HLTCOE at TREC 2023 NeuCLIR Track

Eugene Yang, Dawn Lawrie, James Mayfield
Human Language Technology Center of Excellence, Johns Hopkins University, USA
{eugene.yang,lawrie,mayfield}@jhu.edu

ABSTRACT

The HLTCOE team applied PLAID, an mT5 reranker, and document translation to the TREC 2023 NeuCLIR track. For PLAID we included a variety of models and training techniques – the English model released with ColBERT v2, translate-train (TT), Translate Distill (TD) and multilingual translate-train (MTT). TT trains a ColBERT model with English queries and passages automatically translated into the document language from the MS-MARCO v1 collection. This results in three cross-language models for the track, one per language. MTT creates a single model for all three document languages by combining the translations of MS-MARCO passages in all three languages into mixed-language batches. Thus the model learns about matching queries to passages simultaneously in all languages. Distillation uses scores from the mT5 model over non-English translated document pairs to learn how to score query-document pairs. The team submitted runs to all NeuCLIR tasks: the CLIR and MLIR news task as well as the technical documents task.

KEYWORDS

CLIR, NeuCLIR, ColBERT, Translate-Distill, multilingual training

1 INTRODUCTION

This year the HLTCOE's primary contribution was in experimentation with multiple ways to fine-tune mPLMs for CLIR and MLIR. Specifically, we submitted a suite of ColBERT-X models, including Translate-Train [13] and Translate-Distill [21] with an effective cross-encoder developed at last tear's NeuCLIR track as the teacher [6]. The HLTCOE also contributed a set of baseline BM25 retrieval runs using Patapsco [1] and re-ranked runs with the cross-encoder trained last year for pool enrichment. Table 1 summarizes all submitted runs except the BM25 ones. In the rest of this paper, we describe our systems, submissions, and observations.

2 MT5 RERANKER

At NeuCLIR 2023, the most effective submission was a reranking model using the MonoT5 model with mT5XXL, having 13 billion parameters [6, 8]. This submission uses pointwise re-ranking by taking a softmax over the decoding probabilities of two specific tokens (_ and _true for mT5XXL) as the probability of the document being relevant given the query [15]. In this paper, we use the name "mT5 reranker" to refer to this specific reranker for convenience. The input queries are titles concatenated with descriptions. For the Chinese CLIR task only we use the reversed order as this has been shown to be more effective in prior works [6].

3 COLBERT RETRIEVAL WITH PLAID

Our systems primarily use PLAID [17], an implementation of the ColBERT [7] retrieval architecture that encodes each token as a vector. Prior work on training CLIR dense retrieval models has demonstrated successes augmenting training queries and passages with translation to match the CLIR target languages [13]. In the NeuCLIR 2022 track [8], our ColBERT-X models trained with Translate-Train were the most effective end-to-end neural dense retrieval models.[1]

All ColBERT-X variants (including PLAID) use the title concatenated with the description as the query. We break each document into passages of 180 tokens with a stride of 90. For the PLAID index setting, we use one bit for each dimension of the residual vector. We search for 2500 passages and aggregate passage scores into a document scores using MaxP [2]. For comparison, we also submitted a handful of runs using the original ColBERT-X implementation which does not implement the compression technique; such runs are marked using "ColBERT-X" as the model in Table 1.

The runs called "PLAID_shard_by_date_1bit_v1_tt" and "plaid_v1_mtt_1bit_date" ordered the collection by document creation date if known or by download date, and then created indexes of the three month time windows through the time period of the collection. Each three-month window was indexed separately in a PLAID index. Dates were introduced to topics where applicable. As Table 2 indicates, in some cases, a start date or date range was identified. Start dates and date ranges were identified manually by looking for topics that contain specific events and looking for the start or start and end dates of such events in Wikipedia. These dates were used to identify which segment or segments of the collection would be used to rank documents.

3.1 Translate-Distill for CLIR using mT5 Reranker

Distillation is an effective strategy for training a small yet efficient model that mimics the effectiveness of a larger and computationally more expensive model [4, 16]. In our submissions, we explored training a ColBERT-X model to mimic the behavior of the powerful mT5 reranker. This training method is known as Translate-Distill and is described in more detail in Yang et al. [21].

