HLTCOE at TREC 2024 NeuCLIR Track

Eugene Yang, Dawn Lawrie, Orion Weller, James Mayfield
Human Language Technology Center of Excellence, Johns Hopkins University, USA
{eugene.yang,lawrie,oweller2,mayfield}@jhu.edu

ABSTRACT

The HLTCOE team applied PLAID, an mT5 reranker, GPT-4 reranker, score fusion, and document translation to the TREC 2024 NeuCLIR track. For PLAID we included a variety of models and training techniques – Translate Distill (TD), Generate Distill (GD) and multilingual translate-distill (MTD). TD uses scores from the mT5 model over English MS MARCO query-document pairs to learn how to score query-document pairs where the documents are translated to match the CLIR setting. GD follows TD but uses passages from the collection and queries generated by an LLM for training examples. MTD uses MS MARCO translated into multiple languages, allowing experiments on how to batch the data during training. Finally, for report generation we experimented with system combination over different runs. One family of systems used either GPT-4o or Claude-3.5-Sonnet to summarize the retrieved results from a series of decomposed sub-questions. Another system took the output from those two models and verified/combined them with Claude-3.5-Sonnet. The other family used GPT4o and GPT3.5Turbo to extract and group relevant facts from the retrieved documents based on the decomposed queries. The resulting submissions directly concatenate the grouped facts to form the report and their documents of origin as the citations. The team submitted runs to all NeuCLIR tasks: CLIR and MLIR news tasks as well as the technical documents task and the report generation task.

KEYWORDS

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

1 INTRODUCTION

This year the HLTCOE's primary contribution was in experimentation with multiple ways to fine-tune mPLMs for CLIR. Specifically, we submitted a suite of ColBERT-X models, including Translate-Distill [18] and Generate-Distill (GD), Each model relied on an effective cross-encoder developed for the NeuCLIR 2022 track as the teacher [3], applying scores to English queries and documents. We compared Translate-Distill in the CLIR setting against the zero-shot setting where the model sees English queries and documents rather than English queries and non-English documents. The HLTCOE also experimented with Generate-Distill where the ColBERT-X model is fine-tuned using queries generated by an LLM, following the Translate-Distill approach for CLIR. A few of our submissions investigated the effect of reranked runs with the mT5 cross-encoder and GPT-4 with heapsort. Finally, we implemented token pooling [**?**] in PLAID, reducing the number of tokens in the index by half. Table 1 summarizes our CLIR runs in the news domain; Table 2 summarizes our MLIR runs; and Table 3 summarizes our technical documents runs.[1]

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

The HLTCOE also participated in the pilot report generation task. We produced two families of systems. The first proposed a summarization-based approach to decompose the original report topic into a variable length number of subquestions. Those subquestions then were used to find relevant documents through the search service provided by the track coordinators. Then all subquestions and results were presented to the language model (either Claude-3.5-Sonnet or GPT-4o). A meta-system combined the results from both GPT-4o and Claude-3.5-Sonnet with the retrieved documents, fact checking and updating both of them.

The other family extracted relevant facts from the retrieved documents. This family also decomposed the topic into multiple queries and use the search service to retrieve candidate documents. The facts extracted from the documents are grouped into canonical facts. The canonical fact descriptions are directly used as the report sentence and the documents that contributed to each group are used as the citations.

In the rest of this paper, we describe our systems, submissions, and observations.

2 MT5 RERANKER

At NeuCLIR 2022, the most effective submission was a reranking model that used the MonoT5 model with mT5XXL, a 13 billion parameter model [3, 9]. This submission uses pointwise reranking by taking a softmax over the decoding probabilities of two specific tokens (`_false` and `_true` for mT5XXL) as the probability of the document being relevant given the query [12]. 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.

3 GPT-4 RERANKER

We used GPT-4o in two ways for our submissions. For the Technical documents track, we reranked the 30 top ranked documents from the run, using the same prompt used by Parry et al. [13]. For the news collection, we first had GPT-4 determine the most relevant passage in each of the top 30 documents in the run being used for reranking. Then we performed 4-ary heapsort [7] on these top 30 passages. Each passage contained 450 tokens. The prompt we used from this task was slightly modified from the one used by Parry et al. [13]:

> SYSTEM_MESSAGE You are an intelligent assistant that can identify the best passage based on its relevance to a query.
>
> PROMPT
> I will provide you with 5 passages in no particular order, each indicated by a numerical identifier in square brackets. Identify the best passage based on its relevance to this search query: {description}.

