ColBERT-XM: A Modular Multi-Vector Representation Model for Zero-Shot Multilingual Information Retrieval

Antoine Louis, Vageesh Saxena, Gijs van Dijck, Gerasimos Spanakis
Maastricht University, Netherlands
{a.louis, v.saxena, gijs.vandijck, jerry.spanakis}@maastrichtuniversity.nl

Abstract

State-of-the-art neural retrievers predominantly focus on high-resource languages like English, which impedes their adoption in retrieval scenarios involving other languages. Current approaches circumvent the lack of high-quality labeled data in non-English languages by leveraging multilingual pretrained language models capable of cross-lingual transfer. However, these models require substantial task-specific fine-tuning across multiple languages, often perform poorly in languages with minimal representation in the pretraining corpus, and struggle to incorporate new languages after the pretraining phase. In this work, we present a novel modular dense retrieval model that learns from the rich data of a single high-resource language and effectively zero-shot transfers to a wide array of languages, thereby eliminating the need for language-specific labeled data. Our model, ColBERT-XM, demonstrates competitive performance against existing state-of-the-art multilingual retrievers trained on more extensive datasets in various languages. Further analysis reveals that our modular approach is highly data-efficient, effectively adapts to out-of-distribution data, and significantly reduces energy consumption and carbon emissions. By demonstrating its proficiency in zero-shot scenarios, ColBERT-XM marks a shift towards more sustainable and inclusive retrieval systems, enabling effective information accessibility in numerous languages. We publicly release our code and models for the community.

1 Introduction

Text retrieval models are integral to various day-to-day applications, including search, recommendation, summarization, and question answering. In recent years, transformer-based models have monopolized textual information retrieval and led to significant progress in the field (Lin et al., 2021). However, the existing literature mostly focuses on improving retrieval effectiveness in a handful of widely spoken languages – notably English (Muennighoff et al., 2023) and Chinese (Xiao et al., 2023) – whereas other languages receive limited attention.

As a solution, a few studies have suggested fine-tuning multilingual transformer-based encoders, such as mBERT (Devlin et al., 2019), on aggregated retrieval data across various languages. Nonetheless, this approach faces two major challenges. First, acquiring high-quality relevance labels for various languages proves difficult, particularly for those with fewer resources. Consequently, languages with insufficient representation in training data experience a proficiency gap compared to widely represented ones (MacAvaney et al., 2020). Second, when these multilingual transformers are pretrained on too many languages, their performance on downstream tasks worsens. This issue, known as the *curse of multilinguality* (Conneau et al., 2020), underscores the challenge of developing models that effectively accommodate a broader spectrum of languages.

Our work addresses the challenges above by introducing ColBERT-XM, a novel multilingual dense retrieval model built upon the recent XMOD architecture (Pfeiffer et al., 2022), which combines shared and language-specific parameters pretrained from the start to support the following key features:

1. *Reduced dependency on multilingual data*: Our XMOD-based retriever is designed to learn through monolingual fine-tuning, capitalizing on the rich data from a high-resource language, like English, thereby reducing the need for extensive multilingual datasets.
2. *Zero-shot transfer across languages*: Despite being fine-tuned in a single language, our retriever's modular components enable effective knowledge transfer to a variety of underrepresented languages without any further training.
3. *Significant reduction in carbon footprint*: Trained on just 6.4M English examples, our model contrasts with higher-consuming SoTA

Proceedings of the 31st International Conference on Computational Linguistics, pages 4370–4383
January 19–24, 2025. ©2025 Association for Computational Linguistics

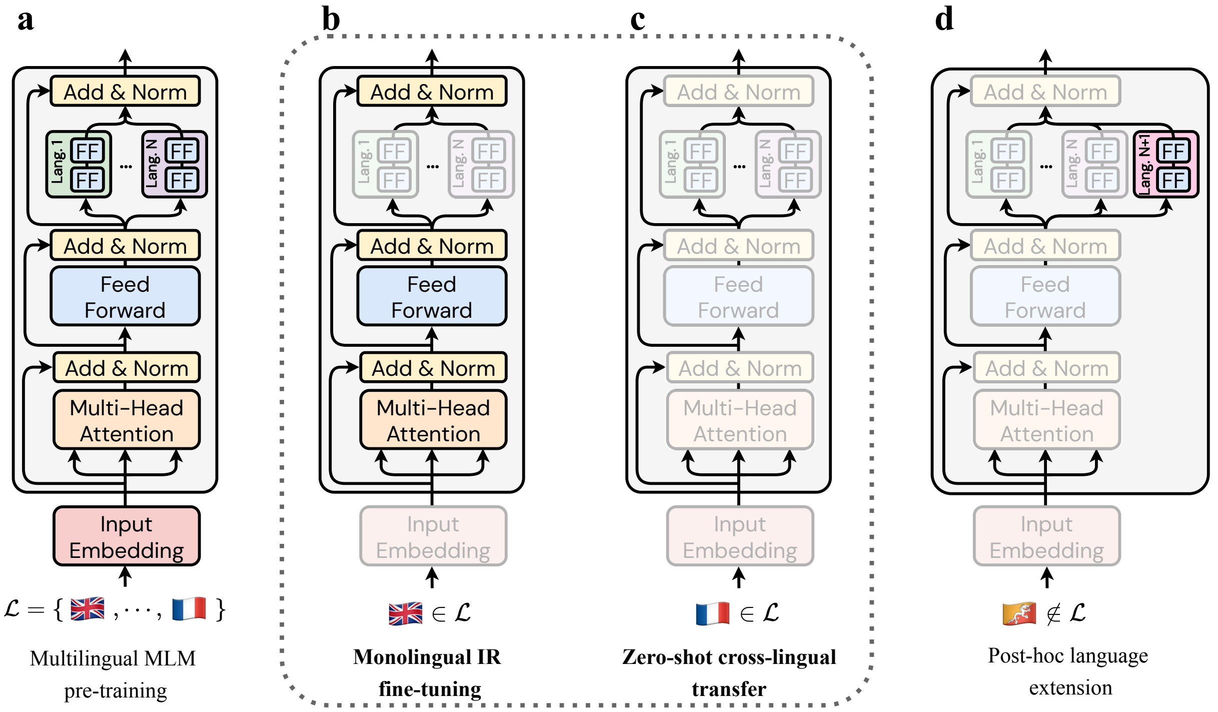

Figure 1: A high-level illustration of XMOD's modular layer during its successive learning stages. Components that are blurred indicate that they remain frozen. **(a)** Initially, the model learns language-specific adapters at each transformer layer via MLM pre-training on a large multilingual corpus. **(b)** We then adapt the pre-trained model to the downstream retrieval task by fine-tuning its shared weights on a high-resource language, while keeping the modular adapters and the embedding layer frozen. **(c)** Next, we perform zero-shot transfer by routing the target language's input text through the corresponding modular units. **(d)** Incidentally, the architecture supports adding new languages post training by learning extra modular adapters only via light MLM training in the new language.

networks and exemplifies environmental responsibility by dramatically lowering energy consumption and carbon emissions.