Translate-Distill starts with selecting hard passages for each training query in the MS MARCO training set. We use an **English** ColBERTv2 model [18] to retrieve the top 50 passages for each query from the MS MARCO training set. To obtain the teacher scores for each query/passage pair, the mT5 reranker scores the **translated** passage along with the English query. [2] Finally, we train the

[1] Given that the authors are also organizers of the track, ColBERT-X runs are marked as manual runs. Although unlikely, performance might have been affected by knowledge that was only accessible to the organizers.

[2] Yang et al. [21] finds that this configuration is not the optimal way to obtain scores. Instead English query/passage pairs should be used to obtain scores.

Table 1: HLTCOE runs. Run names in italic are monolingual runs.

	Run Name	Type	Model	Query	Description
	News Collection with Single Language Documents				
(c1)	PLAIDkd-mT5gt-{td}	Hybrid	PLAID » mT5	TD	PLAID with Translate-Distill followed by mt5 reranker
(c2)	PLAIDkd-monomt5tt-td	Dense	PLAID	TD	PLAID with Translate-Distill
(c3)	mT5gt-{td}	Rerank	mT5	TD	mT5-XXL reranker
(c4)	plaid_v2_eng_1	Dense	PLAID	TD	English ColBERTv2 with PLAID using DT
(c5)	*PLAID192mono-td*	Dense	PLAID	TD	Monolingual PLAID
(c6)	colbertX	Dense	ColBERT-X	TD	ColBERT-X
(c7)	PLAID_shard_by_date_1bit_v1_tt	Dense	PLAID	TD	PLAID with date-sharded indexes
(c8)	PSQ-td	Sparse	PSQ	TD	PSQ-HMM
(c9)	PSQ-t	Sparse	PSQ	T	PSQ-HMM
	News Collection with Multilingual Documents				
(m1)	colbertX	Dense	ColBERT-X	TD	MTT ColBERT-X
(m2)	plaid_v1_mtt_1bit	Dense	PLAID	TD	MTT ColBERT-X with PLAID using 1 residual bit
(m3)	plaid_v2_eng_1	Dense	PLAID	TD	English ColBERTv2 with PLAID using DT
(m4)	plaid_v1_mtt_1bit_date	Dense	PLAID	TD	MTT PLAID with date-sharded indexes
(m5)	PSQraw-td	Sparse	PSQ	TD	Combining CLIR PSQ-HMM scores
(m6)	PSQraw-t	Sparse	PSQ	T	Combining CLIR PSQ-HMM scores
	Technical Document				
(t1)	rerank_mt5gt_td	Rerank	mT5	TD	mT5-XXL reranker using Google translated queries
(t2)	plaid_tt_mt5gt_td	Hybrid	PLAID » mT5	TD	Eng-Zho TT ColBERT-X followed by mT5 reranker
(t3)	plaid_distilled_td	Dense	PLAID	TD	Distilled PLAID
(t4)	psq_td_f32	Sparse	PSQ	TD	PSQ-HMM
(t5)	psq_t_f32	Sparse	PSQ	T	PSQ-HMM
(t6)	colbert_x_td	Dense	ColBERT-X	TD	Eng-Zho TT ColBERT-X
(t7)	plaid_tt_td	Dense	PLAID	TD	Eng-Zho TT ColBERT-X with PLAID
(t8)	plaid_V2model_td	Dense	PLAID	TD	English ColBERT with DT
(t9)	blade-d	L-Sparse	BLADE	T	BLADE
(t10)	blade-td	L-Sparse	BLADE	TD	BLADE
(t11)	blade-t	L-Sparse	BLADE	T	BLADE
(t12)	plaid_jhpolo_td	Dense	PLAID	TD	Eng-Zho TT ColBERT-X with JH-POLO
(t13)	*plaid_monozh_mt5ht_td*	Hybrid	PLAID » mT5	TD	Chinese ColBERT followed by mT5 reranker
(t14)	*rerank_mt5ht_td*	Rerank	mT5	TD	mT5-XXL reranker using human translated queries
(t15)	*plaid_mono_td*	Dense	PLAID	TD	Chinese ColBERT with PLAID

ColBERT-X model using these hard passages in the document language, queries in English, and scores from the mT5 reranker with a KL Divergence loss. Translate-Distill is an extension of the ColBERT-X Translate-Train [13] approach that distills ranking knowledge from a reranker instead of learning from the contrastive labels.

