Multi-stage Training of Bilingual Islamic LLM for Neural Passage Retrieval

Vera Pavlova
rttl labs, UAE
`v@rttl.ai`

Abstract

This study examines the use of Natural Language Processing (NLP) technology within the Islamic domain, focusing on developing an Islamic neural retrieval model. By leveraging the robust XLM-R$_{Base}$ model, the research employs a language reduction technique to create a lightweight bilingual large language model (LLM). Our approach for domain adaptation addresses the unique challenges faced in the Islamic domain, where substantial in-domain corpora exist only in Arabic while limited in other languages, including English.

The work utilizes a multi-stage training process for retrieval models, incorporating large retrieval datasets, such as MS MARCO, and smaller, in-domain datasets to improve retrieval performance. Additionally, we have curated an in-domain retrieval dataset in English by employing data augmentation techniques and involving a reliable Islamic source. This approach enhances the domain-specific dataset for retrieval, leading to further performance gains.

The findings suggest that combining domain adaptation and a multi-stage training method for the bilingual Islamic neural retrieval model enables it to outperform monolingual models on downstream retrieval tasks.[1]

1 Introduction

Despite the advancements in NLP technology, its application in the Islamic domain remains relatively limited. While various fields have harnessed NLP for tasks such as sentiment analysis, language translation, and chatbot development, the rich and complex textual resources within Islamic literature, such as the Holy Qur'an, Hadith, and scholarly articles, have not been fully leveraged.

Information retrieval (IR) plays a crucial role in the exploration of Islamic text. With the vastness of texts spanning centuries, efficient search
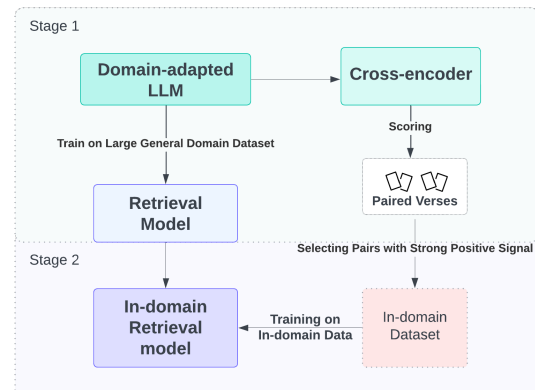

Figure 1: Multi-stage training of Islamic neural retrieval model.

methods are essential for scholars, researchers, and the general public. The ability to quickly locate specific passages, themes, or authors can significantly enhance understanding and facilitate deeper analysis. Moreover, given the intricate styles and diverse languages in which Islamic texts are written, advanced search tools can help bridge the gap between traditional scholarship and contemporary research needs. Effective retrieval not only saves time but also fosters a richer engagement with the cultural and intellectual heritage contained within Islamic literature (Bashir et al., 2023). One of the significant challenges is the diversity of languages used in Islamic texts, including Arabic, English, Urdu, etc, which complicates the creation of robust NLP tools. Researching and developing multilingual retrieval tools could assist in accessing Islamic literature for Arabic and non-Arabic speakers.

In this work, we study efficient ways to prepare a bilingual Islamic retrieval model. Addressing retrieval in both Arabic and English within the context of Islamic literature offers several significant benefits: firstly, it can increase the accessibility of Islamic literature to a broader audience. Classical

---

[1] A system is deployed at `https://rttl.ai/`

Arabic (CA) is the language of the Holy Qur'an and plays a crucial role in conducting retrieval tasks involving sacred scripture. English is also widely used for search across various domains, including the Islamic field.

Secondly, the use of multilingual or bilingual models enables cross-lingual transfer, which is crucial when there is insufficient data in some languages. English is a high-resource language, with a wealth of available corpora and pre-trained language models across various domains (Conneau et al., 2020). On the other hand, Arabic has more advantageous resources within the Islamic domain than English due to the availability of large in-domain corpora such as OpenITI (Romanov and Seydi, 2019). To benefit from both languages, we utilize the robust XLM-R$_{Base}$ model, which has undergone extensive training on general domain corpora predominately in English and offers state-of-the-art performance on downstream tasks.

We employ a language reduction technique Abdaoui et al. (2020) that enables the creation of a robust, lightweight bilingual model, preserving most of the performance of the XLM-R$_{Base}$. This model serves as the backbone for our retrieval system. It is known that retrieval models are sensitive to domain shifts, which can lead to a decline in performance (Thakur et al., 2021b). To address this issue, we perform domain adaptation using available text from Islamic literature and the OpenITI corpus in Arabic.

As a next step, we prepare a retrieval model using a dense retrieval approach (Karpukhin et al., 2020; Izacard et al., 2021). Training a robust retrieval model requires a substantial amount of in-domain labeled data, which is currently not available in the Islamic domain. However, there are large general domain datasets available for training retrieval models. We propose a multi-stage training process for an Islamic neural retrieval model that leverages both the large general domain datasets as well as the small in-domain datasets. (see Figure 1).