Practically, ColBERT-XM learns to effectively predict relevance between queries and passages using only a limited set of English examples, leveraging the late interaction approach introduced in ColBERT (Khattab and Zaharia, 2020). Our experimental results demonstrate competitive performance across a diverse range of languages against state-of-the-art multilingual models trained on vastly larger datasets and many more languages. Moreover, our analysis shows that ColBERT-XM is highly data-efficient, as more training data from the same distribution does not markedly enhance its performance. Even so, further investigations reveal our model's strong ability to generalize to out-of-distribution data, despite its limited training. We also provide evidence that multi-vector representations outperform single-vector approaches within our framework. Finally, we underscore our model's sustainability by examining its environmental impact in comparison to established dense retrievers.

In summary, the contributions of this research are threefold. First, we introduce a novel modular dense retriever that, despite being trained exclusively in one language, demonstrates remarkable adaptability to a broad spectrum of languages in a zero-shot configuration. Second, through comprehensive experiments, we compare the effectiveness of employing multi-vector over single-vector representations, explore the influence of the volume of training examples on the overall model's performance, investigate the model's ability to adapt to out-of-distribution data and languages it has not previously encountered, including low-resource ones, and highlight its sustainable environmental footprint. Finally, we release our source code and model checkpoints at `https://github.com/ant-louis/xm-retrievers`.

2 Related Work

2.1 Multilingual Information Retrieval

The term "multilingual" typically encompasses a wide range of retrieval tasks using one or more languages (Hull and Grefenstette, 1996). In our study, we define it as performing monolingual retrieval across multiple languages.

Monolingual text retrieval approaches have relied on simple statistical metrics based on term frequency, such as TF-IDF and BM25 (Robertson et al., 1994), to represent texts and match documents against a given query. With the advent of transformer-based language models, contextualized representations rapidly got incorporated into retrieval models and gave rise to various neural-based retrieval techniques, including cross-encoder models such as monoBERT (Nogueira et al., 2019) and monoT5 (Nogueira et al., 2020), single-vector bi-encoders like DPR (Karpukhin et al., 2020) and ANCE (Xiong et al., 2021), multi-vector bi-encoders like ColBERT (Khattab and Zaharia, 2020) and XTR (Lee et al., 2023), and sparse neural models such as uniCOIL (Lin and Ma, 2021) and SPLADE (Formal et al., 2021).

Nevertheless, prior work on neural retrievers has predominantly focused on English due to the abundance of labeled training data. In non-English settings, multilingual pretrained language models such as XLM-R (Conneau et al., 2020) and mBERT (Devlin et al., 2019) emerged as an effective solution, capable of adapting the retrieval task across many languages using a shared model (Lawrie et al., 2023). However, these models proved to suffer from the curse of multilinguality (Chang et al., 2023), have shown substantially reduced monolingual abilities for low-resource languages with smaller pretraining data (Wu and Dredze, 2020), and do not effectively extend to unseen languages after the pretraining phase (Pfeiffer et al., 2022).

2.2 Modular Transformers

Traditionally, adapting pretrained transformer-based language models to new data settings involves fully fine-tuning all pretrained weights on relevant data. While effective, this process is computationally expensive. As a parameter-efficient alternative, recent works have proposed inserting lightweight "expert" modules after each transformer layer (Houlsby et al., 2019) to capture specific modeling aspects, such as language-specific (Pfeiffer et al., 2020; Ansell et al., 2021) or task-specific (Bapna and Firat, 2019; He et al., 2021) knowledge. These modular components, commonly referred to as adapters (Rebuffi et al., 2017), are selectively fine-tuned for the downstream task, the core transformer parameters remaining frozen.

Despite their growing use in NLP, adapter-based approaches remain relatively untouched in multilingual information retrieval, with existing IR research primarily concentrating on cross-language retrieval (Litschko et al., 2022; Yang et al., 2022b), which aims to return documents in a language different from the query. A key limitation of these works is that the additional capacity introduced by adapters *after* pretraining is not able to mitigate the curse of multilinguality that has already had a catastrophic impact on the shared transformer weights (Pfeiffer et al., 2022). In contrast, our method employs a model inherently designed for modularity that learns language-specific capacity *during* pretraining, effectively avoiding this limitation.

3 Method

We present a novel multilingual dense retriever that learns to predict relevance between queries and passages via monolingual fine-tuning, while adapting to various languages in a zero-shot configuration. Our model, ColBERT-XM, adopts a traditional bi-encoder architecture (§3.1) based on a modular multilingual text encoder (§3.2), and employs the MaxSim-based late interaction mechanism (§3.3) for relevance assessment. The model is optimized through a contrastive learning strategy (§3.4), and uses a residual compression approach to significantly reduce the space footprint of indexes utilized for fast vector-similarity search at inference time (§3.5). We describe each part in detail below.

3.1 Bi-Encoder Architecture

To predict relevance between query q and passage p, ColBERT-XM uses the popular bi-encoder architecture (Gillick et al., 2018), which consists of two learnable text encoding functions $f(\cdot;\gamma_i) : \mathcal{W}^n \mapsto \mathbb{R}^{n\times d}$, parameterized by γ_i, that map input text sequences of n terms from vocabulary $\mathcal{W}$ to d-dimensional real-valued term vectors, i.e.,

$$\begin{aligned}\hat{\mathbf{H}}_q &= f([q_1, q_2, \cdots, q_i]; \gamma_1)\,, \text{and} \\ \hat{\mathbf{H}}_p &= f([p_1, p_2, \cdots, p_j]; \gamma_2)\,.\end{aligned} \tag{1}$$

The main idea behind this architecture is to find values for parameters γ_i such that a straightforward similarity function $\mathrm{sim} : \mathbb{R}^{n\times d} \times \mathbb{R}^{m\times d} \mapsto \mathbb{R}_+$ approximates the semantic relevance between q and p by operating on their bags of contextualized term embeddings, i.e.,

$$\mathrm{score}(q,p) = \mathrm{sim}\Big(\hat{\mathbf{H}}_q, \hat{\mathbf{H}}_p\Big)\,. \tag{2}$$

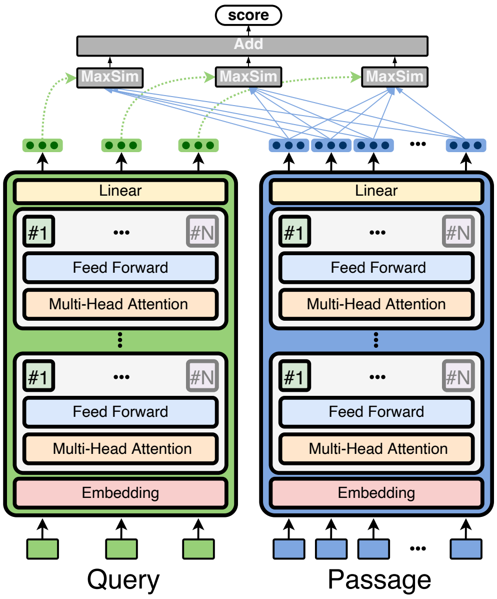

Figure 2: Illustration of the multi-vector late interaction paradigm used in our proposed ColBERT-XM model.

This scoring approach, known as *late interaction* (Khattab and Zaharia, 2020), as interactions between the query and passage are delayed after their independent encoding computations, stands out for its computational efficiency (Reimers and Gurevych, 2019). This contrasts with the popular cross-encoder architecture (Nogueira et al., 2019), which encodes the queries and passages jointly to learn rich interactions directly within the model.

