BLADE: Combining Vocabulary Pruning and Intermediate Pretraining for Scaleable Neural CLIR

Suraj Nair
University of Maryland
College Park, MD, USA
srnair@cs.umd.edu

Eugene Yang
HLTCOE, Johns Hopkins University
Baltimore, MD, USA
eugene.yang@jhu.edu

Dawn Lawrie
HLTCOE, Johns Hopkins University
Baltimore, MD, USA
lawrie@jhu.edu

James Mayfield
HLTCOE, Johns Hopkins University
Baltimore, MD, USA
mayfield@jhu.edu

Douglas W. Oard
University of Maryland
College Park, MD, USA
oard@umd.edu

ABSTRACT

Learning sparse representations using pretrained language models enhances the monolingual ranking effectiveness. Such representations are sparse vectors in the vocabulary of a language model projected from document terms. Extending such approaches to Cross-Language Information Retrieval (CLIR) using multilingual pretrained language models poses two challenges. First, the larger vocabularies of multilingual models affect both training and inference efficiency. Second, the representations of terms from different languages with similar meanings might not be sufficiently similar. To address these issues, we propose a learned sparse representation model, BLADE, combining vocabulary pruning with intermediate pre-training based on cross-language supervision. Our experiments reveal BLADE significantly reduces indexing time compared to its monolingual counterpart, SPLADE, on machine-translated documents, and it generates rankings with strengths complementary to those of other efficient CLIR methods.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

Sparse representation learning, neural CLIR, multilingual LM

ACM Reference Format:
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 (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539618.3591644

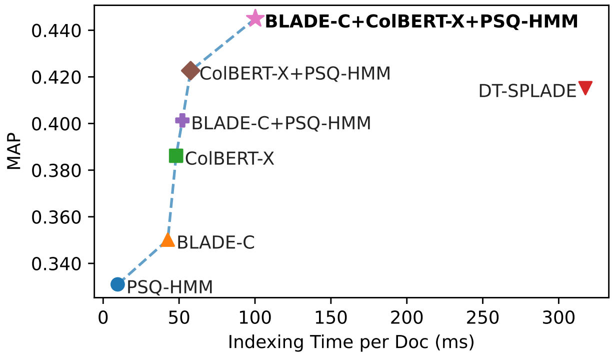

Figure 1: The tradeoff between MAP and indexing times, averaged over six CLIR collections, using English title queries.

1 INTRODUCTION

Creating sparse representation models using Pretrained Language Models (PLMs) such as BERT has proven to be effective and relatively efficient for monolingual retrieval, particularly with English content (where MS MARCO [5] can be used for fine-tuning). The core idea behind these methods, of which SPLADE [20] is presently best, involves projecting queries and documents onto sparse vectors in a high-dimensional space defined by the PLM vocabulary. This represents query and document terms using a small number of features that can be stored in a standard inverted index to support efficient ranking at query time. Lexical expansion can be done by assigning non-zero weights to terms that did not actually appear in the text but could have, thereby addressing some of the vocabulary mismatch faced by lexical matching models such as BM25. In this paper, we present BLADE, a Bilingual Lexical And Document Expansion model for Cross-Language Information Retrieval (CLIR), where text in one language is retrieved by a query in another language. In particular, we focus on the unique challenges Multilingual PLMs (MPLMs) pose when building a sparse neural CLIR model.

Our focus on techniques using MPLMs, rather than on first using Machine Translation (MT) to convert each document into the query language, is driven by our concern for efficiency, not just at query time but also at indexing time. As Figure 1 illustrates, BLADE is roughly an order of magnitude faster at indexing time than MT, as seen by the horizontal gap between DT-SPLADE (which uses MT in indexing) and BLADE-C (which does not). In this paper, we focus on this trade-off between scalability to larger collections on the one hand, which we operationalize as the time required for indexing, and retrieval effectiveness on the other, which we

SIGIR '23, July 23–27, 2023, Taipei, Taiwan
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3591644

measure as Mean Average Precision (MAP) or Recall@100 (R@100). The main contributions of this paper are:

(1) We propose a new CLIR model, BLADE, that uses a pruned mBERT vocabulary to improve training and inference.[1]
(2) We introduce a pretraining step to learn cross-language term associations from aligned text.
(3) We show that pretraining on comparable corpora yields results comparable to pretraining on parallel corpora.[2]
(4) We analyze the trade-offs between the indexing speed and ranking effectiveness of different methods.

2 SPARSE NEURAL INDEXING

In this section, we review prior work on neural methods for improving sparse neural indexing. Neural cross-encoder methods, while being highly effective, might not be suitable for interactive use with large collections. This has led to two broad lines of work. In one approach, neural methods are used to enhance the indexing vocabulary, improving the recall of efficient first-stage search techniques that rely on an inverted index [14, 51]. Less efficient techniques can then be used in subsequent stages to rerank highly ranked documents from first-stage retrieval. In the second approach, query and document processing is separated, so only the query representation needs to be computed at query time [30, 31]. Both approaches move some processing to indexing time, thus generating a trade space in which training speed, indexing speed, query latency, and ranking effectiveness are in tension. The techniques we compare in this paper all have query latency short enough for interactive use and have been well-analyzed in monolingual applications, including for learned sparse models [34, 40–42, 44]. However, their indexing speed varies over a broad range, which is important because indexing speed impacts real world usability.