Additionally, we enhance our in-domain retrieval dataset in English by employing data augmentation techniques, which further improve the performance of the neural retrieval model. Our experiment showed that this approach improves the result on the evaluation dataset and outperforms strong monolingual baselines.

2 Related Work

Recent studies demonstrate that adapting existing LLMs pre-trained on general corpora for a new domain significantly improves performance on downstream tasks (Lee et al., 2019; Huang et al., 2019). The authors of the SciBERT model (Beltagy et al., 2019) showed that constructing a new Scivocab when pre-training SciBERT further enhances the performance of LLM. While pre-training a domain-specific model from scratch (Gu et al., 2020) allows for the inclusion of domain-specific vocabulary, this approach is costly and often impractical when the domain-specific corpora are limited in size. To avoid the random initialization of weights for new tokens and to expedite the pre-training process Poerner et al. (2020); Sachidananda et al. (2021); Pavlova and Makhlouf (2023) experiment with introducing new vocabulary and pre-training domain-specific models using existing checkpoints.

There are several approaches to reducing model size. Research by Sun et al. (2019), Tang et al. (2019), Sanh et al. (2019), and Li et al. (2020) has demonstrated that distilling transformer-based language models (Vaswani et al., 2017) results in significant size reduction while maintaining adequate performance. Another approach is model quantization, as explored in studies by Guo (2018), Jacob et al. (2017), Bondarenko et al. (2021), and Tian et al. (2023). While quantization can help address model size issues, it often compromises performance.

In contrast, language reduction (Abdaoui et al., 2020) does not lead to substantial performance loss. This method decreases the model size by preserving the encoder weights and only trimming the embedding matrix, eliminating languages that are unnecessary for the specific task at hand.

The survey Zhao et al. (2022) provides a detailed overview of dense retrieval, including various model architectures and training approaches. Other studies focusing on dense retrieval include Karpukhin et al. (2020), Qu et al. (2021), Ren et al. (2021). Thakur et al. (2021a), Wang et al. (2021) and Wang et al. (2022) proposed a data augmentation technique to train retrieval models when there is little data for in-domain training. This approach involves creating synthetic data points that can mimic real in-domain scenarios, enriching the existing dataset, and bridging the gap between limited data availability and the need for high-quality model performance.

3 Bilingual Islamic MLLM

The application of cross-lingual transfer capabilities of MLLMs helped to solve important NLP tasks in low-resource languages (Devlin et al., 2019; Lample and Conneau, 2019). Conneau et al. (2020) introduced the XLM-R and XLM-R$_{Base}$ with an increased model capacity trained on a large CommonCrawls corpus covering 100 languages. He demonstrated that increasing model capacity and adding more languages improves cross-lingual performance on low-resource languages to a certain extent. However, beyond a certain point, the overall performance on both monolingual and cross-lingual benchmarks begin to decline, a notion that he referred to as the "curse of multilinguality."

3.1 Size Reduction of LLM

In this work, we want to explore the performance of the XLM-R$_{Base}$ model after performing the language reduction technique, retaining only two languages (English and Arabic). We hypothesize that trimming the extended vocabulary of the XLM-R base model (250k) by removing languages not needed in the experiment will help reduce the model size, enhance model performance on downstream tasks, and facilitate domain adaptation. One of the main advantages of Language Reduction is that it reduces the number of languages by pruning only the embedding matrix while preserving all encoder weights. Unlike Abdaoui et al. (2020), our language reduction method consists of the following steps (see Figure 2):

1. We select English and Arabic texts from a multilingual variant of the C4 corpus.

2. Train a new SentencePiece BPE tokenizer using this corpus.

3. We identify the intersection between the new tokenizer and the XLM-R$_{Base}$ tokenizer. [2] The tokens in this intersection, along with their corresponding weights, are copied to the new embedding matrix of the XLM-R2 model.