In this work, we use a *siamese* bi-encoder, where queries and passages are encoded by two identical copies of a shared network (i.e., $\gamma_1 = \gamma_2$).

3.2 Modular Language Representation

To overcome the limitations posed by multilingual transformer-based encoders outlined in Section 1, we use the XMOD model (Pfeiffer et al., 2022) as our backbone text encoder. As depicted in Figure 1a, XMOD extends the transformer architecture by incorporating language-specific adapters (Houlsby et al., 2019) at every transformer layer, which are learned from the start *during* the masked language modeling (MLM) pretraining phase. This method contrasts with conventional adapter-based approaches that typically extend pretrained multilingual models post-pretraining, thereby building upon sub-optimal parameter initialization already affected by the curse of multilinguality.

Formally, our modular language representation model is defined as a learnable encoding function $g(\cdot;\boldsymbol{\theta},\boldsymbol{\phi}_i) : (\mathcal{W}^k,\mathcal{L}) \mapsto \mathbb{R}^{k\times d}$, with shared parameters $\boldsymbol{\theta}$ and language-specific parameters $\boldsymbol{\phi}_i$, that maps a text sequence t of k terms from vocabulary $\mathcal{W}$ in language $\mathcal{L}_i$ to d-dimensional real-valued representations. Let $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d\times d_{\text{out}}}$ be a linear layer with no activations designed to compress the dimensions of the output representation vectors, Equation (1) then becomes

$$\begin{aligned}\hat{\mathbf{H}}_t &= g([t_1,\cdots,t_k];\boldsymbol{\theta},\boldsymbol{\phi}_i)\cdot\mathbf{W}_{\text{out}}\\ &= \left[\hat{\boldsymbol{h}}_1^t,\hat{\boldsymbol{h}}_2^t,\cdots,\hat{\boldsymbol{h}}_k^t\right].\end{aligned} \quad (3)$$

A key benefit of employing XMOD over traditional multilingual transformers is its proven adaptability to accommodate new languages after the initial pretraining phase while maintaining performance across previously included languages, thereby effectively counteracting the curse of multilinguality. Furthermore, Pfeiffer et al. (2022) demonstrated that the per-language performance remains consistent whether a language is included during pretraining or added afterward. This suggests that XMOD can potentially encompass numerous languages by pretraining on a subset of languages for which sufficient text data exists, and subsequently adapting to additional, underrepresented languages without deteriorating overall performance. As shown in Figure 1d, the post-hoc inclusion of a new language involves learning extra language-specific adapters only via MLM training. While beyond the scope of our study, we hope to explore this characteristic within the retrieval context in future work.

3.3 MaxSim-based Late Interaction

ColBERT-XM adopts the fine-granular late interaction scoring mechanism of ColBERT, depicted in Figure 2. This mechanism calculates the cosine similarity across all pairs of query and passage embeddings, applies max-pooling across the resulting similarity scores for each query term, and then sum the maximum values across query terms to derive the overall relevance estimate, i.e.,

$$\operatorname{sim}\left(\hat{\mathbf{H}}_{\tilde{q}},\hat{\mathbf{H}}_{\tilde{p}}\right) = \sum_{i=1}^{n}\max_{j=1}^{m}\cos\left(\hat{\boldsymbol{h}}_i^{\tilde{q}},\hat{\boldsymbol{h}}_j^{\tilde{p}}\right), \quad (4)$$

where $\tilde{q}$ and $\tilde{p}$ correspond to sequences obtained after incorporating special tokens into q and p, respectively, and truncating to preset maximum lengths n and m. More specifically, we have

$$\begin{aligned}\tilde{p} &= [[\texttt{CLS}],[\texttt{P}],p_1,\cdots,p_j]\text{ , and}\\ \tilde{q} &= [[\texttt{CLS}],[\texttt{Q}],q_1,\cdots,q_i,[\texttt{M}],\cdots,[\texttt{M}]]\text{ ,}\end{aligned} \quad (5)$$

where [M] is a mask token appended to queries to reach the predefined length n. This padding strategy serves as a query augmentation technique, enhancing the model's ability to interpret short queries through the generation of extra contextualized embeddings at the mask positions. The special tokens [P] and [Q] enable the shared XMOD-based encoder to differentiate between passage and query input sequences, respectively.

3.4 Supervision

Let $\mathcal{B} = \{(q_i, p_i^+, p_{\text{H},i}^-)\}_{i=1}^N$ be a batch of N training instances, each comprising a query q_i associated with a positive passage p_i^+ and a hard negative passage $p_{\text{H},i}^-$. By considering the passages paired with all other queries within the same batch, we can enrich each training triple with an additional set of $2(N-1)$ *in-batch* negatives $\mathcal{P}_{\text{IB},i}^- = \{p_j^+, p_{\text{H},j}^-\}_{j\neq i}^N$. Given these augmented training samples, we optimize our model using a contrastive learning strategy that combines two established ranking loss functions, expressed as

$$\mathcal{L}_{\text{TOTAL}}\left(q_i, p_i^+, p_{\text{H},i}^-, \mathcal{P}_{\text{IB},i}^-\right) = \mathcal{L}_{\text{PAIR}} + \mathcal{L}_{\text{IB}} \qquad (6)$$

where $\mathcal{L}_{\text{PAIR}}$ is the pairwise softmax cross-entropy loss computed over predicted scores for the positive and hard negative passages, used in ColBERTv1 (Khattab and Zaharia, 2020) and defined as

$$\mathcal{L}_{\text{PAIR}} = -\log \frac{e^{\text{score}(q_i,p_i^+)}}{e^{\text{score}(q_i,p_i^+)} + e^{\text{score}(q_i,p_{\text{H},i}^-)}}, \qquad (7)$$

while $\mathcal{L}_{\text{IB}}$ is the in-batch sampled softmax cross-entropy loss added as an enhancement for optimizing ColBERTv2 (Santhanam et al., 2022b):

$$\mathcal{L}_{\text{IB}} = -\log \frac{e^{\text{score}(q_i,p_i^+)}}{\sum_{p\in\mathcal{P}_{\text{IB},i}^-\cup\{p_i^+,p_{\text{H},i}^-\}} e^{\text{score}(q_i,p)}}. \qquad (8)$$

These contrastive losses aim to learn a high-quality embedding function so that relevant query-passage pairs achieve higher similarity than irrelevant ones.

3.5 Inference

Since passages and queries are encoded independently, passage embeddings can be precomputed and indexed offline through efficient vector-similarity search data structures, using the faiss library (Johnson et al., 2021). Instead of directly indexing the passage representations as in ColBERTv1, which requires substantial storage even when compressed to 32 or 16 bits, we adopt the centroid-based indexing approach introduced in ColBERTv2, as detailed in Appendix A.

4 Experiments

4.1 Experimental Setup

Data. For training, we follow ColBERTv1 and use triples from the MS MARCO passage ranking dataset (Nguyen et al., 2018), which contains 8.8M passages and 539K training queries. However, unlike the original work that uses the BM25 negatives provided by the official dataset, we sample harder negatives mined from 12 distinct dense retrievers.[1] For a comprehensive evaluation across various languages, we consider the small development sets from mMARCO (Bonifacio et al., 2021), a machine-translated variant of MS MARCO in 13 languages, each comprising 6980 queries. To assess out-of-distribution performance, we use the test sets from Mr. TyDi (Zhang et al., 2021), another multilingual open retrieval dataset including low-resource languages not present in mMARCO.