2.1 Lexical Expansion Models

Pre-BERT sparse neural retrieval models [72] generated query and document vectors using L1 regularization to permit inverted indexing that is efficient at query time. The advent of BERT led to different forms of neural ranking models, including those that generate sparse weights for query and document terms [4, 10, 14, 20, 22, 38, 43, 74]. These models can be applied to generate weights only for terms that appear in the text, or in a more general lexical expansion setting that allows weights for terms that do not appear in the text. Working with cross-language input sequences renders the first approach unsuitable, so instead we focus on the lexical expansion setting. In a monolingual lexical expansion setting, existing approaches either rely on off-the-shelf document expansion models such as doc2query [50] or TILDEv2 [73] to generate additional terms, or they use the vocabulary space of the PLMs for expansion [4]. Among the models built off of the latter framework, SPLADE [20, 22] has been shown to generalize to both in- and out-of-domain task settings, and therefore, we choose SPLADE as the inspiration to design its cross-language cousin, BLADE.

2.2 SPLADE

SPLADE [20, 22] is a bilingual lexical expansion model that generates $|V|$-dimensional term vectors for queries and documents, where the weights represent the relative importance of each term. Given a query q and document d, SPLADE, initialized with a PLM η, computes the similarity score $s(q, d)$ between them as:

$$s(q, d) = \eta(q)^T \eta(d) \tag{1}$$

Let $V_{\mathcal{T}}$ denote the output vocabulary space of the SPLADE model. For a given document text sequence t of length N, SPLADE uses the masked language model head (MLM) from the pretrained encoder to get term weights for every document subword. Specifically, for a document (or a query) subword t_i, the model generates the term weights w_{ij} for the candidate output subword $t_j \in V_{\mathcal{T}}$ as:

$$w_{ij} = \phi(h_i)^T e_j + b_j \tag{2}$$

where ϕ is a combination of a linear layer with GeLU activation, with LayerNorm applied to the contextualized embedding h_i of t_i. Here e_j is the j-th row of the decoder matrix of the Language Model (LM) head, and b_j is the token-level bias.

Once we have $|V|$-dimensional vectors for each subword in the document, an aggregated vector for the document is generated by max pooling over the target vocabulary dimensions as:

$$w_j = \max_i \log\left(1 + \mathrm{ReLU}(w_{ij})\right) \tag{3}$$

A similar explanation follows for generating aggregate query vectors. Given the aggregate query and document vectors, the similarity score can be computed using Eq. (1). Sparsity is enforced in the document and query representations by combining ReLU activation and FLOPS [55] regularization.

SPLADE [22] uses a contrastive ranking loss to train a retrieval model. Given a query q_i, a relevant document d_i^+, a BM25 sampled non-relevant document d_i^-, and in-batch documents $\{d_j^-\}$ that we treat as not relevant, the contrastive ranking loss is:

$$\mathcal{L}_{\text{rank}} = -\log \frac{e^{s(q_i, d_i^+)}}{e^{s(q_i, d_i^+)} + e^{s(q_i, d_i^-)} + \sum_j e^{s(q_i, d_j^-)}} \tag{4}$$

In SPLADE, the ranking loss was trained on MS MARCO training triples. Subsequent versions [20, 21] introduced distillation loss and a hard negative mining step. SPLADEv2 [20] leverages Margin-MSE loss [26] for distilling knowledge from a teacher cross-encoder trained on a monolingual corpus to the student SPLADE model. Once a model is trained using a combination of ranking and distillation loss, SPLADEv2 additionally mines for harder negatives using the trained model to conduct another round of training.

2.3 SPLADE-X

SPLADE-X [49] generalizes the SPLADE model for CLIR applications. SPLADE-X uses a multilingual BERT encoder to generate aggregate term vectors in a similar manner to SPLADE, given a query and document in different languages. One difference, however, is the use of a top-k masking [70] scheme as a way to induce sparsity instead of FLOPS regularization. This technique only preserves the dimensions corresponding to the top-k terms with the highest weights and sets the remaining weights to zero.

[1] https://github.com/hltcoe/BLADE
[2] https://github.com/hltcoe/BLADE/bi-passages

SPLADE-X uses a popular cross-language transfer learning approach known as *translate-train* to learn term associations for CLIR matching. English queries are paired with passages machine-translated from English to the document language, and a contrastive ranking loss is learned as in Eq. (4). This approach relies on machine-translated MS MARCO corpora to generate these pairs, specifically mMARCO [8], and multiple neural CLIR models have been trained on the same or similar corpora [27, 36, 48]. Furthermore, SPLADE-X uses a multilingual distillation approach, where a monolingual SPLADE model is chosen as the teacher to distill the knowledge to a multilingual SPLADE-X. Instead of Margin-MSE loss as in SPLADEv2, SPLADE-X minimizes a KL-divergence loss to match the probability distribution coming from the teacher and student models, as introduced in Yang et al. [70]. SPLADE-X does not include SPLADEv2's hard negative mining step.

3 BLADE

The architecture of BLADE is derived from its monolingual counterpart, SPLADEv2 [20], and its cross-language variant, SPLADE-X. We preserve many of the modeling choices from SPLADE-X, but we modify, a) vocabulary pruning; and b) intermediate pretraining.

3.1 Vocabulary Pruning

Let $V_{\mathcal{Q}}$ and $V_{\mathcal{D}}$ be the subword vocabularies of the query and document languages, respectively. In the original SPLADE, the output vocabulary was the vocabulary of a monolingual BERT language model, i.e., $V_{\mathcal{T}} = V_{\mathcal{Q}} = V_{\mathcal{D}}$ with $|V_{\mathcal{T}}| = 30522$. Vocabulary sizes of multilingual PLMs such as mBERT (119k dimensions) or XLM-R (250k dimensions) are much larger, presenting several challenges. First, the increased number of dimensions impacts sparsity and, thus, the efficiency of the model. Additionally, the larger vocabulary $|V|$ leads to increased memory use during training, and higher inference costs at indexing time and at query time. Vocabulary selection has been well studied in the context of MT [18, 28, 61] to trade between translation latency and output quality, as measured using automatic metrics such as BLEU [54].