4. The encoder weights from XLM-R$_{Base}$ are transferred directly to the new XLM-R2 model.

3.2 Domain Adaptation of MLLM

Though language reduction allows us to benefit from the extensive training the XLM-R$_{Base}$ model underwent, it gives us an LLM pre-trained for the general domain (XLM-R2). It is essential to note that the performance of retrieval models often declines when faced with domain shifts (Thakur et al., 2021b); since our focus in this study is on retrieval tasks and we apply this model as a backbone for retrieval, implementing domain adaptation is a crucial step to mitigate performance deterioration. In most domains, corpora to pre-train MLLMs are English-centric. However, we encounter a unique situation in the Islamic domain where significant domain-specific corpora are available in Arabic rather than English. Performing domain adaptation on a bilingual model brings certain advantages. On the one hand, the XLM-R$_{Base}$ model was trained on extensive general domain English data, which helps to improve performance on Arabic tasks. On the other hand, the availability of larger Islamic corpora in Arabic also enables better results for domain-specific tasks in English. The Open Islamicate Texts Initiative (OpenITI) (Romanov and Seydi, 2019) has provided a substantial corpus of 1 billion words for pre-training LLMs in Classical Arabic, the language of Arabic Islamic literature (Inoue et al., 2021; Malhas and Elsayed, 2022). While the available text in English is primarily composed of Tafseer and Hadith texts. Utilizing the OpenITI corpus can assist in creating a larger in-domain corpus for pre-training. To ensure the corpus is not overly biased towards Arabic, we randomly selected a subset of the OpenITI corpus containing approximately 50 million words. This subset was combined with the text of Hadeeth and Tafseer in English and Arabic, resulting in a total corpus size of 100 million words for domain adaptation. The corpus size is relatively small; nevertheless, since the weights of the XLM-R2 model are initialized from the XLM-R$_{Base}$ model, we can employ continued pre-training. To tackle the word distribution shift, we incorporate new domain-specific vocabulary.

The steps of domain adaptation are the following (see Figure 2):

1. We train a new SentencePiece BPE tokenizer using a multilingual Islamic Corpus and identify the intersection between the new Islamic tokenizer and the XLM-R2 tokenizer. All the tokens outside of the intersection (2k to-
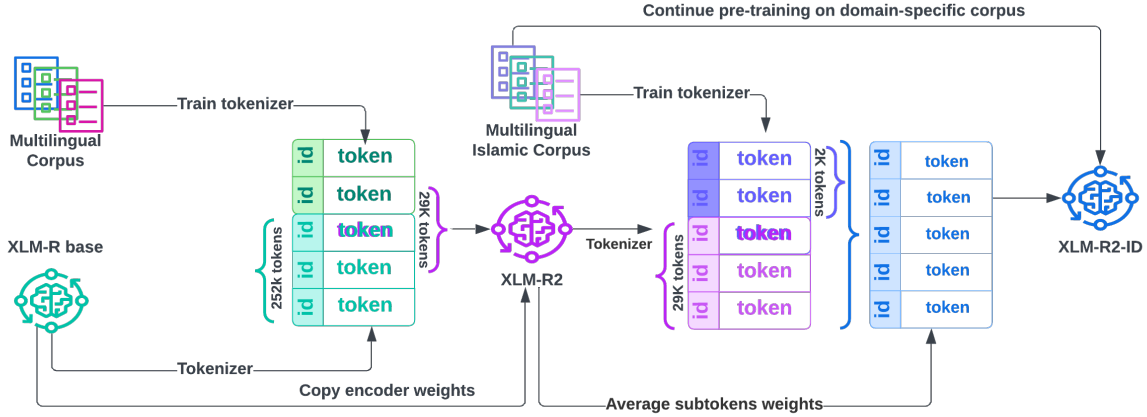
Figure 2: Language Reduction (on the left) gives the XLM-R2 model that goes through Domain Adaptation (on the right) and brings the XLM-R2-ID model. Diagram from Pavlova and Makhlouf (2024) with permission.

kens) are added to the embedding matrix of the XLMR-2 model (domain-specific vocabulary).

2. The weights for new Islamic tokens are assigned by averaging existing weights of subtokens from the XLM-R2 model.

3. We continue pre-training XLM-R2 using the domain-specific corpus, resulting in the XLM-R2-ID (Islamic domain) model. For more details on the hyperparameters, refer to Appendix A.

4 Domain-specific IR

4.1 Datasets, Metrics, and Training Approach

For retrieval, we use a dense retrieval approach (Karpukhin et al., 2020) using the sentence transformer framework that adds a pooling layer on top of LLM embeddings and produces fixed-sized sentence embedding (Reimers and Gurevych, 2019).

We utilize a sizeable general domain dataset (MS MARCO) for the first training stage in our multi-stage approach. The MS MARCO dataset consists of over half a million queries and is paired with a collection of 8.8 million passages (Bajaj et al., 2018). The language dataset is English; Bonifacio et al. (2021) released 13 machine-translated variants for 13 languages, including Arabic. The transfer language for XLM-R$_{\text{Base}}$ is English, while -XLM-R2-ID has been adapted for the Islamic domain, primarily using Arabic. We will experiment with both English and Arabic as transfer languages to assess their effectiveness in addressing the IR task at hand.