Implementation. We train our model for 50k steps using the AdamW optimizer (Loshchilov and Hutter, 2017) with a batch size of 128, a peak learning rate of 3e-6 with warm up along the first 10% of training steps and linear scheduling. We set the embedding dimension to d_{out}=128, and fix the maximum sequence lengths for questions and passages at n=32 and m=256, respectively. Training is performed on a single 80 GB NVIDIA H100 GPU hosted on a Linux server with two 3.20 GHz AMD EPYC 7763 CPUs and 500 GB of RAM. We use the following Python libraries: transformers (Wolf et al., 2020), sentence-transformers (Reimers and Gurevych, 2019), colbert-ir (Khattab and Zaharia, 2020), and wandb (Biewald, 2020).

Metrics & evaluation. To measure effectiveness, we use the official metrics for each query set, i.e., mean reciprocal rank at cut-off 10 (MRR@10) for MS MARCO, and recall at cut-off 100 (R@100) along MRR@100 for Mr. TyDi. We compare our model against established multilingual baselines spanning four retrieval methodologies. For lexical matching, we report the widely adopted bag-of-words BM25 function (Robertson et al., 1994). For the cross-encoders, we include two classification models based on mMiniLM$_{\text{L6}}$ (Wang et al., 2021) and mT5$_{\text{BASE}}$ (Xue et al., 2021), each fine-tuned on mMARCO pairs across 9 languages (Bonifacio et al., 2021). The dense single-vector bi-encoders

[1] https://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives

Model	# Training			*en*	*es*	*fr*	*it*	*pt*	*id*	*de*	*ru*	*zh*	*ja*	*nl*	*vi*	*hi*	*ar*	Avg
	Examp.	Lang.	Param.															
Lexical systems																		
1 BM25 (Pyserini)	-	-	-	18.4	15.8	15.5	15.3	15.2	14.9	13.6	12.4	11.6	14.1	14.0	13.6	13.4	11.1	14.2
Cross-encoders																		
2 mT5$_{\text{BASE}}$ (Bonifacio et al., 2021)	12.8M	9	390M	36.6	31.4	30.2	30.3	30.2	29.8	28.9	26.3	24.9	26.7	29.2	25.6	26.6	23.5	28.6
3 mMiniLM$_{\text{L6}}$ (Bonifacio et al., 2021)	80.0M	9	107M	36.6	30.9	29.6	29.1	28.9	29.3	27.8	25.1	24.9	26.3	27.6	24.7	26.2	21.9	27.8
Single-vector dense bi-encoders																		
4 DPR-X (Yang et al., 2022a)	25.6M	4	550M	24.5	19.6	18.9	18.3	19.0	16.9	18.2	17.7	14.8	15.4	18.5	15.1	15.4	12.9	17.5
5 mE5$_{\text{BASE}}$ (Wang et al., 2024)	981.6M	16	278M	35.0	28.9	30.3	28.0	27.5	26.1	27.1	24.5	22.9	25.0	27.3	23.9	24.2	20.5	26.5
Multi-vector dense bi-encoders																		
6 mColBERT (Bonifacio et al., 2021)	25.6M	9	167M	35.2	30.1	28.9	29.2	29.2	27.5	28.1	25.0	24.6	23.6	27.3	18.0	23.2	20.9	26.5
7 ColBERT-XM (ours)	6.4M	1	86M	37.2	28.5	26.9	26.5	27.6	26.3	27.0	25.1	24.6	24.1	27.5	22.6	23.8	19.5	26.2

Table 1: MRR@10 results on mMARCO small dev set. Performance on languages encountered during fine-tuning is highlighted in orange, whereas zero-shot performance is highlighted in blue. ColBERT-XM reaches near state-of-the-art results while trained on one language only with much fewer examples than competitive models.

are derived from XLM-R (Conneau et al., 2020) and have been fine-tuned on samples in 4 (Yang et al., 2022a) and 16 languages (Wang et al., 2024), respectively. Lastly, we report the performance of a dense multi-vector bi-encoder built on mBERT$_{\text{BASE}}$ (Devlin et al., 2019) and fine-tuned on mMARCO samples across 9 languages (Bonifacio et al., 2021).

4.2 Main Results

Table 1 reports results using the official MRR@10 metric for the 14 languages included in mMARCO. In its training language (i.e. English), ColBERT-XM outperforms all multilingual baselines. The underperformance of certain models, like mT5$_{\text{BASE}}$ and mColBERT, can partly be attributed to their exposure to fewer English examples given their training across 9 languages with 12.8M and 25.6M samples distributed evenly – resulting in only 1.4M and 2.8M English examples, respectively, compared to ColBERT-XM's 6.4M training set. Conversely, models such as mMiniLM$_{\text{L6}}$ and mE5$_{\text{BASE}}$, despite being exposed to a larger number of English examples, still underperform, suggesting that the modular architecture of ColBERT-XM may offer intrinsic benefits over conventional multilingual models.

In languages on which ColBERT-XM was not trained but the baselines were, we observe comparable performance. For instance, when excluding English, the difference in average performance between our model and mE5$_{\text{BASE}}$ is merely 0.5%, even though mE5$_{\text{BASE}}$ was trained in 15 additional languages and 800,000 times more data samples. In languages on which neither ColBERT-XM nor the baselines were trained, we note a slight enhancement in performance among the computationally expensive cross-encoder models, while both the non-modular single-vector and multi-vector bi-encoders lag behind our model in performance.

Overall, ColBERT-XM demonstrates strong knowledge transfer and generalization capabilities across languages while trained on a significantly smaller monolingual set.

4.3 Further Analysis

In this section, we conduct a thorough analysis of several key aspects of our proposed methodology, including the influence of greater volumes of training data on ColBERT-XM's performance (§4.3.1), a performance comparison with a modular single-vector representation variant (§4.3.2), the model's ability to generalize to out-of-distribution data (§4.3.3), and its environmental footprint compared to existing multilingual retrievers (§4.3.4).

4.3.1 How does training on more examples affect ColBERT-XM performance?

Despite being trained on substantially fewer examples, ColBERT-XM demonstrates competitive results compared to existing multilingual models, raising the question of whether an increased volume of training data would further enhance its performance. To investigate, we train five instances of our modular retriever on a varying number of MS MARCO training triples, namely 3.2M, 6.4M, 12.8M, 19.2M, and 25.6M examples. Figure 3 shows the resulting models' performance on the mMARCO small dev set across MRR@10 and recall at various cut-offs, alongside the fixed performance of mE5$_{\text{BASE}}$ for comparison. The results reveal an initial performance boost with an increase in training data, which plateaus quickly after 6.4M examples, suggesting diminishing returns from additional data of the same distribution. This contrasts with existing baselines that were trained on comparatively more samples from diverse languages to reach their peak performance, thereby underscoring ColBERT-XM's efficiency

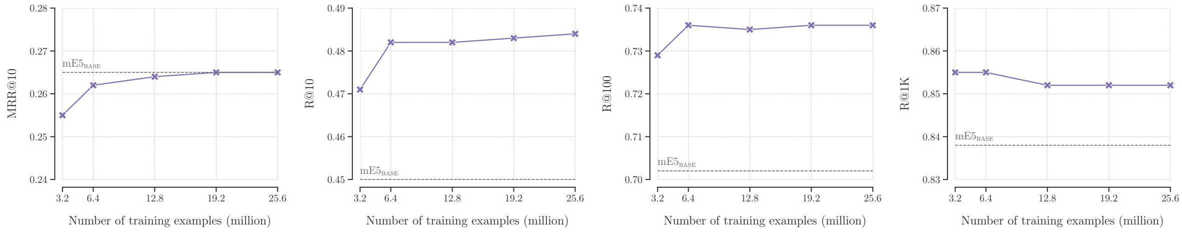

Figure 3: Performance of ColBERT-XM on mMARCO small dev set, based on the volume of training examples.

in low-resource scenarios. For a comprehensive breakdown of performance across individual languages, we refer to Table 4 in Appendix B.