3.2 Multilingual Translate-Train

With an eye toward multilingual retrieval (MLIR), we would like the model to be capable of retrieving documents across a set of languages. MTT [9] generalizes TT by translating the training documents into each target language; this gives the model the ability to retrieve content expressed in any target language. A key to making MTT successful is to include documents from every target language in each batch.

Unlike training one ColBERT CLIR model with TT for each language pair (resulting in three models), we apply the same ColBERT MLIR model trained with MTT to all three language pairs simultaneously. This produces a single model capable of participation in all of the NeuCLIR tasks.

4 SPARSE RETRIEVAL

4.1 Probabilistic Structured Queries

Probabilistic Structured Queries (PSQ) [3] is a translation approach that probabilistically matches a token from one language to a distribution of tokens in another. This technique can be used to translate queries, documents, or both. Prior work [19] has concluded that mapping documents to the query language at indexing time achieves the best effectiveness while minimizing query latency. The resulting documents are bags of probabilistic tokens in the query language. They can be indexed as ordinary documents in a sparse retrieval model such as BM25 and HMM with real-valued weights.

Table 2: Topics where dates were introduced as a metadata filter to the topic

topic id	start	end
203	3/23/2021	3/29/2021
207	9/21/2020	
220	4/7/2018	4/7/2018
226	6/6/2019	6/6/2019
231	11/30/2018	11/30/2018
232	1/6/2018	1/14/2018
238	11/27/2020	
240	12/12/2019	
244	12/11/2017	
245	12/12/2019	
247	3/21/2018	
249	7/12/2019	7/12/2019
253	4/22/2019	
255	12/2018	
256	3/29/2018	
257	4/15/2019	
260	3/8/2014	
264	7/27/2018	
265	4/1/2020	4/1/2020
266	2/26/2017	2/26/2017
267	1/1/2018	1/31/2019
273	3/11/2018	
274	3/11/2021	

Our submission uses PSQ to translate the documents, and uses a Hidden Markov Model (HMM) [20] for retrieval.

4.2 Patapsco BM25 Retrieval

The Patapsco framework [1] supports CLIR lexical retrieval through Pyserini [10]. Patapsco ensures that language-specific processing is consistent for both queries and documents. The HLTCOE team submitted BM25 monolingual runs that used human-translated queries to search documents in their native language (QHT), CLIR runs that used the track-provided machine query translations to search the native documents (QGT), and runs that used English queries to search the track-provided document translations (DT). All languages used spaCy [5] for tokenization. For Russian and English machine translation, spaCy also provided stemming, while Parsivar [11] was used for Persian stemming. (We did not stem Chinese.) We explored three query variants: title, description, and title+description. We also explored the addition of ten RM3 expansion terms.

5 RESULTS

5.1 CLIR Tasks

The CLIR runs submitted by the HLTCOE are summarized in Tables 3 (non BM25) and 4 (BM25). The Translate-Distill models (c2) performed substantially better than the ColBERT-X models trained with Translate-Train (c6). The mT5-distilled models also outperformed the monolingual ColBERT model that indexes the translated documents (c4). Note that Run (c5) is trained with the PLAID training implementation using both training queries and passages in the original document language, i.e., a monolingual model in the document language. That it underperforms the Translate-Train models indicates either that the human translation of the queries is bad (unlikely) or that training is suboptimal.[3]

The student ColBERT-X models (c2) perform on par with their mT5 reranker (c3) teacher, but are much more efficient. Stacking the teacher reranker on top of the student model (c1) produces the most effective submission across all three languages. This gap suggests room for improvement in knowledge distillation training.

To investigate the stability and compatibility of the compressed document token representation in PLAID retrieval, run (c7) shards the collection by date using the ColBERT-X model. The degradation from the ColBERT-X retrieval (c6) suggests that temporal vocabulary shifts create incompatibilities in the decompressed document token embeddings used in the final scoring and ranking. It is also possible that the download date is not a good approximation of the creation date and negatively affects topics for which a date range was introduced, especially an end date since the download date by definition is after the document is created.