Table 1: HLTCOE CLIR runs over newswire collection. XXX indicates the language of the newswire collection.

	Run Name	Type	Model	Query	Description
(c1)	plaid_eqsynms_distill_engXXX	Dense	PLAID	TD	GD/TD with an equal number of generate queries and MS MARCO queries
(c2)	kitchen_rankfuse.mt5rerank.gpt4rerank	Hybrid	multiple » mT5 » GPT	TD	Rerank top 30 c4 run with gpt4 using heapsort
(c3)	plaid_distill_engXXX.mt5rerank	Hybrid	PLAID » mT5	TD	Rerank c6 with mt5
(c4)	kitchen_rankfuse.mt5rerank	Hybrid	multiple » mT5	TD	Rerank c9 with mt5
(c5)	plaid_syn_distill_engXXX	Dense	PLAID	TD	Trained with GD
(c6)	plaid_distill_engXXX	Dense	PLAID	TD	TD on English query and translated MS MARCO documents
(c7)	plaid_distill_engeng_zs2engXXX	Dense	PLAID	TD	Distill on English MS MARCO, zero shot to English qry and native docs
(c8)	plaid_distill_engXXX.mt5rerank.gpt4rerank	Hybrid	PLAID » mt5 » GPT	TD	Rerank top 30 t3 run with gpt4 using heapsort
(c9)	kitchen_rankfuse	Hybrid	multiple	TD	ranked-based fusion over c5, c10, coord...-MNES-fast_psq_td, coord...-MNES-patapscoBM25qtRM3td, coord...-MTES-patapscoBM25dtRM3td, and ISI_SEARCHER-ANE_run1
(c10)	plaid_distill_engeng	Dense	PLAID	TD	Distill on English MS MARCO, indexing translated documents
(c11)	plaid_distill_mono_XXXXXX	Dense	PLAID	TD	TD on translated MS MARCO queries and documents, run with GT queries
(c12)	plaid_distill_engeng_zs2XXXXXX	Dense	PLAID	TD	Distill on English MS MARCO, zero shot to GT queries
(c13)	plaid_distill_engXXX_450p	Dense	PLAID	TD	TD on English query and translated MS MARCO documents, indexing passages of 450 tokens without overlap
(c14)	plaid_distill_engmlir	Dense	PLAID	TD	TD on English query and translated MS MARCO into all 3 NeuCLIR languages
(c15)	plaid_distill_engXXX_termpool2	Dense	PLAID	TD	TD on English query and translated MS MARCO documents, indexed with term pooling removing 1/2 tokens

> [1] ...
> [2] ...
> [3] ...
> [4] ...
> [5] ...
> General search topic: {title} Search Query: {description}.
> Identify the best passage above based on its relevance to the search query. The output format should be [], e.g., [4] meaning document 4 is most relevant to the search query. Only respond with the passage number; do not say any word or explain.

If during passage selection or a heapsort call GPT4 failed to return a document id, the first document was selected. For passage selection, this seemed reasonable because generally the main information in a news article is summarized in the beginning of the article. For heapsort this default may not always be optimal. For initial heap construction the decision is likely correct because the documents had already been sorted by the prior algorithm. However, this is less true when using heapify to rebuild the heap because heapsort is not a stable sorting algorithm. Heapsort was used to identify the top 20 passages. The remaining passages were left in an unsorted order since the primary measure used in NeuCLIR is nDCG@20.

4 COLBERT RETRIEVAL WITH PLAID

Our systems primarily use PLAID [15], an implementation of the ColBERT [6] retrieval architecture that encodes each token as a vector. Prior work on training CLIR dense retrieval models has demonstrated success augmenting training queries and passages with translation to match the CLIR target languages [11]. In the NeuCLIR 2023 track [10], our ColBERT-X models trained with Translate-Distill were the most effective end-to-end neural dense retrieval models, so that was the starting point for this year's submissions.

All ColBERT-X variants use the PLAID-X implementation with the title concatenated with the description as the query. Unless otherwise noted, we break each document into passages of 180 tokens with a stride of 90. 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 [1].

Table 2: HLTCOE MLIR runs over newswire collection.