To address these issues, SPLADE-X [49] limited the output to query-language terms (i.e., $V_{\mathcal{T}} = V_{\mathcal{Q}}$). That essentially made SPLADE-X an encoder-only MT model that translated document language terms to query language terms, albeit with overgeneration and without a target language model. For BLADE, we opt instead for a pruned bilingual language model [1], mBERT$_{\text{en-xx}}$. This bilingual model consists of a pruned mBERT vocabulary corresponding to the subword terms in both the query and document languages (i.e., $V_{\mathcal{T}} = V_{\mathcal{Q}} \cup V_{\mathcal{D}}$). Only embeddings corresponding to the pruned vocabulary are kept; all others are discarded. This reduces model size, as most parameters of PLMs are stored in the input/output embedding matrix. Across the six document languages we use for evaluation, the reduction in vocabulary size leads to, on average, a 36.5% reduction in the number of parameters relative to the original mBERT model. With an effective batch size of 128 on 8 V100 GPUs, total training time is reduced by 30%. With a batch size of 64 on one V100 GPU, the reduction in inference time averages 55%.

3.2 Intermediate Pretraining

In CLIR, vocabulary mismatch between queries and documents poses a significant challenge for multilingual PLMs. To match terms in different languages, MPLMs must generate similar representations for words in different languages that have similar meanings. Previous studies have shown that off-the-shelf MPLMs are often undertrained, and thus require fine-tuning for the CLIR task [39]. To address this, SPLADE-X used *translate-train* to learn cross-language term associations from translated mMARCO pairs. However, machine-generated translations can introduce *translationese* [66], which has been shown to affect cross-language transfer due to translation artifacts [3]. Using a translate-train approach, the model learns term associations based only on translated document texts rather than from what would have been their natural written forms. To address this limitation, we propose an intermediate pretraining step that uses aligned text pairs in the query and document languages, both expressed in more natural written forms.

3.2.1 Intermediate Pretraining Objective. Consider a set of aligned pairs $[(P^{\mathcal{Q}}_1, P^{\mathcal{D}}_1), (P^{\mathcal{Q}}_2, P^{\mathcal{D}}_2), \ldots, (P^{\mathcal{Q}}_n, P^{\mathcal{D}}_n)]$ in languages $\mathcal{Q}$ and $\mathcal{D}$. We compute contrastive ranking loss similarly to Eq. (4). Treating $P^{\mathcal{Q}}_i$ as the query, $P^{\mathcal{D}}_i$ as the relevant document, and a set of in-batch documents $P^{\mathcal{D}}_j$ that we treat as non-relevant, we model the loss as:

$$\mathcal{L}^{\mathcal{Q}\mathcal{D}}_{\text{CO}} = -\log \frac{e^{s(P^{\mathcal{Q}}_i, P^{\mathcal{D}}_i)}}{e^{s(P^{\mathcal{Q}}_i, P^{\mathcal{D}}_i)} + \sum_j e^{s(P^{\mathcal{Q}}_i, P^{\mathcal{D}}_j)}} \tag{5}$$

The similarity score s is computed using Eq. (1). With this pretraining objective, an off-the-shelf mPLM can use aligned human-written document-language and query-language texts to learn cross-language term associations. This can serve as a complementary source of knowledge in contrast to relying solely on machine-translated passages with the translate-train approach.

We use a Whole Word Masking (WWM) loss in both languages $\mathcal{Q}$ and $\mathcal{D}$, denoted as $\mathcal{L}^{\mathcal{Q}}_{\text{WWM}}$ and $\mathcal{L}^{\mathcal{D}}_{\text{WWM}}$, respectively. WWM masks all subwords for a given word, in contrast to the commonly used MLM that only masks subwords which sometimes are only part of a whole word.[3] Our overall pretraining loss $\mathcal{L}_{\text{pretrain}}$ is:

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}^{\mathcal{Q}\mathcal{D}}_{\text{CO}} + \mathcal{L}^{\mathcal{Q}}_{\text{WWM}} + \mathcal{L}^{\mathcal{D}}_{\text{WWM}} \tag{6}$$

As a design choice, we only update the model parameters associated with the MLM head, keeping the remaining parameters frozen. Our motivation was to avoid the catastrophic forgetting problem of neural networks in general by limiting the number of parameters we need to update, thereby preserving the original knowledge from pretraining. We also tried updating all the model's parameters but that did not provide any downstream effectiveness gain.

4 EXPERIMENT SETUP

We describe our experiment setup, including test collection, evaluation method, implementation details, and baselines.

4.1 Test Collections & Evaluation

The test collections used in our experiments are from the CLEF 2003 multilingual ad-hoc retrieval track [9] for documents in French (FR),

[3] For Chinese, we use LTP (https://github.com/HIT-SCIR/ltp) to segment words.

Table 1: Test collection statistics. Queries are English with at least one relevant doc, Passages are as split for BLADE, MT Passages are splits of English translations for DT-SPLADE.

	CLEF 03				NeuCLIR 22	
	FR	IT	DE	ES	ZH	RU
Queries	52	51	56	57	47	44
Documents	130K	158K	295K	454K	3,179K	4,628K
Passages	0.5M	0.6M	1.3M	2.1M	18.3M	21,6M
MT Passages	0.5M	0.6M	1.1M	1.7M	13.7M	16.8M

Italian (IT), German (DE), and Spanish (ES), and from the TREC 2022 NeuCLIR track[4] for documents in Chinese (ZH)[5] and Russian (RU) [35]. In every case, we use the English title field as the query, which produces queries with lengths typical of a Web search. Table 1 provides collection statistics. To evaluate effectiveness, we focus on Mean Average Precision (MAP) and Recall@100 (R@100). For significance testing, we use a paired t-test ($p < 0.05$) with Holm-Bonferroni multiple test correction for the difference in means.