5.2 MLIR Task

The results of our MLIR submissions are summarized in Table 5. The most effective run among our submissions is the Multilingual Translate-Train (MTT) ColBERT-X model without using the PLAID vector compression (m1). Vector compression (m2) provides efficiency in both search time as well as disk space but sacrifices some effectiveness compared to the uncompressed version. Both runs using MTT ColBERT-X model outperform the English ColBERTv2 run that indexes the translated documents (m3), which provides a unified platform to compare scores across documents originating in different languages.

Run (m4) also shares the same MTT ColBERT-X model but sharding the collection by date with date-augmented topics. However, such argumentation provides the worst effectiveness.

5.3 Technical Document Task

The technical documents task represents a huge domain shift from news documents and from the MS MARCO training data, which in turn leads to different algorithm rankings. The HLTCOE submitted many of the same variants to this task as it did for the news tasks. The big differences were the experiments with dense monolingual retrieval represented by (t13), (t14), and (t15) and the addition of BLADE (t9, t10, t11) [14]. BLADE runs were submitted to the news track by the University of Maryland [12]. Table 6 shows that the mt5 cross-encoder (t1,t2) is superior to the bi-encoder (t3, t6-t12), just as in the news domain. However, in this case, reranking the BM25 document translation run (t18) outperforms reranking the PLAID translate train run (t7) for nDCG@20 despite the fact that without reranking PLAID translate-train performs better than document translation BM25 in both measures. The other attribute that stands out is that lexical matching with either PSQ (t4,t5) or BM25 with query translation (t19,t21) outperforms the ColBERT architecture

[3] This finding is consistent with the one documented in the working note of our participation of CIRAL@FIRE 2023.

Table 3: CLIR Results

		Persian		Russian		Chinese	
		nDCG	R@1k	nDCG	R@1k	nDCG	R@1k
(c1)	PLAIDkd-mT5gt-{td}	**0.547**	0.931	**0.540**	0.916	**0.485**	0.940
(c2)	PLAIDkd-monomt5tt-td	0.525	**0.936**	0.500	**0.932**	0.474	**0.951**
(c3)	mT5gt-{td}	0.519	0.878	0.529	0.838	0.483	0.863
(c4)	plaid_v2_eng_1	0.481	0.900	0.463	0.883	0.423	0.886
(c5)	*PLAID192mono-td*	0.466	0.822	0.424	0.830	0.404	0.867
(c6)	colbertX	0.495	0.873	0.488	0.896	0.437	0.923
(c7)	PLAID_shard_by_date_1bit_v1_tt	0.456	0.797	0.467	0.865	0.407	0.875
(c8)	PSQ-td	0.423	0.850	0.366	0.785	0.324	0.840
(c9)	PSQ-t	0.383	0.811	0.329	0.742	0.293	0.796

Table 4: BM25 Runs on CLIR Tasks.

				Persian		Russian		Chinese	
	Translation	Query	RM3	nDCG	R@1k	nDCG	R@1k	nDCG	R@1k
	CLIR								
(c10)	DT	T	✗	0.379	0.838	0.332	0.721	0.346	0.790
(c11)	DT	T	✓	**0.397**	**0.881**	0.321	0.771	0.380	0.861
(c12)	DT	D	✗	0.346	0.782	0.307	0.725	0.340	0.766
(c13)	DT	D	✓	0.357	0.842	0.316	0.777	0.361	0.820
(c14)	DT	TD	✗	**0.397**	0.839	0.363	0.766	0.368	0.805
(c15)	DT	TD	✓	0.385	0.878	**0.355**	**0.838**	**0.377**	**0.863**
(c16)	QGT	T	✗	0.283	0.660	0.280	0.638	0.262	0.677
(c17)	QGT	T	✓	0.302	0.748	0.279	0.714	0.295	0.746
(c18)	QGT	D	✗	0.303	0.702	0.276	0.667	0.269	0.661
(c19)	QGT	D	✓	0.308	0.760	0.300	0.764	0.288	0.768
(c20)	QGT	TD	✗	0.328	0.743	0.349	0.732	0.319	0.747
(c21)	QGT	TD	✓	0.328	0.781	0.315	0.792	0.328	0.822
	Monolingual Retrieval								
(c22)	QHT	T	✗	0.357	0.784	0.347	0.710	0.275	0.701
(c23)	QHT	T	✓	0.374	0.809	0.352	0.772	0.299	0.757
(c24)	QHT	D	✗	0.347	0.759	0.357	0.703	0.287	0.641
(c25)	QHT	D	✓	0.375	0.795	0.369	0.785	0.277	0.758
(c26)	QHT	TD	✗	**0.380**	**0.822**	0.405	0.754	0.329	0.733
(c27)	QHT	TD	✓	**0.380**	0.821	**0.406**	**0.815**	**0.344**	**0.800**