	Run Name	Type	Model	Query	Description
(m1)	plaid_distill_mlir_bycoll_scorefuse	Dense	PLAID	TD	MTD with mixed entries indexing each language separately; fusing results by scores
(m2)	plaid_distill_clir.mt5rerank.scorefuse.gpt4rerank	Dense	PLAID » mT5 » GPT	TD	Reranked top 30 of run m10 with GPT-4
(m3)	kitchen_rankfuse.mt5rerank.scorefuse.gpt4rerank	Dense	multiple » mT5 » GPT	TD	Reranked top 30 of run mX with GPT-4
(m4)	plaid_distill_engeng	Dense	PLAID	TD	PLAID trained on English MS MARCO with a single index of translated documents
(m5)	plaid_distill_engeng_zs	Dense	PLAID	TD	PLAID trained on English MS MARCO with a single index of native documents
(m6)	plaid_distill_mlir_mixedentry	Dense	PLAID	TD	MTD PLAID trained with mixed entries using a single index of native documents
(m7)	plaid_distill_mlir_mixedpass	Dense	PLAID	TD	MTD PLAID trained with mixed passages using a single index of native documents
(m8)	plaid_distill_mlir_rr	Dense	PLAID	TD	MTD PLAID trained with round robin using a single index of native documents
(m9)	plaid_distill_mlir_mixedentry_termpool2	Dense	PLAID	TD	MTD PLAID trained with mixed entries using a single index of native documents indexed with term pooling removing 1/2 token
(m10)	plaid_distill_clir.mt5rerank.scorefuse	Dense	PLAID » mT5	TD	Reranked run m11 with mT5
(m11)	plaid_distill_clir_scorefuse	Dense	PLAID	TD	Fused scores of CLIR runs c5 for each language
(m12)	kitchen_rankfuse.mt5rerank.scorefuse	Dense	multiple » mT5	TD	Fused scores of CLIR runs c4 for each language

Table 3: HLTCOE runs over technical documents.

	Run Name	Type	Model	Query	Description
(t1)	plaid_eqsynms_distill_engzho	Dense	PLAID	TD	GD/TD with an equal number of generate queries and MS MARCO queries
(t2)	kitchen_rankfuse.mt5rerank.gpt4rerank	Hybrid	multiple » mT5 » GPT	TD	Rerank top 30 t4 run with gpt4 all at once
(t3)	plaid_distill_engzho.mt5rerank	Hybrid	PLAID » mT5	TD	Rerank t6 with mt5
(t4)	kitchen_rankfuse.mt5rerank	Hybrid	multiple » mT5	TD	Rerank t9 with mt5
(t5)	plaid_syn_distill_engzho	Dense	PLAID	TD	Trained with GD
(t6)	plaid_distill_engzho	Dense	PLAID	TD	TD on English query and translated MS MARCO documents
(t7)	plaid_distill_engeng_zs2engzho	Dense	PLAID	TD	Distill on English MS MARCO, zero shot to Eng. qry and native docs
(t8)	plaid_distill_engzho.mt5rerank.gpt4rerank	Hybrid	PLAID » mt5 » GPT	TD	Rerank top 30 t3 run with gpt4 all at once
(t9)	kitchen_rankfuse	Hybrid	multiple	TD	ranked-based fusion over t5, t10, coordinators-MNES-fast_psq_td, coordinators-MNES-patapscoBM25qtRM3td, coordinators-MTES-patapscoBM25dtRM3td, and ISI_SEARCHER-ANE_run1
(t10)	plaid_distill_engeng	Dense	PLAID	TD	TD on English MS MARCO, indexing translated documents
(t11)	plaid_distill_zhozho	Dense	PLAID	TD	TD on translated MS MARCO queries and documents, run with GT queries
(t12)	plaid_distill_engeng_zs2zhozho	Dense	PLAID	TD	Distill on English MS MARCO, zero shot to GT queries

4.1 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 [2, 14]. In our submissions, we explored training a ColBERT-X model to mimic the behavior of the powerful mT5 reranker. This training method, known as Translate-Distill, is described in more detail in Yang et al. [18].

Translate-Distill starts by selecting hard passages for each training query in the MS MARCO training set. We use an **English**

ColBERTv2 model [16] 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 ColBERT-X model using these hard passages in the document language, queries in English, and scores from the mT5-xxl reranker with a KL Divergence loss. Translate-Distill is an extension of the ColBERT-X Translate-Train [11] approach that distills ranking knowledge from a reranker instead of learning from the contrastive labels.