4.2 Parallel and Comparable Corpora

We explore two sources of aligned text: (1) parallel texts, which are direct translations; and (2) comparable texts, which convey similar meanings but may not be direct translations. For sentence-aligned parallel text, we use a diverse range of OPUS [64] corpora, including from EuroParl [33], GlobalVoices,[6] MultiUN [19], NewsCommentary,[7] QED [2], TED [57], UNPC [75], and WMT-News [6].

Prior work has primarily used parallel corpora of aligned sentences to train MT systems. However, the limited context present in aligned sentences may be suboptimal for fine-tunin a PLM for CLIR. To test that hypothesis, we created a new passage-aligned parallel corpus. For each source of bitext, we obtain the original monolingual corpora in the query and document languages, along with the sentence alignment file. We then generate a list of aligned sentences within these documents using the information provided in the alignment file.[8] We then construct overlapping passages, where each passage is defined as a set of consecutive sentences from the list of aligned sentences. To ensure homogeneity in the lengths of aligned passages, we select consecutive sentences such that the total number of subword tokens does not exceed the maximum sequence length (256). We follow a similar process to move the stride by selecting the first sentence beyond 128 subword tokens.

To create aligned comparable passages, we start with CLIRMatrix [63], a collection built using Wikipedia's inter-language links. CLIRMatrix, originally designed for evaluating CLIR systems, pairs the title of a Wikipedia article in one language (which modeled a query) with a ranked list of passages from Wikipedia pages in some other language on the same topic (which modeled relevant documents). Passages average about 200 whitespace-separated tokens for non-CJK languages; Chinese passages are roughly 600 characters. Following the procedure used by Yang et al. for C3 [69], for each language pair *en-xx*, we identify the highest ranked non-English passage with a score value 6 in *xx* for every *en* query and then align them with the corresponding highest ranked passage in *en*. The two passages are then aligned, and the page title used to align them is discarded. Table 2 shows corpus statistics.

Table 2: Statistics of aligned pairs.

	FR	IT	DE	ES	ZH	RU
Parallel Sentences	53.4M	3.2M	3.5M	45.9M	31.2M	43.2M
Parallel Passages	18.2M	1.0M	1.2M	15.8M	11.6M	17.0M
Comparable Passages	1.2M	1.0M	1.2M	1.0M	0.6M	0.8M

4.3 Implementation Details

We implement BLADE using the Tevatron toolkit [24], which is built on top of the HuggingFace Transformers [67] library. To initialize BLADE, we use a smaller bilingual language model, released by Geotrend,[9] which thus defines our pruned bilingual vocabulary. For task-specific fine-tuning, we adopt a translate-train approach, using English queries paired with translations produced using Google MT that are distributed as mMARCO [8].[10] We perform 100,000 steps of fine-tuning with an effective batch size of 256 using 8 V100 GPUs and a learning rate of 1e-5 with the Adam [32] optimizer. Our maximum query length is 32 tokens, and passage lengths are limited to 256 tokens. Our BLADE implementation differs from that described in the SPLADEv2 paper [20] in that like SPLADE-X for training we use in-batch negative samples rather than the noise contrastive estimation process for mining hard negative training examples. As our DT-SPLADE comparison below indicates, the effect of this change is small (on the order of 2%).

For intermediate pretraining, we use parallel or comparable passages or parallel sentences, with the pretraining objective in Eq. (6). We pretrain the model for 200,000 steps with an effective batch size of 192 on 8 V100 GPUs and a learning rate of 1e-5 using Adam. When pretraining, the English passage is encoded as the query and the non-English passage is encoded as the document passage. The maximum passage length in each language is set to 256 tokens. In both intermediate pretraining and task-specific fine-tuning, we set k to 1% of the total vocabulary size of the corresponding Geotrend bilingual model. Also, we lowercase queries and documents for both intermediate pretraining and task-specific fine-tuning.

These configurations yield three BLADE variants: **BLADE-S** pretrained on parallel sentences; **BLADE-P** pretrained on parallel passages; and **BLADE-C** pretrained on comparable passages. All variants then receive task-specific fine-tuning. We refer to any BLADE model without this pretraining as vanilla BLADE.

For inference, we segment the documents into overlapping passages of 256 subword tokens with a stride of 128 subword tokens. We use the Anserini toolkit to index the top-k passage term weights generated by BLADE. We then perform retrieval using the indexed passages and queries generated by BLADE to generate a ranked list of passages. The final step uses MaxP [7, 15] score aggregation

[4] https://neuclir.github.io/

[5] We use the script provided by NeuCLIR organizers to convert traditional Chinese characters to simplified.

[6] https://casmacat.eu/corpus/global-voices.html

[7] https://data.statmt.org/news-commentary/v16/

[8] We drop sentences that have no aligned counterparts in the other language

[9] An example EN-DE model: https://huggingface.co/Geotrend/bert-base-en-de-cased

[10] Due to the limited number of queries in the test collections, we utilize an external collection with relevance judgments for task-specific fine-tuning, while reserving the test collections solely for evaluation purposes.

to generate a document ranking from a ranked list of passages.[11] The primary reason for selecting MaxP is to assess neural models, trained on passage-level judged collections like MS MARCO, using document-level judged collections such as CLEF or TREC.

