subwords produced by the mBERT tokenizer for the ar and bn corpora accounts for 57.3% of the total unique subwords in the bn corpus. The values are highlighted columnwise: The color gradient ranges from blue (high values) to orange (low values).

(b) XLM-RLarge tokenizer

in essence provides parallel data. Wikipedia is part of the pretraining corpus for most pretrained models, and it is likely that code-switching is common as well in other genres of text in the pretraining corpus. Thus, when we fine-tune in English, these cross-language anchors enable the model to transfer relevance matching capabilities across languages. Since we have also demonstrated the existence of non-English cross-lingual anchors, this might begin to explain why fine-tuning on language K helps language L (where both K and L are not English). We further note that in QA applications, getting the named entity right is “half the battle” (if not more). Consider a question about the birthdate of a person; if the model simply retrieves passages about the correct person, then the results will already be “reasonable” (and may in fact contain the answer, just by sheer luck).

9.2 Shared Subword Vocabularies

We empirically observe that after tokenization by pretrained multilingual transformers, corpora from different languages share a large overlap in subwords, reported in Table 14. For example, consider Table 14(a), where the corpus in each language is tokenized using the mBERT tokenizer. To illustrate the information captured in the table, we provide two examples as follows:

- Overlapping subwords between Arabic and English represent 98.1% (row en, column ar) of subwords in the Arabic corpus and 62.2% (row ar, column en) in the English corpus.
• Overlapping subwords between te and th represent 87.3% (row th, column te) in Telugu and 61.0% (row te, column th) in Thai.

The numbers of shared subwords were a total surprise to us. In Table 14(b), where the tokenization is performed using XLM-R_{Large}, the overlap percentages are generally smaller but remain quite high. We can rule out data quality issues, because these results are consistent with the white-space tokenization analysis above; furthermore, these figures pass both sanity checks and spot checks.

The reality here is twofold: First, many shared subwords across languages are not from either language. That is, the overlap of subwords found in both the English and Arabic corpora (for example) comprises a large number of subwords from other languages. This makes sense given the results in Table 13, e.g., the English corpus code-switches into every other language (recall that our analysis is performed in terms of unique words.) Second, we observe that even a small number of words can produce a large number of subwords in the vocabulary space of that model. This is because the vocabulary space for all languages is rather constrained: The numbers of subwords in mBERT and XLM-R_{Large} are respectively 120K and 250K but shared by around 100 languages. For example, consider Table 13 row bn, column ar: The subwords produced by the mBERT tokenizer from the 2.2% of Arabic words in the Bengali corpus covers 57.3% of the Arabic subwords processed from the Arabic corpus, as seen in Table 14(a).

The upshot is that multilingual transformers are almost never presented with clean, “pure,” monolingual corpora. Hence, we expect the models to learn connections across multiple languages (using the cross-lingual anchors) and model the subwords jointly, thus enabling downstream cross-lingual transfer capabilities. Furthermore, subword overlap may explain why English BERT works well even when fine-tuned on other languages in Section 8. The English corpus already contains a non-trivial amount of subwords from other languages. We hypothesize that these shared subwords (which we show exist even in monolingual English BERT) provide anchors for downstream cross-lingual transfer. Indeed, this explanation is consistent with the literature [11, 13, 25, 29, 47, 58], where at least some researchers agree that higher subword overlap correlates with improved cross-lingual ability.

10 FUTURE WORK AND CONCLUSIONS

Practitioners today encounter complex situations when trying to build dense retrieval models for a specific (non-English) language, depending on the availability of training data, model backbones, compute resources, and other factors. In this article, we provide concrete guidance toward best practices for mDPR models, organized into three common scenarios. While we experimented with extensive settings, we acknowledge that our findings and recommendations are only based on evaluation with Mr. TYDI. Our ability to generalize is unfortunately limited by available data as well as computational costs, considering the scale of our numerous configurations. This may result in our guidance being biased toward application scenarios that are similar to Mr. TYDI, namely natural-language style queries on Wikipedia-based passages.

That said, our efforts comprehensively explore approaches for exploiting different sources of training data to enhance the cross-lingual transfer ability of dense retrieval models under different scenarios. Some of our recommendations are consistent with previous work (e.g., the beneficial effects of pre–fine-tuning), but we reach some surprising conclusions as well (e.g., cross-lingual transfer across languages in different scripts and the fact that language-specific knowledge learned prior to fine-tuning does not benefit mDPR as much when in-language, in-domain fine-tuning data are available). Further generalizations of our findings should be explored in future work, but we believe that our practical engineering guidance already provides substantial value.
In a separate thread of work, researchers today are still trying to understand why multilingual BERT exhibits such impressive capabilities. In contrast to most existing research on NLP tasks, our scientific contribution is to provide two possible explanations specifically for information retrieval: natural code-switching and shared vocabularies. While we have not provided a completely satisfactory explanation, these ideas provide a foundation that future researchers can build on.