4.2 Generate-Distill for CLIR using mT5 Reranker

Generate-Distill [8] is aimed at overcoming domain mismatch between the training and search data and the training bottleneck caused by translation quality. This approach fine-tunes a CLIR retrieval model on the search corpus directly. Since we assume no queries or relevance judgments are available at training time, we use the model `Mixtral-8x7B-Instruct`[3] to generate training queries and the mt5 cross-encoders as a teacher model to provide their associated relevance signals for CLIR retrieval fine-tuning. Once the queries are generated, we follow the same process as Translate-Distill to train the model.

4.3 Token Pooling to Reduce Index Size

We submitted runs that integrated the technique of token pooling introduced by **?**] with PLAID-X. This technique pools a passage's tokens and uses k-means clustering to determine the representative vectors for the tokens. We submitted runs that reduced the number of tokens by half.

4.4 Multilingual Translate-Distill for MLIR using mT5 Reranker

We submitted the multilingual ColBERT-X models described in Yang et al. [17], which is an extended version of the Translate-Distill for CLIR [18]. We submitted all three variants of the Multilingual Translate-Distill (MTD), including mixed entry (`mixedentry`), mixed passages (`misedpass`), and round robin (`rr`). Each model is initialized with XLM-RoBERTa-Large and trained with translated MS MSMARCO provided by the coordinators [9]. The teacher scores are obtained by applying the mT5-xxl cross-encoder on the selected query-passage pairs, which are the same as those used in the CLIR version.

5 REPORT GENERATION SYSTEMS

We developed two families of systems: bottom-up and top-down.

Summarization-based Approach. The summarization-based approach used a language model with the initial topic as input, generated a persona for that topic, split the topic into a variable number of subqueries to find information, then aggregated that information using a language model. The prompt was tuned to help the model have an unbiased tone, cite properly, and not cite too little or too much. GPT-Researcher[4] was used as the framework for this approach.

We used two language models, GPT-4o-08-06 and Claude-3.5-Sonnet. These two models were chosen because of their strong performance in the NeuCLIR languages. The language model received all subqueries and the top ten retrieved documents for each subquery. It then summarized these elements into a report.

Finally, as these models can be prone to hallucination, we developed a meta-system that took in both reports as well as the subquestions and retrieved documents, and created a final report. This allowed for self-reflection and updating to ensure that the report correctly found all aspects in the retrieved documents.

The submission IDs of this approach are `{lang}-jhu-orion-aggregated-w-{gpt4o,claude}`.

Extractive Approach. The extractive approach also uses a language model to generate the final report. Despite conceptually being extractive, the pipeline still heavily uses the language model to "extract" information instead of using a traditional BIO-style extraction model. In this approach, we use the framework DSPy[4, 5] and the rely on the models GPT-4o-08-06 and GPT3.5Turbo.

Similar to the summarization-based approach, the extractive approach also first splits the topic into multiple queries. Specifically, the generative model is prompted to generate ten "Google-like" queries based on the report request. The queries are then issued to the search service provided by the coordinators to obtain the top three documents for each query.

For each retrieved documents, we prompted the generative model to extract key facts based on the report requests. The model then groups the extracted facts into twenty "canonical" facts from all retrieved documents with the document ID being attached to each group.

Since the evaluation only rewards individual sentences in scoring reports instead of favoring a coherent, human-readable report, we treat each canonical fact as the report sentence and the documents containing the fact as the citations.

The submission IDs of this approach are `{lang}-hltcoe-eugene-{gpt4o,gpt35turbo}`.

6 RESULTS

6.1 CLIR Tasks

The CLIR runs submitted by the HLTCOE are summarized in Table 4. The most effective run among our submissions is reranking using 4-ary heapsort. For Chinese and Persian, c2 was the most effective using a large variety of other runs. For Russian starting with TD PLAID in run c8 was the most effective. Systems in the second tier of algorithm performance were much less uniform by language; however, using passages of 450 tokens was always among them (c13). Interestingly, only for Chinese was reranking with mt5 more effective than just fusing many disparate runs (c4). Finally the introduction of Generate-Distill (c1) was most effective for Persian.