4.4 Baselines

We implement the following baseline retrieval systems:

Non-Neural Baselines

- **HQT-BM25** is a monolingual baseline run in the document language, where the queries are human-generated translations of the English queries into the document language and the documents are in their native language. Specifically, we perform retrieval with the BM25 implementation [58] from the Anserini toolkit [71] with Anserini's default hyperparameters. We compute the indexing time using Anserini run on 48 threads.
- **PSQ-HMM** uses Probabilistic Structured Queries (PSQ) [16], a CLIR approach in which alternative term translations are used to estimate query language (i.e., English) term counts for each non-English document using translation probabilities for English terms given document-language terms. To obtain translation probabilities, we combine results from three alignment tools: GIZA++ [52], BerkeleyAligner [37], and Eflomal [53]. For each language pair, we train each aligner using parallel sentences from all sources listed in §4.2 except UNPC, and we also train on bilingual Panlex dictionaries [29].[12] We use the same preprocessing for the bilingual corpora as for the queries and the document collections: lowercasing tokens, removing punctuation and normalizing diacritics. We exclude translation probabilities of less than 1e-4 and then apply a cumulative distribution function threshold of 0.97. Given a vector of term counts in a document language, we generate a corresponding vector of English term counts at indexing time and then build an index based on those English counts. Note that this is an indexing-time implementation of the query-time implementation proposed in the original PSQ paper. The ranking is then performed using a Hidden Markov Model (HMM) [68] implementation.

Document Translation (DT) CLIR Baselines

- **DT-BM25** is a CLIR baseline run in the query language, where queries are in English and documents are machine translations of the text. The MT model is a 6-layer encoder/decoder transformer stack implemented using Sockeye 2 [17, 25] trained on publicly available bitext sources. Sockeye 2 has a decoding speed of 50 sentences/sec, roughly amounting to 285 ms per document averaged across the six document collections. Similar to HQT-BM25, we perform retrieval using Anserini and compute the indexing time by adding two factors: the time it takes to translate the documents and the indexing time using Anserini run on 48 threads.
- **DT-SPLADE** is a CLIR system built by applying SPLADEv2 with English queries to documents that have been automatically translated into English. We train a monolingual task-specific SPLADE model, initialized with an uncased BERT-Base encoder, using the same fine-tuning recipe as BLADE with the original MS MARCO triples. For a fair comparison with SPLADE-X and BLADE training, no hard negative mining step described in the original SPLADEv2 paper [20] was used. Our experiments on the four CLEF languages show a 2% drop in MAP using our version of the model compared to the publicly available SPLADEv2 checkpoint.[13] Indexing time is the combination of translation time, inference time per document to run the SPLADE model on a single V100 GPU, and Anserini's indexing time using 48 threads.

Other CLIR Baselines

- **ColBERT-X** [48] is a generalization of ColBERT [31] for CLIR. This belongs to a family of multi-vector dense retrieval models, which are generally more effective than single-vector dense retrieval models such as DPR [30]. ColBERT-X is a multi-stage retrieval model. The first stage finds the documents most similar to the query terms using an approximate nearest neighbor search. Then the second stage performs a MaxSim operation, which computes a term-by-term interaction matrix between the query and those documents. We use a ColBERT-X model initialized with an XLM-R Large [11] multilingual encoder and adopt a translate-train approach to fine-tune the model, using mMARCO passage translations (as generated by Bonifacio et al. [8] using a Marian MT model, referred to as Helsinki), each paired with an untranslated English MS MARCO query. We index the overlapping passages of 180 tokens with a stride of 90 using 8 V100 GPUs and compute the final per-GPU indexing time by multiplying the indexing time by the number of GPUs (8).
- **SPLADE-X** is the cross-language generalization of the SPLADE model initialized with a multilingual BERT model. We reimplement the translate-train version of SPLADE-X, primarily adhering to the design choices outlined by Nair et al. [49].[14]

We also report results from system combinations, using Reciprocal Rank Fusion (RRF) [12]. Because system combination has implications for both effectiveness and efficiency, this allows us to explore a broader range of options in that trade space. For fusion results, we report indexing time by summing the per-document indexing times of individual systems.

5 RESULTS AND ANALYSIS

This section describes the results of the experiments conducted and presents our analysis to answer the following research questions:

RQ1 How much improvement does the intermediate pretraining step contribute to BLADE's effectiveness?

RQ2 How does BLADE compare to CLIR baselines for retrieval effectiveness?

RQ3 What is the relative indexing speed of different CLIR methods?

RQ4 What is the trade-off of effectiveness and efficiency?

RQ5 Can we tune BLADE's query expansion to better balance retrieval effectiveness with query-time efficiency?

5.1 The Effect of Intermediate Pretraining

We analyze the effect of intermediate pretraining to answer RQ1 using results from Table 3. We start by comparing the sparse neural

[11] Besides MaxP, we experimented with aggregating passage weights using functions including sum, max, and average to calculate document-level weights; however, each of these aggregations yielded lower effectiveness than MaxP.

[12] We omit UNPC from the parallel corpora used to train PSQ because at 18M-30M sentence pairs it is far larger than is needed to obtain stable term translation probabilities.

[13] https://github.com/naver/splade/tree/main/weights/distilsplade_max

[14] Our key modification involves increasing the sequence length from 128 to 256 tokens.