4.3.2 How does a single-vector representation variant compare to ColBERT-XM?

To analyze the effects of single-vector vs. multi-vector representations on our model's performance, we implement a variant of our modular dense retriever that maintains the bi-encoder architecture and modular encoder outlined in Sections 3.1 and 3.2, respectively, yet adopts a different late interaction scoring mechanism that operates on single-vector representations of the input sequences, i.e.,

$$\operatorname{sim}\left(\hat{\mathbf{H}}_q, \hat{\mathbf{H}}_p\right) = \cos\left(\operatorname{pool}\left(\hat{\mathbf{H}}_q\right), \operatorname{pool}\left(\hat{\mathbf{H}}_p\right)\right), \tag{9}$$

where $\operatorname{pool} : \mathbb{R}^{k \times d} \rightarrow \mathbb{R}^d$ distills a global representation for the whole text sequence using mean, max, or [CLS] pooling on the corresponding bags of contextualized term embeddings. We train this model, dubbed DPR-XM, on 25.6M MS MARCO triples with a batch size of 128 and learning rate warm up along the first 10% of steps to a maximum value of 2e-5, after which linear decay is applied.

Figure 4 illustrates the comparative performance of our XMOD-based dense retrievers. We observe that ColBERT-XM surpasses DPR-XM in the training language (i.e., English) by 4.5% on MRR@10. Furthermore, it consistently outperforms DPR-XM across the other 13 languages not encountered during training by an average of 4.9%. Supported by findings from Santhanam et al. (2022b), our results confirm that multi-vector models bypass the restrictive information bottleneck inherent in single-vector models, enabling a richer and more nuanced representation of queries and passages, thereby yielding higher retrieval performance.

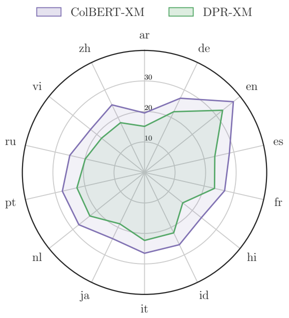

Figure 4: MRR@10 results of our multi-vector representation retriever (ColBERT-XM) compared to its single-vector counterpart (DPR-XM) on mMARCO dev set.

4.3.3 How does ColBERT-XM generalize to out-of-distribution data?

To assess ColBERT-XM's capabilities for out-of-distribution generalization, we conduct a zero-shot evaluation on Mr. TyDi, encompassing five languages not covered in mMARCO – notably Swahili, Bengali, and Telugu, which are commonly identified as low-resource. Table 2 reports the zero-shot performance of ColBERT-XM alongside the BM25, mT5-based cross-encoder, and mColBERT baselines. We find that ColBERT-XM shows substantial generalization across the out-of-distribution data. While not as effective as the computationally expensive cross-attentional mT5$_{\text{BASE}}$ reranking model on the rank-aware MRR@100 metrics, ColBERT-XM outperforms its non-modular mColBERT counterpart. Notably, on the rank-unaware R@100 metrics, ColBERT-XM matches closely and even surpasses the more resource-intensive mColBERT and mT5 retrieval models, which have been trained on many more samples and languages. These findings highlight our model's ability to efficiently adapt to domains and languages beyond its original training scope.

4.3.4 What is the environmental footprint of ColBERT-XM?

Given the growing concerns over carbon emissions and climate change, the environmental impact of AI models has become a crucial issue. In a quest for achieving ever-increasing performance, many works prioritize effectiveness over ef-

	Model	Type	*ar*	*bn*	*en*	*fi*	*id*	*ja*	*ko*	*ru*	*sw*	*te*	Avg
							MRR@100						
1	BM25 (Pyserini)	LEXICAL	36.8	41.8	14.0	28.4	37.6	21.1	28.5	31.3	38.9	34.3	31.3
2	mT5_{BASE} (Bonifacio et al., 2021)	CROSS	**62.2**	**65.1**	35.7†	**49.5**	61.1†	**48.1**	**47.4**	**52.6**†	**62.9**	**66.6**	**55.1**
3	mColBERT (Bonifacio et al., 2021)	MULTI	55.3	48.8	32.9†	41.3	55.5†	36.6	36.7	48.2†	44.8	61.6	46.1
4	ColBERT-XM (ours)	MULTI	55.2	56.6	**36.0**†	41.8	57.1	42.1	41.3	52.2	56.8	50.6	49.0
							R@100						
5	BM25 (Pyserini)	LEXICAL	79.3	86.9	53.7	71.9	84.3	64.5	61.9	64.8	76.4	75.8	72.0
6	mT5_{BASE} (Bonifacio et al., 2021)	CROSS	88.4	**92.3**	72.4†	**85.1**	92.8†	83.2	76.5	76.3†	83.8	85.0	83.5
7	mColBERT (Bonifacio et al., 2021)	MULTI	85.9	91.8	78.6†	82.6	91.1†	70.9	72.9	86.1†	80.8	**96.9**	83.7
8	ColBERT-XM (ours)	MULTI	**89.6**	91.4	**83.7**†	84.4	**93.8**	**84.9**	**77.6**	**89.1**	**87.1**	93.3	**87.5**

Table 2: Out-of-domain retrieval performance on Mr. TYDI test set. All supervised models were fine-tuned on one or more languages from mMARCO. † indicates performance on languages seen during training. The best results are marked in **bold**, and the second best are underlined.

ficiency, leading to models whose training requires significant energy consumption often derived from non-renewable resources, thereby exacerbating the global carbon footprint. Our comparative analysis demonstrates that ColBERT-XM exhibits reduced energy consumption and carbon emissions while performing comparably to leading retrieval models, underscoring its economic and environmental advantages.[2] Table 3 reveals that ColBERT-XM, trained for 7.5 hours only on private infrastructure with a carbon efficiency of 0.432 kgCO$_2$eq/kWh, utilized only 2.3 kWh of power for a carbon footprint of about 1.01 kgCO2eq, which is approximately the amount of emissions produced by burning 0.5 kg of coal. This contrasts significantly with competing models like mE5, which, despite its high performance, consumed about 100× more power during training (i.e., 230.4 kWh), emitting carbon emissions equivalent to burning 49.6 kg of coal. We calculate the estimated carbon emissions E_c as

$$E_{\text{c}} = \overbrace{\underbrace{P_{\text{TDP}}}_{\substack{\text{Thermal}\\\text{Design Power}}} \times \underbrace{T_{\text{train}}}_{\substack{\text{Training}\\\text{time}}}}^{\text{Power consumption}} + \underbrace{C_{\text{eff}}}_{\substack{\text{Carbon}\\\text{efficiency}}} . \qquad (10)$$

Our analysis not only highlights the potential for reduced carbon emissions associated with multilingual dense retrievers, but also reflects a deliberate stride toward aligning AI models with the pressing need for environmental sustainability. By demonstrating a comparable performance with a fraction of the energy and carbon output, we hope to set a precedent for future research and development in the field, emphasizing the importance of eco-friendly retrieval systems. Note that we also report inference costs in Appendix C.