(t6,t7,t8) unless distillation is used (t3). This appears to indicate that while MS MARCO positive and negative examples are insufficient to train the model how to rank technical Chinese abstracts, distillation, which teaches the model how to score documents, is better able to train the model. PLAID over translated documents is a weaker model than translate-train; this is also different from the news domain, likely due to machine translation being less effective on technical documents. BLADE (t9, t10, t11) is also weaker than PSQ, but does outperform BM25 with document translation (t16,t17, t18). Finally, we unintentionally used a faulty training method with JH Polo (t15), which led to its dismal performance.

Turning to the BM25 runs, the poor performance of document translation especially compared to query translation is a big difference between the news domain and the technical documents domain. In the news domain, document translation has a fairly substantial edge, especially over machine query translation. For the technical abstracts, document translation is weakest (t16, t17, t18).

Among the monolingual runs, reranking dense retrieval (t13) is the most effective. Of particular note, recall at 1000 goes up from the base run (t6) to the reranked run (t1). This indicates that relevant documents were ranked at levels past 1000, which the reranking was able to recover. In addition, BM25 (t19) is nearly as effective

Table 5: MLIR Runs

		nDCG@20	R@1000
(m1)	colbertX	**0.362**	0.771
(m2)	plaid_v1_mtt_1bit	0.359	0.780
(m3)	plaid_v2_eng_1	0.355	**0.804**
(m4)	plaid_v1_mtt_1bit_date	0.335	0.720
(m5)	PSQraw-td	0.295	0.693
(m6)	PSQraw-t	0.256	0.667
	BM25 Runs with DT		
(m7)	T w/ RM3	0.288	0.750
(m8)	T w/o RM3	0.261	0.682
(m9)	D w/ RM3	0.282	0.712
(m10)	D w/o RM3	0.251	0.662
(m11)	TD w/ RM3	0.306	0.764
(m12)	TD w/o RM3	0.290	0.721

Table 6: Technical Document Track Results. Italicized names indicate monolingual runs.

		nDCG@20	R@1000
(t1)	rerank_mt5gt_td	**0.394**	0.656
(t2)	plaid_tt_mt5gt_td	0.383	0.755
(t3)	plaid_distilled_td	0.360	**0.824**
(t4)	psq_td_f32	0.314	0.768
(t5)	psq_t_f32	0.310	0.760
(t6)	colbert_x_td	0.287	0.693
(t7)	plaid_tt_td	0.279	0.717
(t8)	plaid_V2model_td	0.273	0.749
(t9)	blade-d	0.260	0.716
(t10)	blade-td	0.246	0.717
(t11)	blade-t	0.240	0.724
(t12)	plaid_jhpolo_td	0.072	0.212
(t13)	*plaid_monozh_mt5ht_td*	**0.410**	**0.812**
(t14)	*rerank_mt5ht_td*	0.378	0.722
(t15)	*plaid_mono_td*	0.359	0.758
	BM25 + RM3 Runs		
(t16)	DT + T	0.222	0.660
(t17)	DT + D	0.189	0.616
(t18)	DT + TD	0.210	0.656
(t19)	QGT + T	**0.322**	0.793
(t20)	QGT + D	0.290	0.755
(t21)	QGT + TD	0.314	**0.814**
(t22)	*QHT + T*	**0.350**	0.812
(t23)	*QHT + D*	0.276	0.722
(t24)	*QHT + TD*	0.331	**0.813**

as monolingual PLAID (t15), again indicating that MS MARCO positive and negative pairs are insufficient to train the model.