6.2 MLIR Task

The results of our MLIR submissions are summarized in Table 5. The most effective run among our submissions is m2, which relies on

[2] Yang et al. [18] finds that this configuration is not the optimal way to obtain scores. Instead English query/passage pairs should be used to obtain scores.
[3] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1
[4] https://github.com/assafelovic/gpt-researcher

Table 4: CLIR Results

	nDCG@20			R@1000		
	Persian	Russian	Chinese	Persian	Russian	Chinese
kitchen_rankfuse.mt5rerank.gpt4rerank	0.690	0.585	0.664	0.974	0.984	0.993
plaid_distill_engXXX.mt5rerank.gpt4rerank	0.676	0.592	0.650	0.917	0.968	0.947
kitchen_rankfuse.mt5rerank	0.619	0.509	0.584	0.974	0.984	0.993
plaid_distill_engXXX.mt5rerank	0.610	0.508	0.582	0.917	0.968	0.947
kitchen_rankfuse	0.629	0.531	0.569	0.974	0.984	0.993
plaid_distill_engXXX_450p	0.611	0.543	0.562	0.928	0.967	0.941
plaid_distill_engmlir	0.592	0.497	0.560	0.923	0.961	0.955
plaid_distill_engXXX_termpool2	0.600	0.504	0.555	0.916	0.963	0.941
plaid_distill_engXXX	0.607	0.508	0.552	0.917	0.968	0.947
plaid_distill_engeng	0.605	0.498	0.541	0.940	0.962	0.961
plaid_syn_distill_engXXX	0.578	0.490	0.540	0.915	0.954	0.954
plaid_distill_engeng_zs2XXXXXX	0.587	0.478	0.527	0.928	0.947	0.940
plaid_distill_mono_XXXXXX	0.574	0.493	0.520	0.912	0.952	0.929
plaid_distill_engeng_zs2engXXX	0.541	0.514	0.499	0.909	0.950	0.920
plaid_eqsynms_distill_engXXX	0.622	0.504	0.490	0.940	0.960	0.922

Table 5: MLIR Runs

	nDCG@20	R@1000
plaid_distill_clir.mt5rerank.scorefuse.gpt4rerank	0.460	0.876
kitchen_rankfuse.mt5rerank.scorefuse.gpt4rerank	0.459	0.899
plaid_distill_clir.mt5rerank.scorefuse	0.405	0.876
kitchen_rankfuse.mt5rerank.scorefuse	0.404	0.899
plaid_distill_mlir_mixedpass	0.383	0.877
plaid_distill_mlir_mixedentry_termpool2	0.377	0.871
plaid_distill_mlir_mixedentry	0.376	0.874
plaid_distill_engeng	0.373	0.875
plaid_distill_mlir_bycoll_scorefuse	0.365	0.875
plaid_distill_mlir_rr	0.360	0.862
plaid_distill_engeng_zs	0.297	0.827
plaid_distill_clir_scorefuse	0.296	0.799

GPT-4 to execute the heapsort algorithm over the results of PLAID TD. Just below the GPT-4 reranking is the mT5 re-ranking, which may indicate that for MLIR to cope with documents in multiple languages it is advantageous to view the query and document at the same time.

6.3 Technical Documents 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. Table 6 shows that GPT4 is more effective than the other techniques. After reranking, combining TD and GD is the most effective of the dense retrieval approaches.

6.4 Report Generation Task

The HLTCOE submitted four runs per language to the report generation task. Table 7 contains the results. While the report request was the same for each language, the documents over which the report was generated varied by language. Overall, system ranking is consistent by language. Our extractive approach performed better than our abstractive approach in terms of the ARGUE score and nugget recall. This is not surprising, as the extractive approach was designed to favor the inclusion of more facts over producing a fluent report. Given the underlying LLMs used to assist the report creation process, GPT3.5 struggled with the task more than the other models, especially in terms of citation precision.