Table 3: MAP and R@100 for BLADE model variants for retrieving content in 6 languages using English title queries

	CLEF 03								NeuCLIR 2022				Average	
	French		Italian		German		Spanish		Chinese		Russian			
Systems	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100
SPLADE-X	0.402	0.771	0.355	0.676	0.340	0.580	0.332	0.578	0.218	0.436	**0.270**	0.422	0.320	0.577
vanilla BLADE	0.434	0.767	0.361	0.675	0.340	0.574	0.152	0.345	0.244	0.465	0.050	0.177	0.264	0.501
BLADE-S	0.437	0.774	0.359	0.680	0.368	0.606	0.385	0.609	**0.266**	**0.487**	0.242	**0.454**	0.343	0.602
BLADE-P	**0.453**	0.763	0.341	0.677	0.378	0.598	**0.396**	0.618	0.264	0.475	0.233	0.437	0.344	0.595
BLADE-C	0.448	**0.783**	**0.389**	**0.730**	**0.386**	**0.634**	0.387	**0.640**	0.248	0.453	0.243	0.429	**0.350**	**0.612**

models SPLADE-X and vanilla BLADE. Vanilla BLADE, which lacks intermediate pretraining, is fine-tuned only on the task-specific loss; it differs from SPLADE-X in that it uses bilingual Geotrend embeddings rather than only query-language (English) embeddings in the output space. SPLADE-X has a higher MAP and R@100 than vanilla BLADE, on average across all test collections. Vanilla BLADE performs very similarly to SPLADE-X in three CLEF languages, French, Italian, and German, and numerically outperforms SPLADE-X in NeuCLIR Chinese. The only statistically significant differences are in Spanish and Russian, where using an off-the-shelf pruned bilingual model leads to a drop in effectiveness, indicating that the same fine-tuning process cannot achieve the desired output quality.

Now adding intermediate pretraining, we see BLADE-C (the best of our BLADE models, on average) improving over SPLADE-X in both MAP and R@100, averaging a 9% MAP improvement and a 6% R@100 improvement across all the languages. We similarly see improvements for BLADE-P and BLADE-S over SPLADE-X. These consistent differences indicate that intermediate pretraining and extending the vocabulary from SPLADE-X's query-language tokens to include tokens from both the query and document languages is beneficial. Intermediate pretraining on aligned passages accounts for a part of this difference, but including document-language terms is important, especially in Chinese. Most importantly, these gains in effectiveness are achieved with a reduction in model size from SPLADE-X to BLADE's pruned bilingual model.

We see that pretraining on comparable passages (BLADE-C) produces results broadly similar to training on parallel passages (BLADE-P), with each yielding better MAP than the other on three of the six languages. Only the improvement from using comparable rather than parallel passages in Italian is statistically significant. Similarly, we see that pretraining with parallel passages or parallel sentences yields similar results, with each achieving numerically better MAP than the other in three of the six languages; none of the differences are statistically significant. We focus the remainder of our analysis on BLADE-C for two reasons. First, BLADE-C's use of comparable passages offers greater potential for diversity that can be beneficial when combined using RRF with results from systems trained on parallel text (as all other systems are). Second, BLADE-C attains a higher average MAP and R@100 across the six languages compared to any other approach, establishing it as an equally suitable choice, if not better, than the alternatives.

We observe that intermediate pretraining using comparable passages numerically improves the MAP of the BLADE-C model in every language over the vanilla BLADE model. Compared to pretraining with comparable text, MAP degrades without pretraining by 25% on average. The reductions in MAP without pretraining for Spanish and Russian are particularly large, suggesting that the vanilla BLADE model for those languages may be less well-tuned. To confirm this, we randomly selected a Spanish sentence, replaced one of the original tokens with the [MASK] token, and checked the output from different models, including off-the-shelf mBERT/Geotrend and BLADE model variants. While the off-the-shelf and the other BLADE model outputs look reasonable (related terms or exact matches), vanilla BLADE outputs only punctuations. A similar phenomenon is observed in the case of Russian. We find this to be a case of representation degeneration [23], where the vanilla BLADE model defaults to expanding to rogue dimensions corresponding to those characters. Several solutions have been proposed for this issue, which includes normalizing/whitening embeddings [62] or using a regularization step [56]. The design of SPLADE-X avoids this issue, as it includes only alphanumeric characters in its vocabulary. However, intermediate pretraining acts as a form of regularization since we only update the LM head during pretraining. The differences are statistically significant for both MAP and R@100 in Spanish, where BLADE-C surpasses the effectiveness of SPLADE-X, and not in Russian, where SPLADE-X has numerically better MAP and R@100 than BLADE-C. This further underscores the importance of intermediate pretraining.

5.2 Optimizing for Effectiveness

Table 4 shows MAP and Recall@100 (R@100) for different methods across the six language pairs. Comparing the two non-neural baselines, we see PSQ-HMM performing comparably to HQT-BM25 on average, with the average MAP across all six collections numerically equal and the average R@100 slightly favoring PSQ-HMM. These broadly comparable results demonstrate that our PSQ-HMM framework is a strong non-neural baseline, indicating that the term expansion effect of using multiple translations in PSQ is (on average) sufficient to compensate for the more selective term choice of human translators who generate only a single translation of each query, which is then run without query expansion.

We have two CLIR baselines that rely on applying MT to every document at indexing time: DT-BM25 and DT-SPLADE. This is a computationally expensive approach, although it does have the benefit (which none of our other CLIR approaches share) of obviating the need to rapidly produce new translations when the user wishes to see a translation. With just one exception (MAP for Italian), DT-SPLADE yields numerically better retrieval effectiveness than DT-BM25 by both MAP and R@100, although the MAP difference is only significant for Russian, and the R@100 difference

Table 4: MAP and Recall@100 for retrieving content in 6 languages using English title queries.