	Model	Hardware	TDP (W)	Training time (h)	Power (kWh)	Emission (kgCO$_2$eq)
1	mE5$_{\text{BASE}}$	32× V100	300	24	230.4	99.52
2	mMiniLM$_{\text{L6}}$	1× A100	400	50	20.0	8.64
3	mColBERT	1× V100	300	36	10.8	4.67
4	mT5$_{\text{BASE}}$	1× TPUv3	283	27	7.6	3.30
5	ColBERT-XM	1× H100	310	7.5	2.3	1.01

Table 3: Power efficiency and carbon footprint of existing multilingual retrieval models.

5 Conclusion

This research presents ColBERT-XM, a multilingual dense retrieval model built upon the XMOD architecture, which effectively learns from monolingual fine-tuning in a high-resource language and performs zero-shot retrieval across multiple languages. Despite being trained solely in English, ColBERT-XM demonstrates competitive performance compared to existing state-of-the-art neural retrievers trained on more extensive datasets in various languages. An in-depth analysis reveals that our modular model learns faster, consumes a fraction of energy, and has a lower carbon footprint than existing multilingual models, thereby balancing its efficacy with environmental sustainability goals. Additionally, ColBERT-XM generalizes on out-of-distribution data and low-resource languages without further training, performing closely or surpassing strong retrievers. We believe that our research can help build effective retrieval systems for many languages while eliminating the need for language-specific labeled data, thus fostering inclusivity and linguistic diversity by helping individuals access information in their native languages.

Limitations

This section enumerates our work's limitations.

[2] Estimations were conducted using the MachineLearning Impact calculator (Lacoste et al., 2019).

Broader evaluation across diverse datasets. While our model's evaluation predominantly relies on the mMARCO dataset (Bonifacio et al., 2021), future investigations could benefit from exploring a broader spectrum of multilingual retrieval datasets, such as MIRACL (Zhang et al., 2022), SWIM-IR (Thakur et al., 2023), and MLDR (Chen et al., 2024). Additionally, examining the model's proficiency in domain-specific retrieval could offer valuable insights into its adaptability to specialized knowledge areas. However, such benchmarks are scarce in multilingual contexts.

Distillation of expressive retrieval models. Instead of the pairwise cross-entropy loss employed in ColBERTv1, a KL-divergence loss aimed at distilling the scores from a more sophisticated cross-encoder model, as used in ColBERTv2, could yield notable performance improvement (Santhanam et al., 2022b). Nevertheless, our estimates suggest this supervision scheme would require around $9.3\times$ more computational time for training on our system, surpassing our current resource allocation. As such, we let this exploration for future work.

Adaptability to cross-lingual retrieval. While this study presents a multilingual model designed for information retrieval within the same language, investigating its cross-lingual retrieval capabilities – i.e., identifying relevant passages in a target language based on queries in a different source language – represents a compelling direction for future research, especially in light of increasing needs for systems that can transcend language barriers.

Model interpretability. Enhancing the interpretability of retrieval model outputs is essential for boosting user confidence and ensuring system transparency, particularly given the complex linguistic and cultural nuances present in multilingual contexts. Building on seminal works in the area (Sudhi et al., 2022; Anand et al., 2023), our future efforts will focus on deepening our understanding of ColBERT-XM's decision-making mechanisms.

Evaluation of post-hoc language addition. Existing research suggests that XMOD can accommodate new languages post-hoc while mitigating the curse of multilinguality, a property demonstrated in text classification (Pfeiffer et al., 2022). However, extending this exploration to retrieval is limited by the availability of IR datasets in low-resource languages beyond the 81 already integrated in the pre-trained XMOD checkpoint, since creating such datasets is extremely tedious and beyond the scope of this work. We hope to investigate this characteristic within retrieval contexts in future work.

Broader Impacts

This section discusses our approach's ethical considerations, societal implications, and misuse.

Ethical considerations. Our work mostly leverages the widely recognized MS MARCO dataset, which contains over half a million anonymized queries collected from Bing's search logs, ensuring that our data sourcing practices are ethical and protect individual privacy. By leveraging mMARCO's direct translations, we ensure a fair and unbiased distribution of samples across languages, thereby avoiding the reinforcement of stereotypes. Furthermore, the combination of automated translation and manual labeling of the dataset ensures the reliability and precision of the ground truth data. This approach is essential for reducing label bias, which can arise from human annotators' varying proficiency levels and backgrounds.

Societal implications. Multilingual retrieval models significantly impact society by reducing language barriers and improving information accessibility for all. Our research aims to foster inclusivity and linguistic diversity, helping non-English speakers and those desiring information in their native languages. By developing models capable of effectively retrieving information in lesser-used languages, we contribute to equitable learning opportunities worldwide, enable businesses to serve a diverse international clientele, and prevent the digital marginalization of linguistic minorities.

Potential misuse. The premature deployment of a modular retrieval system presents a few risks. Notably, flaws or biases acquired during the monolingual fine-tuning phase could be inadvertently propagated to other languages when performing zero-shot transfer, thus perpetuating these malfunctions. More generally, the integrity of the underlying knowledge corpus is crucial, as even an effective system may retrieve relevant yet factually inaccurate content, thus unwittingly spreading misinformation. These concerns underscore the need for interpretability of retrieval model predictions to bolster user trust in such systems, which ColBERT-XM addresses with its interpretable MaxSim-based scoring mechanism.

Acknowledgments

This research is partially supported by the Sector Plan Digital Legal Studies of the Dutch Ministry of Education, Culture, and Science. In addition, this research was made possible, in part, using the Data Science Research Infrastructure (DSRI) hosted at Maastricht University.

References

Avishek Anand, Procheta Sen, Sourav Saha, Manisha Verma, and Mandar Mitra. 2023. Explainable information retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3448–3451. ACM. [Page 9]

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavas, Ivan Vulic, and Anna Korhonen. 2021. MAD-G: multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781. ACL. [Page 3]

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1538–1548. ACL. [Page 3]

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com. [Page 5]

Luiz Henrique Bonifacio, Israel Campiotti, Roberto de Alencar Lotufo, and Rodrigo Nogueira. 2021. mmarco: A multilingual version of MS MARCO passage ranking dataset. *CoRR*, abs/2108.13897. [Pages 5, 6, 8, and 9]

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2023. When is multilinguality a curse? language modeling for 250 high- and low-resource languages. *CoRR*, abs/2311.09205. [Page 3]

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *CoRR*, abs/2402.03216. [Page 9]

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*, pages 8440–8451. ACL. [Pages 1, 3, and 6]

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. ACL. [Pages 1, 3, and 6]

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR conference on research and development in Information Retrieval*, pages 2288–2292. ACM. [Page 3]

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *CoRR*, abs/1811.08008. [Page 3]

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2208–2222. ACL. [Page 3]

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799. PMLR. [Pages 3 and 4]