The loss function is designed within the framework of contrastive learning, which helps create an embedding space that brings related queries and their relevant passages closer together while distancing queries and irrelevant passages (van den Oord et al., 2018) , and formally defined as:

$$J_{\text{CL}}(\theta) =$$
$$\frac{1}{M}\sum_{i=1}^{M}\log\frac{\exp\sigma(f_\theta(x^{(i)}), f_\theta(y^{(i)}))}{\sum_{j=1}^{M}\exp\sigma(f_\theta(x^{(i)}), f_\theta(y^{(j)}))}$$

where σ is a similarity function (a cosine similarity), f_θ is the sentence encoder. To enhance training efficiency, we utilize in-batch negatives (Henderson et al., 2017; Gillick et al., 2019; Karpukhin et al., 2020) (for hyperparameter details, see Appendix A).

In-domain training of retrieval model. Training a retrieval model on large-size general domain data would produce a robust model that can distinguish similar passages from dissimilar ones. However, training on a small amount of in-domain data can further enhance the performance (Wang et al., 2022; Lu et al., 2021). For in-domain training for the second stage, we use QUQA (Alnefaie et al., 2023), an Arabic question-answering (QA) dataset based on the Holy Qur'an. It has 3382 pairs, including 1166 pairs from AyaTEC (Malhas and El-sayed, 2020), which we exclude as we use them for evaluation (see below). Some questions relate to more than one verse. We take each relation as a separate anchor-positive pair, which gives us 3252 pairs. The dataset has only an Arabic version; we

curate English in-domain training data to improve the in-domain training.

The challenge of limited domain-specific data can often be addressed by augmenting the training data using various methods. These methods include generating synthetic data (dos Santos Tanaka and Aranha, 2019), paraphrasing with synonyms (Wei and Zou, 2019), sampling and recombining new training pairs (Thakur et al., 2021a), employing round-trip translation (Yu et al., 2018; Xie et al., 2020), and utilizing denoising autoencoders (Wang et al., 2021). However, these techniques can distort the data, which is not ideal for religious and heritage datasets. To prevent data distortion, we create anchor-positive pairs based on the verse relations mentioned in Tafseer Ibn Katheer. This method facilitates the creation of relevant, verified, high-quality in-domain data without costly human annotations.

- First, we pair all the verses that have relation mentioned in Tafseer Ibn Katheer.

- Next, we filter out the pairs that may not be interpreted by the model as indicating a strong positive correlation. To filter out these pairs, we scored them using a cross-encoder model trained using the XLM-R2-ID model (for details on cross-encoder training see Appendix A). Cross-encoder is a powerful but expensive approach to identifying similar pairs. Though they are suboptimal to apply in solving real-world retrieval tasks due to high computational overhead, they can help in data augmentation, distillation, and re-ranking without enduring considerable domain shift (Humeau et al., 2020; Wang et al., 2022). Cross-encoder helps to filter out pairs with low similarity scores, leaving us with 2133 pairs for in-domain training.

- Lastly, we combine Arabic QUQA pairs with English pairs, which results in 5385 pairs for training.

Unlike Pavlova (2023), we do not create hard negatives for training the retrieval model. Instead, we apply the same contrastive learning framework, utilizing in-batch negatives as described earlier, to train on in-domain data. We ensure that there are no duplicate entries within the same batch and sample from English and Arabic in-domain datasets in proportion to their respective sizes. This approach allows all samples from each dataset to be utilized.

For evaluation, we combined the train and development split of the QRCD (Qur'anic Reading Comprehension Dataset) (Malhas and Elsayed, 2020) and converted it to the IR dataset (169 queries in total for testing). The QRCD dataset is in Arabic; to conduct evaluations in English, we utilized verified translations of this dataset into English. We use the Holy Qur'an text (Arabic) and Sahih International translation (English) as retrieval collections.[3] The QRCD is designed to retrieve passages composed of verses from the Holy Qur'an. The Holy Qur'an texts mentioned above are organized according to the passages based on the QRCD. We evaluate the models' performance using decision support metric Recall@100 and the order-aware metric MRR@10 (MS MARCO's official metric).

4.2 Baselines and Models

We use two monolingual models as our baselines. For English, we use ST/all-mpnet-base-v2, a robust monolingual model trained with contrastive learning objectives on 1B sentence pairs. [4] For Arabic, we use CL-AraBERT (Malhas and Elsayed, 2022), which was pre-trained on OpenITI corpus, and we fine-tuned as a retrieval model on Arabic MS MARCO and in-domain Arabic data using the same training loss described above (for hyperparameters see Appendix A). This choice of baselines serves two purposes. First, it enables us to evaluate how our bilingual model performs compared to monolingual models. Second, comparing our model against a strong retrieval model that is not domain-adapted (ST/all-mpnet-base-v2) allows us to assess the effects of domain adaptation. Additionally, contrasting it with a retrieval model trained using CL-AraBERT—adapted with the full OpenITI dataset, which consists of about 1 billion words — will help us evaluate the potential of utilizing a smaller corpus for domain adaptation.