6 CONCLUSION

The HLTCOE team participated in all tasks offered in NeuCLIR 2023. While officially all runs were marked as manual because, as organizers, we created some of the topics, only the "date" runs where we introduce date metadata to each topic had any direct manual input. In general, runs reranked with mT5 outperformed end-to-end neural approaches, which in turn outperformed sparse retrieval models. Directions for future research include more investigation into the role dates play in topics as well as exploring other training strategies for MLIR.

REFERENCES

[1] Cash Costello, Eugene Yang, Dawn Lawrie, and James Mayfield. 2022. Patapsco: A Python Framework for Cross-Language Information Retrieval Experiments. In *Proceedings of the 44th European Conference on Information Retrieval (ECIR)*.

[2] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 985–988.

[3] Kareem Darwish and Douglas W Oard. 2003. Probabilistic structured query methods. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 338–344.

[4] 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*. 2288–2292.

[5] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. (2020). https://doi.org/10.5281/zenodo.1212303

[6] Vitor Jeronymo, Roberto Lotufo, and Rodrigo Nogueira. 2023. NeuralMind-UNICAMP at 2022 TREC NeuCLIR: Large Boring Rerankers for Cross-lingual Retrieval. *arXiv preprint arXiv:2303.16145* (2023).

[7] 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*. 39–48.

[8] Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldaini, and Eugene Yang. 2023. Overview of the TREC 2022 NeuCLIR Track. arXiv:2304.12367 [cs.IR]

[9] Dawn Lawrie, Eugene Yang, Douglas W Oard, and James Mayfield. 2023. Neural Approaches to Multilingual Information Retrieval. In *European Conference on Information Retrieval*. Springer, 521–536.

[10] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2356–2362.

[11] Salar Mohtaj, Behnam Roshanfekr, Atefeh Zafarian, and Habibollah Asghari. 2018. Parsivar: A Language Processing Toolkit for Persian. In *International Conference on Language Resources and Evaluation*. https://api.semanticscholar.org/CorpusID:21715688

[12] Suarj Nair and Douglas W. Oard. 2023. BLADE: The University of Maryland at the TREC 2023 NeuCLIR Track. In *Proceedings of The Sixteenth Text REtrieval Conference Proceedings (TREC 2023)*.

[13] Suraj Nair, Eugene Yang, Dawn Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W. Oard. 2022. Transfer Learning Approaches for Building Cross-Language Dense Retrieval Models. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I* (Stavanger, Norway). Springer-Verlag, Berlin, Heidelberg, 382–396. https://doi.org/10.1007/978-3-030-99736-6_26

[14] Suraj Nair, Eugene Yang, Dawn Lawrie, James Mayfield, and Douglas W. Oard. 2023. BLADE: Combining Vocabulary Pruning and Intermediate Pretraining for Scaleable Neural CLIR. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 1219–1229. https://doi.org/10.1145/3539618.3591644

[15] Rodrigo 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*. Association for Computational Linguistics, Online, 708–718. https://doi.org/10.18653/v1/2020.findings-emnlp.63

[16] 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*. Association for Computational Linguistics, Online, 5835–5847. https://aclanthology.org/2021.naacl-main.466

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

[18] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. 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*. Association for Computational Linguistics, Seattle, United States, 3715–3734. https://aclanthology.org/2022.naacl-main.272

[19] Jianqiang Wang and Douglas W. Oard. 2012. Matching meaning for cross-language information retrieval. *Information Processing & Management* 48, 4 (2012), 631–653. https://doi.org/10.1016/j.ipm.2011.09.003

[20] Jinxi Xu and Ralph Weischedel. 2000. Cross-lingual information retrieval using hidden Markov models. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. 95–103.

[21] Eugene Yang, Dawn Lawrie, James Mayfield, Douglas W Oard, and Scott Miller. 2024. Translate-Distill: Learning Cross-Language Dense Retrieval by Translation and Distillation. In *Advances in Information Retrieval: 46th European Conference on IR Research, ECIR 2024*.