David A. Hull and Gregory Grefenstette. 1996. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–57. ACM. [Page 2]

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547. [Page 5]

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781. ACL. [Page 3]

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48. ACM. [Pages 2, 3, 4, 5, and 14]

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *CoRR*, abs/1910.09700. [Page 8]

Dawn J. Lawrie, Eugene Yang, Douglas W. Oard, and James Mayfield. 2023. Neural approaches to multilingual information retrieval. In *Proceedings of the 45th European Conference on Information Retrieval*, pages 521–536. Springer. [Page 3]

Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Y. Zhao. 2023. Rethinking the role of token retrieval in multi-vector retrieval. *CoRR*, abs/2304.01982. [Page 3]

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*, pages 11891–11907. Association for Computational Linguistics. [Page 14]

Jimmy Lin and Xueguang Ma. 2021. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *CoRR*, abs/2106.14807. [Page 3]

Jimmy Lin, Rodrigo Frassetto Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers. [Page 1]

Robert Litschko, Ivan Vulic, and Goran Glavas. 2022. Parameter-efficient neural reranking for cross-lingual and multilingual retrieval. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1071–1082. International Committee on Computational Linguistics. [Page 3]

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*. [Page 5]

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*, pages 246–254. Springer. [Page 1]

Joel Mackenzie, Andrew Trotman, and Jimmy Lin. 2021. Wacky weights in learned sparse representations and the revenge of score-at-a-time query evaluation. *CoRR*, abs/2110.11540. [Page 14]

Yury A. Malkov and Dmitry A. Yashunin. 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836. [Page 14]

Antonio Mallia, Michal Siedlaczek, Joel M. Mackenzie, and Torsten Suel. 2019. PISA: performant indexes and search for academia. In *Proceedings of the Open-Source IR Replicability Challenge co-located with 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, volume 2409 of *CEUR Workshop Proceedings*, pages 50–56. CEUR-WS.org. [Page 14]

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2006–2029. ACL. [Page 1]

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2018. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268v3. [Page 5]

Rodrigo Frassetto Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718. ACL. [Page 3]

Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424. [Pages 3 and 4]

Jonas Pfeiffer, Naman Goyal, Xi Victoria Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pretraining modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495. ACL. [Pages 1, 3, 4, and 9]

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: an adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7654–7673. ACL. [Page 3]

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. *Advances in Neural Information Processing Systems*, 30:506–516. [Page 3]

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3980–3990. ACL. [Pages 4 and 5]

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of the 3rd Text REtrieval Conference*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology. [Pages 3 and 5]

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. PLAID: an efficient engine for late interaction retrieval. In *Proceedings of* *the 31st ACM International Conference on Information & Knowledge Management*, pages 1747–1756. ACM. [Page 14]

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734. ACL. [Pages 5, 7, 9, 12, and 14]

Josef Sivic and Andrew Zisserman. 2003. Video google: A text retrieval approach to object matching in videos. In *9th IEEE International Conference on Computer Vision*, pages 1470–1477. IEEE Computer Society. [Page 14]

Viju Sudhi, Sabine Wehnert, Norbert Michael Homner, Sebastian Ernst, Mark Gonter, Andreas Krug, and Ernesto William De Luca. 2022. Bite-rex: An explainable bilingual text retrieval system in the automotive domain. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3251–3255. ACM. [Page 9]

Nandan Thakur, Jianmo Ni, Gustavo Hernández Ábrego, John Wieting, Jimmy Lin, and Daniel Cer. 2023. Leveraging llms for synthesizing training data across many languages in multilingual dense retrieval. *CoRR*, abs/2311.05800. [Page 9]

Andrew Trotman and Matt Crane. 2019. Micro- and macro-optimizations of saat search. *Software: Practice and Experience*, 49(5):942–950. [Page 14]

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 text embeddings: A technical report. *CoRR*, abs/2402.05672. [Page 6]

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, pages 2140–2151. ACL. [Page 5]

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics. [Page 5]

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130. ACL. [Page 3]

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597. [Page 1]

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*. OpenReview.net. [Page 3]

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 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498. ACL. [Page 5]

Eugene Yang, Suraj Nair, Ramraj Chandradevan, Rebecca Iglesias-Flores, and Douglas W. Oard. 2022a. C3: continued pretraining with contrastive weak supervision for cross language ad-hoc retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2507–2512. ACM. [Page 6]

Eugene Yang, Suraj Nair, Dawn J. Lawrie, James Mayfield, and Douglas W. Oard. 2022b. Parameter-efficient zero-shot transfer for cross-language dense retrieval with adapters. *CoRR*, abs/2212.10448. [Page 3]

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using lucene. *ACM Journal of Data and Information Quality*, 10(4):16:1–16:20. [Page 14]

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. tydi: A multi-lingual benchmark for dense retrieval. *CoRR*, abs/2108.08787. [Page 5]

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2022. Making a MIRACL: multilingual information retrieval across a continuum of languages. *CoRR*, abs/2210.09984. [Page 9]

A Centroid-based Indexing

ColBERTv2's centroid-based indexing consists of three main stages (Santhanam et al., 2022b).

First, we select a set of cluster centroids $\mathcal{C} = \{\mathbf{c}_j \in \mathbb{R}^{d_{\text{out}}}\}_{j=1}^{N}$ of size N, proportional to the square root of the estimated number of term embeddings across the entire passage collection, by

# Training Examples	*en*	*es*	*fr*	*it*	*pt*	*id*	*de*	*ru*	*zh*	*ja*	*nl*	*vi*	*hi*	*ar*	Avg
								MRR@10							
3.2M	35.7	27.7	25.9	26.2	26.9	25.3	26.2	24.4	24.0	23.9	26.5	21.8	23.2	19.2	25.5
6.4M	37.2	28.5	26.9	26.5	27.6	26.3	27.0	25.1	24.6	24.1	27.5	22.6	23.8	19.5	26.2
12.8M	38.1	28.6	26.8	26.9	27.5	26.6	27.1	25.4	24.9	24.2	27.3	22.9	23.8	19.5	26.4
19.2M	38.2	28.7	26.8	26.7	27.9	26.7	27.1	25.7	25.0	24.1	27.5	23.2	23.7	19.3	26.5
25.6M	38.0	28.4	26.7	26.8	27.8	26.6	27.1	26.0	25.2	24.2	27.5	23.2	23.8	19.6	26.5
								R@10							
3.2M	63.8	50.4	48.2	47.8	49.6	46.8	48.3	46.1	44.9	44.3	49.2	41.2	43.4	35.6	47.1
6.4M	65.7	52.0	49.2	48.2	50.5	48.3	49.5	47.3	46.0	44.6	49.8	42.4	44.2	36.5	48.2
12.8M	66.4	51.8	48.7	48.6	50.5	48.3	49.6	47.1	45.9	45.0	50.0	42.3	43.8	36.4	48.2
19.2M	67.0	52.0	49.1	48.2	50.4	48.9	49.6	47.8	46.0	44.8	50.0	42.8	43.6	35.7	48.3
25.6M	67.0	51.9	48.7	48.8	50.5	48.6	49.7	47.9	46.4	45.0	50.0	42.7	43.8	36.2	48.4
								R@100							
3.2M	88.5	77.2	75.1	73.6	75.3	73.3	73.1	73.0	71.6	71.2	74.4	66.8	68.6	59.3	72.9
6.4M	89.3	77.5	75.2	74.1	75.8	74.5	73.9	73.6	72.2	71.4	75.2	67.5	69.8	60.4	73.6
12.8M	90.1	77.7	75.3	73.8	75.6	73.9	73.9	73.6	72.2	71.4	75.0	67.2	69.0	59.7	73.5
19.2M	90.0	77.4	75.2	73.6	75.7	74.4	74.1	73.8	72.5	71.3	75.1	67.9	69.1	59.7	73.6
25.6M	90.0	77.5	75.3	73.6	75.7	74.1	74.2	73.9	72.7	71.4	75.3	67.8	69.4	59.7	73.6
								R@1000							
3.2M	96.3	88.7	87.5	86.3	87.4	86.2	85.5	85.7	84.7	83.8	86.8	81.5	81.4	75.1	85.5
6.4M	96.5	88.4	87.3	86.1	87.1	86.7	86.0	85.7	84.8	83.6	86.8	81.6	82.2	74.8	85.5
12.8M	96.5	88.0	87.5	85.8	86.9	86.0	85.4	85.6	84.7	83.6	86.4	80.9	81.4	74.3	85.2
19.2M	96.6	87.8	87.2	85.9	86.8	86.5	85.2	85.4	84.4	83.2	86.9	81.1	81.5	74.0	85.2
25.6M	96.7	87.8	87.3	85.8	87.0	86.2	85.3	85.4	84.3	83.4	87.0	80.9	81.6	74.0	85.2