As previously mentioned in Section 4.1, we experimented with MS MARCO in both English and Arabic to select which language performs better for retrieval tasks in the Islamic domain. We trained two models, XLM-R2-ID-EN and XLM-R2-ID-AR, which correspond to the first stage of our multi-stage training approach. We selected the model that exhibited superior performance on the evaluation dataset for use in the second stage.

In the second stage, where we train on in-

---

[3] https://tanzil.net/trans/
[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Models	EN	AR
ST/all-mpnet-base-v2	<u>0.388</u>	-
CL-AraBERT-indomain	-	<u>0.512</u>
XLM-R-in-domain	0.211	0.218
XLM-R-EN	0.302	0.267
XLM-R-AR	0.287	0.264
XLM-R-AR-in-domain	0.308	0.316
XLM-R2-ID-in-domain	0.348	0.465
XLM-R2-ID-EN	0.329	0.416
XLM-R2-ID-AR	0.387	0.498
XLM-R2-ID-AR-in-domain	**0.441**	**0.534**

Table 1: Model performance for MRR@10.

Models	EN	AR
ST/all-mpnet-base-v2	<u>0.619</u>	-
CL-AraBERT-indomain	-	<u>0.756</u>
XLM-R-in-domain	0.451	0.462
XLM-R-EN	0.492	0.496
XLM-R-AR	0.493	0.541
XLM-R-AR-in-domain	0.528	0.584
XLM-R2-ID-in-domain	0.592	0.706
XLM-R2-ID-EN	0.571	0.675
XLM-R2-ID-AR	<u>0.619</u>	0.72
XLM-R2-ID-AR-in-domain	**0.646**	**0.766**

Table 2: Model performance for Recall@100.

domain data, we developed the XLM-R2-ID-AR-in-domain. To evaluate the impact of multi-stage training, we also produced the XLM-R2-ID-in-domain model, which was trained solely on in-domain data without using MS MARCO. We aimed to analyze the performance of the XLM-R2-ID model after implementing domain adaptation and language reduction. To facilitate this analysis, we prepared four additional models trained from the XLM-R$_{Base}$ model for comparison with four models trained from the XLM-R2-ID. These include two models trained on the MS MARCO dataset in English and Arabic (XLM-R-EN and XLM-R-AR), a model trained in a multi-stage approach (XLM-R-AR-in-domain), and a model trained solely on in-domain data (XLM-R-in-domain).

4.3 Model Comparison

For our model comparison, we will examine four key aspects: First, we will compare model performance against a baseline. Second, we will assess how models trained from domain-adapted XLM-R2-ID models perform compared to those trained from general domain XLM-R$_{Base}$. Third, we will evaluate model performance that was trained in a multi-stage approach against training conducted solely with in-domain datasets (only stage two) or only with large general domain datasets (only stage one). Finally, we will analyze how the models perform on the evaluation dataset in English versus the evaluation dataset in Arabic.

In Tables 1 and 2, the best-performing model is in bold, and the second-best is underlined. In terms of MRR@10 for English (see Table 1), we see that most models perform worse than the baseline monolingual model, all-mpnet-base-v2 (0.388), with the exception of XLM-R2-ID-AR-in-domain (0.441). This model, which was trained using a multi-stage approach, outperforms the baseline. A similar trend is observed for MRR@10 in Arabic.

Table 1 shows that the second stage of in-domain training provides significant benefits for English, resulting in a 12% performance improvement (increasing from 0.387 to 0.441). For Arabic, the second stage yields an improvement of approximately 6%. The same is true for Recall@100 (see Table 2); only the models that utilized a multi-stage training approach were able to surpass a strong monolingual baseline, but the improvement was less pronounced than that observed in MRR@10.

Figures 3 and 4 illustrate the performance comparison between all stages of the models trained from XLM-R$_{Base}$ and the XLM-R2-ID for MRR@10, and Figures 5 and 6 for Recall@100. It is evident that all XLM-R2-ID models outperform the XLM-R$_{Base}$ models in both Arabic and English, with a particularly significant difference in performance observed in Arabic. This indicates that domain adaptation has been especially beneficial for the Arabic language.

Another important observation is that XLM-R$_{Base}$ models do not respond to multi-stage training as effectively as models based on XLM-R2-ID. Additionally, training solely on in-domain data results in competitive models; however, these still fall short compared to models trained using a multi-stage approach, similar to those trained only on general domain data.