	CLEF 03								NeuCLIR 2022				Average		Indexing
	French		Italian		German		Spanish		Chinese		Russian				Time per
Systems	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	doc (ms)
Non-Neural Baselines															
HQT-BM25	0.406	0.737	**0.387**	**0.720**	0.296	0.485	**0.431**	**0.695**	0.183	0.431	**0.281**	0.428	**0.331**	0.583	0.29
PSQ-HMM	**0.419**	**0.775**	0.325	0.632	**0.379**	**0.624**	0.374	0.606	**0.236**	**0.465**	0.253	**0.447**	**0.331**	**0.592**	9.60
DT Baselines															
DT-BM25	0.446	0.772	**0.421**	0.725	0.465	0.702	0.425	0.650	0.266	0.469	0.269	0.442	0.382	0.630	285.04
DT-SPLADE	0.486	**0.846**	0.418	0.731	**0.476**	0.756	0.448	0.670	0.310	0.576	**0.353**	0.552	0.415	0.689	317.61
+ PSQ-HMM	**0.487**	**0.846**	0.384	**0.779**	0.475	**0.757**	**0.472**	**0.726**	**0.328**	**0.578**	**0.353**	**0.578**	**0.417**	**0.711**	327.21
Other Neural Baselines															
ColBERT-X	0.457	0.765	0.404	0.710	0.408	0.643	0.381	0.621	0.332	0.542	0.335	0.521	0.386	0.634	48.05
+ PSQ-HMM	**0.496**	**0.826**	**0.413**	**0.750**	**0.465**	**0.696**	**0.439**	**0.710**	**0.363**	**0.594**	**0.360**	**0.574**	**0.423**	**0.692**	57.65
BLADE Family															
BLADE-C	0.448	0.783	0.389	0.730	0.386	0.634	0.387	0.640	0.248	0.453	0.243	0.429	0.350	0.612	42.59
+ PSQ-HMM	0.491	0.825	0.397	0.727	0.446	0.713	0.440	0.698	0.306	0.539	0.328	0.510	0.401	0.669	52.19
+ ColBERT-X + PSQ-HMM	**0.516**	**0.835**	**0.442**	**0.779**	**0.487**	**0.747**	**0.466**	**0.719**	**0.377**	**0.598**	**0.382**	**0.585**	**0.445**	**0.709**	100.24

is only significant for French, Chinese and Russian. The clear advantage of DT-SPLADE results from the lexical expansion for both documents and queries. This is consistent with the reported results for monolingual English applications of SPLADEv2, indicating that the method is fairly robust to whatever errors MT might introduce. Compared to the PSQ-HMM baseline, we see that DT-SPLADE is numerically better for every language by both measures. Moreover, these differences are statistically significant, except for MAP in French and R@100 in Spanish. From this, we can conclude that DT-SPLADE's slower indexing time is indeed rewarded by better retrieval effectiveness. Using RRF to combine DT-SPLADE and PSQ-HMM results has little effect on MAP on average across the six languages except Italian (the differences with DT-SPLADE are not significant), but it sometimes helps (and, on average over the query set, never hurts) R@100 for any language (only the gain in R@100 for Spanish and Italian are significant).

As a neural CLIR baseline that does not rely on using MT at indexing time, we experiment with ColBERT-X. With just one exception (R@100 for French), ColBERT-X consistently numerically outperforms the PSQ-HMM baseline by both MAP and R@100, although those improvements are significant (by both measures) only in Italian, Chinese, and Russian. In particular, the ColBERT MaxSim heuristic allows each query term and its matching (most similar) document term to have different representations, thus achieving greater representational fidelity than the lexical match-based PSQ-HMM approach. Using RRF to combine ColBERT-X with PSQ-HMM consistently results in further improvement over ColBERT-X alone for every language by both measures; the improvements are statistically significant by R@100 in every case and by MAP for German and Spanish. From this, we conclude that among techniques that are more efficient at indexing time than those that require running MT on every document, ColBERT-X + PSQ-HMM is a strong baseline. Moreover, the numerical effectiveness of this combination is within 4% of DT-SPLADE + PSQ-HMM on average across the six languages (slightly better for MAP, slightly worse for R@100), indicating that the substantially better indexing-time efficiency of ColBERT-X + PSQ-HMM can be achieved at little cost in effectiveness.

Having established baselines, we now answer RQ2. Looking first at single systems, we observe that, on average, across six languages, BLADE-C numerically outperforms PSQ-HMM by both MAP and R@100; the differences are only significant (by both measures) for Italian. DT-SPLADE achieves numerically higher MAP and R@100 than BLADE-C in every language, although that improvement comes at a large indexing time cost. We attribute the better performance of DT-SPLADE to its even smaller language model (covering one language rather than two), the cleaner fine-tuning from English MS MARCO without translationese, and the MT system leveraging a target language model. The picture is a bit more mixed for ColBERT-X, which numerically outperforms BLADE-C on five of the six languages by MAP (Spanish being the exception), but only three of six languages (German, Chinese, and Russian) by R@100. Differences between COLBERT-X and BLADE-C are significant (by each measure) only for Chinese and Russian,

Using RRF to combine results from efficient systems, it is possible to substantially improve over the effectiveness of any of the constituent systems. BLADE-C and PSQ-HMM are clearly complementary, with significant improvements over BLADE-C alone for five of the six languages by MAP (Italian is the exception) and for all six languages by R@100. Moreover, combining BLADE-C, ColBERT-X, and PSQ-HMM yields further improvement. That combination numerically outperforms every other system or system combination, including the combination of DT-SPLADE with PSQ-HMM, on five of the six languages by MAP (Spanish is the exception) and on three of the six languages by R@100. In almost all cases, a 3-system combination is better than the base individual systems, with significant differences for MAP and R@100. In aggregate, combining BLADE-C, ColBERT-X, and PSQ-HMM yields an average 7% improvement (across six languages) over DT-SPLADE + PSQ-HMM