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

	nDCG@20	R@1000
plaid_distill_engzho.mt5rerank.gpt4rerank	0.455	0.838
kitchen_rankfuse.mt5rerank.gpt4rerank	0.454	0.919
kitchen_rankfuse.mt5rerank	0.434	0.919
plaid_distill_engzho.mt5rerank	0.429	0.838
plaid_eqsynms_distill_engzho	0.385	0.867
plaid_distill_engeng_zs2zhozho	0.379	0.851
kitchen_rankfuse	0.378	0.919
plaid_distill_zhozho	0.372	0.846
plaid_distill_engeng	0.370	0.834
plaid_distill_engzho	0.368	0.838
plaid_syn_distill_engzho	0.356	0.838
plaid_distill_engeng_zs2engzho	0.265	0.736

7 CONCLUSION

The HLTCOE team participated in all tasks offered in NeuCLIR 2024. Officially all runs were marked as manual because, as organizers, we had input into creating the document collections, running the annotation for the technical documents task, and writing the guidelines for the tasks. In general, runs reranked with GPT4 outperformed runs reranked with mT5, which in turn outperformed end-to-end neural approaches (although this general trend did not hold for CLIR in the news domain where mT5 ranking was not as helpful). Increasing the passage size to 450 tokens and not overlapping the

Table 7: Report Generation Task Scores.

Language	Run ID	ARGUE Score	Citation Precision	Nugget Recall	Nugget Support	Sentence Support
Chinese	zho-hltcoe-eugene-gpt4o-fixed	0.726	0.859	0.327	0.298	0.840
	zho-hltcoe-eugene-gpt35turbo	0.587	0.813	0.250	0.248	0.720
	zho-jhu-orion-aggregated-w-claude	0.528	0.900	0.177	0.238	0.662
	zho-jhu-orion-aggregated-w-gpt4o	0.496	0.899	0.182	0.237	0.636
Persian	fas-hltcoe-eugene-gpt4o	0.872	0.918	0.303	0.311	0.919
	fas-hltcoe-eugene-gpt35turbo	0.646	0.846	0.215	0.201	0.792
	fas-jhu-orion-aggregated-w-claude	0.519	0.945	0.213	0.268	0.650
	fas-jhu-orion-aggregated-w-gpt4o	0.449	0.941	0.220	0.272	0.565
Russian	rus-hltcoe-eugene-gpt4o	0.808	0.904	0.339	0.420	0.874
	rus-hltcoe-eugene-gpt35turbo	0.602	0.799	0.313	0.309	0.715
	rus-jhu-orion-aggregated-w-claude	0.519	0.902	0.255	0.323	0.673
	rus-jhu-orion-aggregated-w-gpt4o	0.415	0.904	0.247	0.287	0.495

passages led to a clear performance boost, while decreasing the size of the index by half. Reducing the index size using token pooling was less effective and slowed indexing time substantially. For report generation, optimizing the report to include as many facts as possible is beneficial. Directions for future research include more investigation into training CLIR and MLIR models using generate-distill.

REFERENCES

[1] 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.

[2] 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.

[3] 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).

[4] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-Search-Predict: Composing Retrieval and Language Models for Knowledge-Intensive NLP. *arXiv preprint arXiv:2212.14024* (2022).

[5] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. In *The Twelfth International Conference on Learning Representations*.

[6] 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.

[7] Dawn Lawrie, Efsun Kayi, James Mayfield, Eugene Yang, Andrew Yates, and Douglas W. Oard. 2025. A Reproducibility Study of LLM Setwise Reranking using Heapsort. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 10 pages. https://doi.org/10.1145/3726302.3730338

[8] Dawn Lawrie, Efsun Kayi, Eugene Yang, James Mayfield, Douglas W. Oard, and Scott Miller. 2025. Generate-Distill: Training Cross-Language IR Models with Synthetically-Generated Data. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 6 pages. https://doi.org/10.1145/3726302.3730201

[9] Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldanini, and Eugene Yang. 2022. Overview of the TREC 2022 NeuCLIR Track. In *The Thirty-first Text REtrieval Conference (TREC 2022) Proceedings*.

[10] Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldanini, and Eugene Yang. 2023. Overview of the TREC 2023 NeuCLIR Track. In *The Thirty-second Text REtrieval Conference (TREC 2023) Proceedings*.

[11] 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

[12] 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

[13] A. Parry, S. MacAvaney, and D. Ganguly. 2024. Top-Down Partitioning for Efficient List-Wise Ranking. In *Proceedings of ReNeuIR'24*. Washington, DC.

[14] 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

[15] 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.

[16] 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

[17] Eugene Yang, Dawn Lawrie, and James Mayfield. 2024. Distillation for Multilingual Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2368–2373.

[18] 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*.