Interestingly, although machine-translated, training on the Arabic version of MS MARCO yields better results than training on the original English version of MS MARCO. This trend holds true for evaluations in both Arabic and English. Despite starting the domain adaptation process from the XLM-R$_{Base}$ model, which is an English-
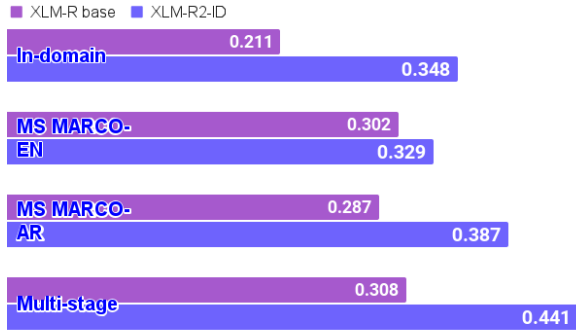
47

Figure 3 (left top chart):

Legend: XLM-R base ■ XLM-R2-ID ■

	XLM-R base	XLM-R2-ID
In-domain	0.211	0.348
MS MARCO-EN	0.302	0.329
MS MARCO-AR	0.287	0.387
Multi-stage	0.308	0.441

Figure 3: Comparison of the performance of the retrieval models trained from XLM-R$_{Base}$ and XLM-R2-ID for MRR@10 in English.

Figure 4 (left bottom chart):

Legend: XLM-R base ■ XLM-R2-ID ■

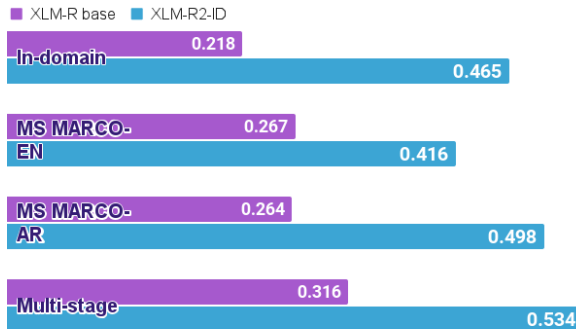|	XLM-R base	XLM-R2-ID
In-domain	0.218	0.465
MS MARCO-EN	0.267	0.416
MS MARCO-AR	0.264	0.498
Multi-stage	0.316	0.534

Figure 4: Comparison of the performance of the retrieval models trained from XLM-R$_{Base}$ and XLM-R2-ID for MRR@10 in Arabic.

Figure 5 (right top chart):

Legend: XLM-R base ■ XLM-R2-ID ■

	XLM-R base	XLM-R2-ID
In-domain	0.451	0.592
MS MARCO-EN	0.492	0.571
MS MARCO-AR	0.493	0.619
Multi-stage	0.528	0.646

Figure 5: Comparison of the performance of the retrieval models trained from XLM-R$_{Base}$ and XLM-R2-ID for Recall@100 in English.

Figure 6 (right bottom chart):

Legend: XLM-R base ■ XLM-R2-ID ■

	XLM-R base	XLM-R2-ID
In-domain	0.462	0.706
MS MARCO-EN	0.496	0.675
MS MARCO-AR	0.541	0.72
Multi-stage	0.584	0.766

Figure 6: Comparison of the performance of the retrieval models trained from XLM-R$_{Base}$ and XLM-R2-ID for Recall@100 in Arabic.

centric model, training on a small amount of Arabic data and adding domain-specific vocabulary significantly improved the results for the Arabic language.

As a final part of our comparison, we closely examine the performance of models using English and Arabic datasets. Figure 7 (MRR@10) and Figure 8 (Recall@100) demonstrate that the results for Arabic are superior to those for English across all models. The disparity is particularly pronounced in MRR@10, while the difference is less significant for Recall@100.

Overall, our experiments and comparisons reveal several important findings: domain adaptation of LLMs, even with a small corpus, significantly contributes to improved performance on downstream tasks. However, adaptation alone was insufficient to surpass strong monolingual baselines. Instead, the multi-stage training approach enhanced the results, allowing us to outperform the baselines.

Additionally, the XLM-R2-ID-in-domain model outperformed the retrieval model trained on CL-AraBERT, which was developed on a larger corpus.
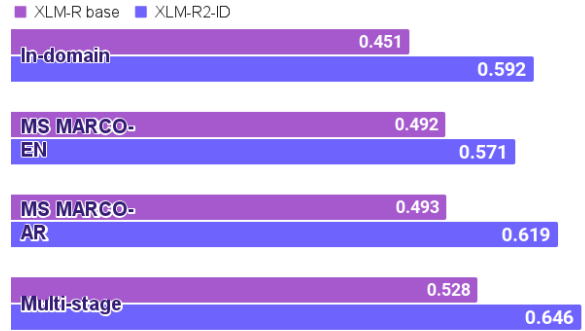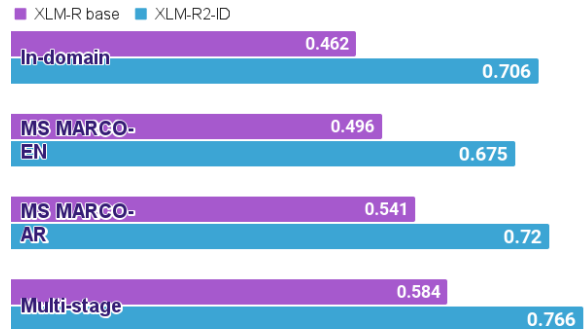
Models	EN	AR
XLM-R2-ID-AR-in-domain	**0.441**	**0.534**
XLM-R2-ID*-AR-in-domain	0.414	0.498
XLM-R2-ID-in-domain*	0.381	0.521

Table 3: Model performance for MRR@10 with two ablated models.