ACKNOWLEDGMENTS

Computational resources were provided in part by Compute Ontario and Compute Canada. We acknowledge Cloud TPU support from Google’s TPU Research Cloud (TRC).

REFERENCES

[1] Jordi Armengol-Estapé, Casimiro Pio Carrino, Carlos Rodríguez-Penagos, Ona de Gibert Bonet, Carme Armentano-Oller, Aitor Gonzalez-Agirre, Maite Melero, and Marta Villegas. 2021. Are multilingual models the best choice for moderately under-resourced languages? A comprehensive assessment for Catalan. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. 4933–4946.

[2] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 4623–4637.

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

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

[5] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNa马拉, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A human generated MAchine reading comprehension dataset. arXiv:1611.09268. Retrieved from https://arxiv.org/abs/1611.09268

[6] Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19). 1538–1548.

[7] Luiz Henrique Bonifacio, Vitor Jeronimo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. mMARCO: A multilingual version of the MS MARCO passage ranking dataset. arXiv:2108.13897. Retrieved from https://arxiv.org/abs/2108.13897

[8] Leonid Boytsov, David Novak, Yury Malkov, and Eric Nyberg. 2016. Off the beaten path: Let’s replace term-based retrieval with k-NN search. In Proceedings of 25th International Conference on Information and Knowledge Management. 1099–1108.

[9] 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. Trans. Assoc. Comput. Ling. 8 (2020), 454–470.

[10] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 8440–8451.

[11] Alexis Conneau, Shijie Wu, Haozhi Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 6022–6034.

[12] 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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 4171–4186.

[13] Philipp Dufert and Hinrich Schütze. 2020. Identifying elements essential for BERT’s multilinguality. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20). 4423–4437.

[14] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. arXiv:2109.10086. Retrieved from https://arxiv.org/abs/2109.10086

[15] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. 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. 2353–2359.
Toward Best Practices for Training Multilingual Dense Retrieval Models

[16] Petra Galuščáková, Douglas W. Oard, and Suraj Nair. 2021. Cross-language information retrieval. arXiv:2111.05988. Retrieved from https://arxiv.org/abs/2111.05988

[17] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Tevatron: An efficient and flexible toolkit for neural retrieval. In Proceedings of the 46th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 3120–3124.

[18] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection. In Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI’20). 7780–7788.

[19] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz. Bellagay, Doug Downey, and Noah A. Smith. 2020. Don’t stop pretraining: Adapt language models to domains and tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 8342–8360.

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv:1503.02531. Retrieved from https://arxiv.org/abs/1503.02531

[21] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. arXiv:2010.02666. Retrieved from https://arxiv.org/abs/2010.02666

[22] 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. https://jmlr.org/tmlr/papers/bib/jKN1pXi7b0.bib

[23] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. Trans. Mach. Learn. Res. (2022).

[24] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. IEEE Trans. Big Data 7, 3 (2021), 535–547.

[25] Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: An empirical study. In Proceedings of the 8th International Conference on Learning Representations (ICLR’20). https://openreview.net/forum?id=HJeT3yrtDr

[26] 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 Conference on Empirical Methods in Natural Language Processing (EMNLP’20). 6769–6781.

[27] 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. Trans. Assoc. Comput. Ling. 7 (2019), 452–466.

[28] Carlos Lassance. 2023. Extending English IR methods to multi-lingual IR. arXiv:2302.14723. Retrieved from https://arxiv.org/abs/2302.14723

[29] Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP’20). 4483–4499.

[30] Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. CITADEL: Conditional token interaction via dynamic lexical routing for efficient and effective multi-vector retrieval. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 11891–11907.

[31] Jimmy Lin, David Alfonso-Hermelo, Vitor Jeronimo, Ehsan Kamalloo, Carlos Lassance, Rodrigo Nogueira, Odunayo Ogundepo, Mehdi Rezagholizadeh, Nandan Thakur, Jheng-Hong Yang, and Xinyu Zhang. 2023. Simple yet effective neural ranking and reranking baselines for cross-lingual information retrieval. arXiv:2304.01019. Retrieved from https://arxiv.org/abs/2304.01019

[32] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. Morgan & Claypool Publishers.