Table 4: Influence of training samples on the performance of ColBERT-XM model on mMARCO small dev set.

applying k-means clustering to the contextualized term embeddings of only a sample of all passages.

Then, every passage in the corpus is processed using the modular language representation model, as detailed in Section 3.2, and the resulting contextualized term embeddings are assigned the identifier of the closest centroid $\mathbf{c}_j \in \mathcal{C}$, which requires $\lceil \log_2 |\mathcal{C}| \rceil$ bits to be encoded. Additionally, a *residual* representation $\mathbf{r}_i^p \in \mathbb{R}^{d_{\text{out}}}$ is computed for each term embedding to facilitate its reconstruction given $\mathbf{r}_i^p = \hat{\mathbf{h}}_i^p - \mathbf{c}_j$. To enhance storage efficiency, each dimension of this residual vector is quantized into 2-bit values. Consequently, storing each term vector requires $2d_{\text{out}} + \lceil \log_2 |\mathcal{C}| \rceil$ bits, i.e. roughly $7\times$ less than the $16d_{\text{out}}$ bits needed for the 16-bit precision compression used in ColBERTv1, without compromising on retrieval quality.

Finally, the identifiers of the compressed term embeddings linked to each centroid are grouped together and saved on disk within an inverted list. At search time, the n_{probe} centroids closest to every term representation of a given query are identified, and the embeddings indexed under these centroids are fetched for a first-stage candidate generation. Specifically, the compressed embeddings associated with the selected centroids are accessed via the inverted list structure, decompressed, and scored against each query vector using the similarity metric. The computed similarities are then aggregated by passage for each query term and subjected to a max-pooling operation. Since not all terms from a given passage are evaluated but only those associated with the selected centroids, the scores from this preliminary retrieval stage serve as an approximate of the MaxSim operation described in Section 3.3, thus providing a lower bound on actual scores. These approximated values are summed across query terms, and the k passages with the highest scores undergo a secondary ranking phase. Here, the full set of term embeddings for each candidate passage is considered to calculate the exact MaxSim scores. The selected passages are then reordered based on these refined scores and returned.

B Experimental Details

Table 4 provides a comprehensive breakdown of ColBERT-XM's performance across individual languages on mMARCO small dev set, depending on

Model & Engine	Index Storage		Latency (ms/q)	
	Disk	Ratio	GPU	CPU
BM25				
w/ JASS (Trotman and Crane, 2019)	1.2GB	×0.4	–	16
w/ Anserini (Yang et al., 2018)	0.7GB	×0.2	–	40
w/ PISA (Mallia et al., 2019)	0.7GB	×0.2	–	8
mE5$_{\text{BASE}}$				
w/ HNSW (Malkov and Yashunin, 2020)	28GB	×9.2	1	66
w/ IVF-Flat (Sivic and Zisserman, 2003)	26GB	×8.5	2	552
ColBERT(-XM)				
w/ v1 (Khattab and Zaharia, 2020)	154GB	×50.5	178	–
w/ v2 (Santhanam et al., 2022b)	29GB	×9.5	122	3275
w/ PLAID (Santhanam et al., 2022a)	22GB	×7.3	55	370

Table 5: Efficiency results on MS MARCO. We report numbers from Mackenzie et al. (2021) and Li et al. (2023) for BM25 and ColBERT, respectively. Ratio denotes the factor between index size and plain text size. Measures not applicable are denoted "–".

the number of examples used for training.

C Inference Costs

To assess our approach's practicality for real-world deployment, we compare its computational and memory efficiency against sparse lexical (BM25) and single-vector dense (mE5$_{\text{BASE}}$) models.

Index size. First, we report the storage footprint associated with indexing MS MARCO's 8.8M passages using several retrieval engines for each method. All indices store dense vectors in `fp32` and corpus ids in `int64` (if necessary). While ColBERTv1's original indexing approach is highly inefficient, we observe that ColBERTv2's residual compression technique matches the storage of the widely-used HNSW nearest-neighbor index for a 768-dimensional single-vector model. Employing the PLAID retrieval engine further enhances storage efficiency by 1.3×.

Search latency. We then examine the average retrieval latency per query (in milliseconds) across the different retrieval engines. Following prior work (Mackenzie et al., 2021; Santhanam et al., 2022a), we exclude the online query encoding latency for neural models. For reference, this latency is identical for both ColBERT-XM and mE5$_{\text{BASE}}$ models – as they derive from the same XLM-RoBERTa architecture – with encoding times around 11ms on GPU and 44ms on CPU. The reported numbers are calculated by averaging the times required to return the top-1000 candidates for all MS MARCO dev set queries at a batch size of 1 to simulate streaming queries. Despite its more complex scoring mechanism, ColBERT exhibits competitive low-latency performance using the PLAID engine, especially on CPU where it demonstrates efficiency comparable to single-vector dense retrieval.

Note that the BM25 results are sourced from Mackenzie et al. (2021), which were generated on a 3.50 GHz Intel(R) Xeon(R) Gold 6144 CPU, while the ColBERT results are derived from Li et al. (2023), who used a NVIDIA A100 for GPU search and a 3.00 GHz Intel(R) Xeon(R) Platinum 8275CL for CPU search. For mE5$_{\text{BASE}}$, we measure latency on a NVIDIA H100 for GPU search and a 3.20 GHz AMD EPYC 7763 for CPU search. Given these measurements were conducted on distinct hardware configurations, they are meant to establish ColBERT(-XM)'s competitive efficiency rather than serving as absolute comparisons.

D Reproducibility

We ensure the reproducibility of the experimental results by releasing our source code on Github at `https://github.com/ant-louis/xm-retrievers`. In addition, we release our model checkpoints on the Hugging Face Hub at `https://huggingface.co/antoinelouis/colbert-xm` and `https://huggingface.co/antoinelouis/dpr-xm`.