This suggests that even a small corpus can be effective, especially when leveraging strong XLM-R$_{Base}$ weights for a warm start.

5 Ablation Study

In this section, we conduct an ablation study on various aspects of our highest-performing model, XLM-R2-ID-AR-in-domain, which was trained using domain adaptation with extended corpus and a multi-stage approach with augmented in-domain data. To better understand the relative importance of each component, we examine the effects of removing each one individually.

First, we removed the domain adaptation that involved extending the Islamic corpus with OpenITI. We then used only the available Islamic texts from
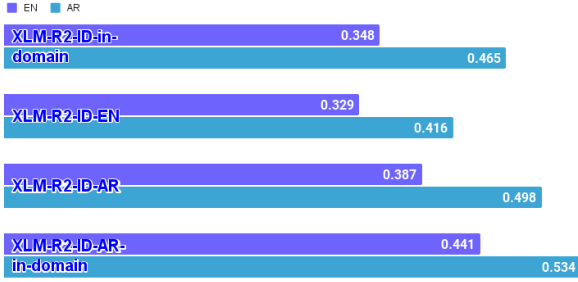
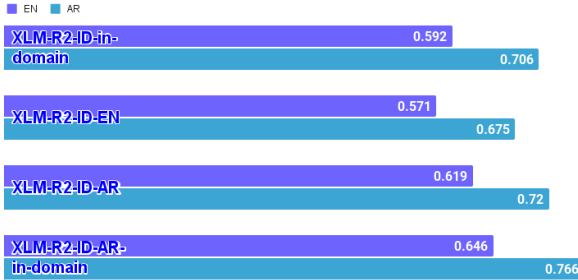Figure 7: Comparison of results between Arabic and English for MRR@10.

Figure 8: Comparison of results between Arabic and English for Recall@100.

Models	EN	AR
XLM-R2-ID-AR-in-domain	**0.646**	**0.766**
XLM-R2-ID*-AR-in-domain	0.643	0.739
XLM-R2-ID-in-domain*	0.622	0.751

Table 4: Model performance for Recall@100 with two ablated models.

Hadith and Tafseer in English and Arabic, totaling 50 million words. In Tables 3 and 4, we present the performance of the model XLM-R2-ID*-AR-in-domain, which was trained without OpenITI but still employed a multi-stage approach. The results show that MRR@10 decreased by approximately 6% for both Arabic and English. For Recall@100, the difference in performance for English was relatively small, whereas for Arabic, it was about 3.5%.

Next, we removed the augmentation of the in-domain corpus with English data, resulting in the model XLM-R2-ID-in-domain*. As shown in Table 3, this led to a more substantial decline in performance for English — around 14% - while the decrease for Arabic was only about 2.43%. Again, for Recall@100 (Table 4), the difference was less pronounced. This indicates that augmenting the in-domain data with English has a significant impact, especially for performance on retrieval task in English.

Moreover, our findings suggest that expanding

the pre-training corpus with OpenITI improved results for both Arabic and English. Since adding additional texts amounting to 50 million words does not significantly prolong pre-training time, we recommend this approach; however, as we demonstrated in the section 4.3, even without pre-training on an extended corpus of 1 billion words, using a relatively small corpus for domain adaptation can still yield significant improvements.

6 Conclusion

This study emphasizes the importance of leveraging domain adaptation to enhance the performance of LLMs on downstream tasks such as retrieval. By combining language reduction with domain adaptation applied to the XLM-R$_{Base}$ model, we developed a lightweight bilingual Islamic LLM (XLM-R2-ID). This model underwent a multi-stage training process and demonstrated improved performance on retrieval tasks, surpassing monolingual models on the evaluation datasets.

Moreover, incorporating an augmented in-domain dataset in English further enhanced the performance of the retrieval model during the second training phase.

Overall, our research demonstrates that combining domain adaptation of LLMs with multi-stage training of neural retrieval models leads to improved results in downstream tasks such as IR.

Limitations

One of the limitations of our study is that we conducted experiments only in English and Arabic; experiments that involve other languages may vary. Additionally, we used machine-translated datasets. While machine translation has not yet reached the quality of expert human translation, our use of the Arabic machine-translated version of MS MARCO has shown promising results.

Ethics Statement

We do not anticipate any considerable risks associated with our work. The data and other related resources in this work are publically available, and no private data is involved.