[33] 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 (RePLaNLP’21). 163–173.

[34] Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2021. Evaluating multilingual text encoders for unsupervised cross-lingual retrieval. In Proceedings of the 43rd European Conference on Information Retrieval. 342–358.

[35] 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. Retrieved from https://arxiv.org/abs/1907.11692
[36] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. B-PROP: Bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 1513–1522.

[37] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. PROP: Pre-training with representative words prediction for Ad-hoc retrieval. Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 283–291.

[38] Sean MacAvaney, Luca Soldaini, and Nazli Goharian. 2020. Teaching a new dog old tricks: Resurrecting multilingual retrieval using zero-shot learning. In Proceedings of the 42nd European Conference on Information Retrieval, Part II. 246–254.

[39] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: A tasty french language model. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 7203–7219.

[40] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. 9844–9855.

[41] Jian-Yun Nie. 2010. Cross-Language Information Retrieval. Morgan & Claypool.

[42] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. arXiv:1901.04085. Retrieved from https://arxiv.org/abs/1901.04085

[43] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. arXiv:1910.14424. Retrieved from https://arxiv.org/abs/1910.14424

[44] Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? No problem! Exploring the viability of pretrained multilingual language models for low-resource languages. In Proceedings of the 1st Workshop on Multilingual Representation Learning. 116–126.

[45] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Ashwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 46–54.

[46] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20). 7654–7673.

[47] Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 4996–5001.

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

[49] Subendhu Rongali, Abhyuday Jagamnatha, Bhanu Pratap Singh Rawat, and Hong Yu. 2020. Continual domain-tuning for pretrained language models. arXiv:2004.02288. Retrieved from https://arxiv.org/abs/2004.02288

[50] Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. Is multilingual BERT fluent in language generation? In Proceedings of the 1st NLPL Workshop on Deep Learning for Natural Language Processing. 29–36.

[51] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. CoIBERTv2: Effective and efficient retrieval via lightweight late interaction. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 3715–3734.

[52] 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. 2768–2773.

[53] Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2021. Cross-lingual training of dense retrievers for document retrieval. In Proceedings of the 1st Workshop on Multilingual Representation Learning. 251–253.

[54] 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 Conference on Empirical Methods in Natural Language Processing (EMNLP'20). 4160–4170.

[55] AnTi Virtanen, Jenna Kanerva, Rami Ilo, Jouini Luoma, Juhan Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for finnish. arXiv:1912.07076. Retrieved from https://arxiv.org/abs/1912.07076

[56] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In Proceedings of the 12th Language Resources and Evaluation Conference. 4003–4012.

[57] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020.
Toward Best Practices for Training Multilingual Dense Retrieval Models

Transformers: State-of-the-art natural language processing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 38–45.

[58] Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP’19). 833–844.

[59] Yuqing Xie, Wei Yang, Luchen Tan, Kun Xiong, Nicholas Jing Yuan, Baoxing Huai, Ming Li, and Jimmy Lin. 2020. Distant supervision for multi-stage fine-tuning in retrieval-based question answering. In Proceedings of the Web Conference. 2934–2940.

[60] 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 Proceedings of the 9th International Conference on Learning Representations (ICLR’21). https://openreview.net/forum?id=zeFrfgyZln

[61] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. Trans. Assoc. Comput. Ling. 10 (2022), 291–306.

[62] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 483–498.

[63] Jingtao Zhan, Qingyao Ai, Yiqun Liu, Jiaxin Mao, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Disentangled modeling of domain and relevance for adaptable dense retrieval. arXiv:2208.05753. Retrieved from https://arxiv.org/abs/2208.05753

[64] Rui Zhang, Caitlin Westerfield, Sungrok Shim, Garrett Bingham, Alexander Fabbri, William Hu, Neha Verma, and Dragomir Radev. 2019. Improving low-resource cross-lingual document retrieval by reranking with deep bilingual representations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 3173–3179.

[65] 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. 127–137.

[66] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamaloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2022. Making a MIRACL: Multilingual information retrieval across a continuum of languages. arXiv:2210.09984. Retrieved from https://arxiv.org/abs/2210.09984

[67] Xinyu Zhang, Andrew Yates, and Jimmy Lin. 2021. Comparing score aggregation approaches for document retrieval with pretrained transformers. In Proceedings of the 43rd European Conference on Information Retrieval, Part II. 150–163.

Received 24 August 2022; revised 26 April 2023; accepted 9 July 2023.