Acknowledgment

References

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of mutililingual BERT. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 119–123, Online. Association for Computational Linguistics.

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. HAQA and QUQA: Constructing two Arabic question-answering corpora for the Quran and Hadith. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 90–97, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, 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. *Preprint*, arXiv:1611.09268.

Muhammad Huzaifa Bashir, Aqil M Azmi, Haq Nawaz, Wajdi Zaghouani, Mona Diab, Ala Al-Fuqaha, and Junaid Qadir. 2023. Arabic natural language processing for qur'anic research: A systematic review. *Artificial Intelligence Review*, 56(7):6801–6854.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Luiz Henrique Bonifacio, Israel Campiotti, Roberto de Alencar Lotufo, and Rodrigo Frassetto Nogueira. 2021. mmarco: A multilingual version of MS MARCO passage ranking dataset. *CoRR*, abs/2108.13897.

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, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. 2019. Data augmentation using gans. *Preprint*, arXiv:1904.09135.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *CoRR*, abs/2007.15779.

Yunhui Guo. 2018. A survey on methods and theories of quantized neural networks. *Preprint*, arXiv:1808.04752.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *Preprint*, arXiv:1705.00652.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *CoRR*, abs/1904.05342.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *Preprint*, arXiv:1905.01969.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2017. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Preprint*, arXiv:1712.05877.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E. Gonzalez. 2020. Train large, then compress: Rethinking model size for efficient training and inference of transformers. *CoRR*, abs/2002.11794.

Jing Lu, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2021. Multi-stage training with improved negative contrast for neural passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6091–6103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rana Malhas and Tamer Elsayed. 2020. Ayatec: building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(6):1–21.

Rana Malhas and Tamer Elsayed. 2022. Arabic machine reading comprehension on the holy qur'an using cl-arabert. *Information Processing and Management*, 59(6):103068.

Vera Pavlova. 2023. Leveraging domain adaptation and data augmentation to improve qur'anic IR in English and Arabic. In *Proceedings of ArabicNLP 2023*, pages 76–88, Singapore (Hybrid). Association for Computational Linguistics.

Vera Pavlova and Mohammed Makhlouf. 2023. BIOptimus: Pre-training an optimal biomedical language model with curriculum learning for named entity recognition. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 337–349, Toronto, Canada. Association for Computational Linguistics.

Vera Pavlova and Mohammed Makhlouf. 2024. Building an efficient multilingual non-profit IR system for the islamic domain leveraging multiprocessing design in rust. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 981–990, Miami, Florida, US. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. Inexpensive domain adaptation of pretrained language models: Case studies on biomedical NER and covid-19 QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1482–1490, Online. Association for Computational Linguistics.

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

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 (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maxim Romanov and Masoumeh Seydi. 2019. Openiti: a machine-readable corpus of islamicate texts. *Zenodo, URL: https://doi. org/10.5281/zenodo*, 3082464.

Vin Sachidananda, Jason Kessler, and Yi-An Lai. 2021. Efficient domain adaptation of language models via adaptive tokenization. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 155–165, Virtual. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from BERT into simple neural networks. *CoRR*, abs/1903.12136.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021a. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021b. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) - Datasets and Benchmarks Track (Round 2)*.

Rong Tian, Zijing Zhao, Weijie Liu, Haoyan Liu, Weiquan Mao, Zhe Zhao, and Kan Zhou. 2023. SAMP: A model inference toolkit of post-training quantization for text processing via self-adaptive mixed-precision. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 123–130, Singapore. Association for Computational Linguistics.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360, Seattle, United States. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training. *Preprint*, arXiv:1904.12848.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *Preprint*, arXiv:1804.09541.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42:1 – 60.

A Appendix

Computing Infrastructure	1x H100 (80 GB)
Hyperparameter	**Assignment**
number of epochs	60
batch size	128
maximum learning rate	0.0005
learning rate optimizer	Adam
learning rate scheduler	None or Warmup linear
Weight decay	0.01
Warmup proportion	0.06
learning rate decay	linear

Table 5: Hyperparameters for pre-training of XLM-R2-ID model.

Computing Infrastructure	1x H100 (80 GB)
Hyperparameter	**Assignment**
number of epochs	1
batch size	32
learning rate	2e-5
pooling	mean

Table 6: Hyperparameters for training retrieval models.

Cross-encoder training details

In a cross-encoder architecture, a pair of sentences are simultaneously fed into a transformer-like model, allowing attention to be applied across all tokens to generate a similarity score. The model is trained using triples provided by MS MARCO, starting from the XLM-R2-ID model checkpoint, with a classification task and employing Cross Entropy Loss.

Although this approach does not enable end-to-end information retrieval and involves significant computational overhead, it often outperforms other methods in many information retrieval (IR) tasks. Additionally, it can be utilized for mining hard negatives, data augmentation, and reranking.