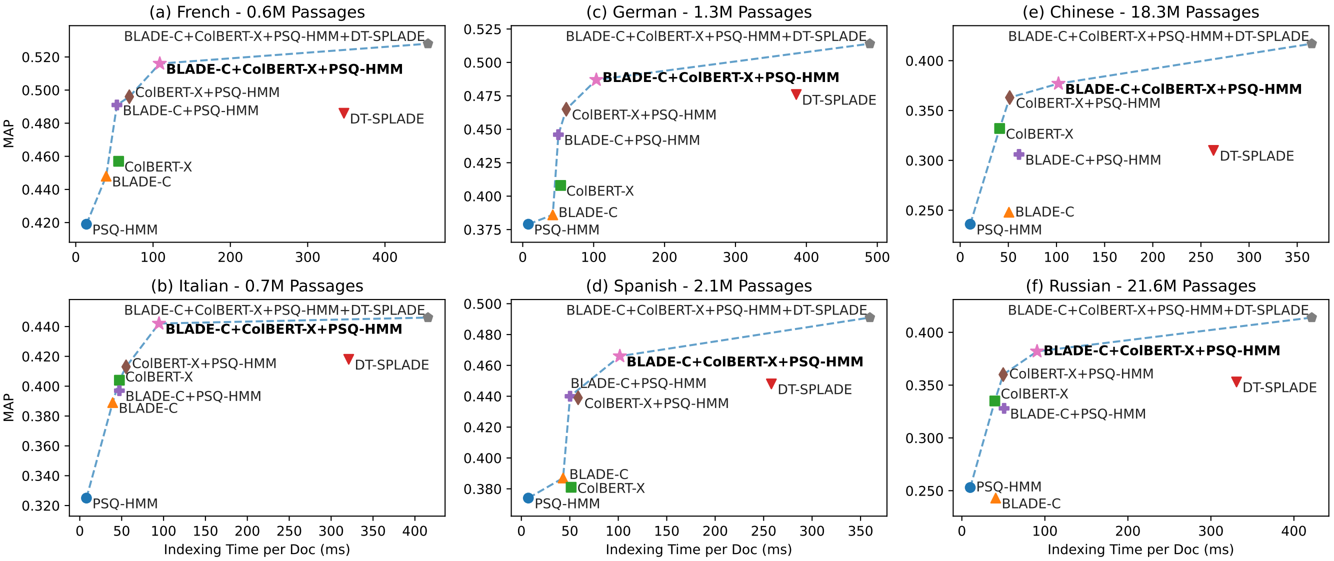

Figure 2: Indexing Efficiency vs. MAP for six collections using English queries.

by MAP with significant differences in Italian and Chinese while yielding comparable values (within 1%) for R@100 with no significant differences between the two. As we show below, these strong results can be achieved at a fraction of DT-SPLADE's indexing cost.

5.3 Optimizing for Indexing Efficiency

To this point, we have described efficiency qualitatively as being faster or slower. Here, we quantify efficiency differences, illustrating the tradeoff between efficient indexing and effective retrieval to answer RQ3. We operationalize efficiency at indexing time as the time (in milliseconds) to perform any necessary translation, run any needed model inference, and index the documents. PSQ-HMM has the fastest indexing speed since our implementation requires only the multiplication of a (translation probability) matrix and a (document term count) vector that generates a vector (of estimated English term counts). Our present PSQ-HMM implementation is CPU-based, and we could surely make it faster with a GPU-based implementation, which we leave for future work.

BLADE-C is the next fastest method, averaging slightly faster than ColBERT-X (42.6 vs. 48.1 ms). One factor that helps BLADE-C is the smaller bilingual model compared to the larger XLM-RoBERTa encoder used by ColBERT-X. In this case, averages hide some systematic variation, as the situation is reversed for Chinese and Russian, with ColBERT-X indexing at speeds comparable to BLADE. We attribute that to the difference in maximum input sequence length for the two models. The maximum sequence length for ColBERT-X is 180, as opposed to 256 for BLADE. As the collection size grows, we see an inflection point given the $O(n^2)$ time complexity of the self-attention in transformer layers. As Table 1 shows, the NeuCLIR collections for Chinese and Russian are an order of magnitude larger than a typical CLEF collection.

BLADE and ColBERT-X are much faster than DT-SPLADE (by a factor of 7.5 on average for BLADE, 6.6 for ColBERT-X). This is primarily because DT-SPLADE's inference time includes three costs: a) translating documents to the query language, b) running monolingual SPLADE on the translated texts, and c) indexing the SPLADE vectors using Anserini. The average inference time for monolingual SPLADE inference is slightly lower than that of BLADE due to SPLADE's monolingual BERT encoder having fewer parameters than BLADE's bilingual encoder. Moreover, the ratio of subwords in a bilingual model is higher than in a monolingual model, which also impacts the inference time. For the RRF system combination, indexing times are additive because each model has a different index. Our three-system combination (BLADE-C, ColBERT-X, and PSQ-HMM) is, on average, 3 times faster than DT-SPLADE.

5.4 Balancing Effectiveness and Efficiency

To investigate RQ4, we illustrate the tradeoff between effectiveness and efficiency using MAP. Use of R@100 yields similar results. Figure 2 shows this tradeoff for each language, and Figure 1 summarizes those plots using averages across all six languages. The best outcome would be in the upper left corner of those figures, where the system achieves both fast indexing and high effectiveness. In practice, the Pareto frontier is the curve that identifies the best effectiveness achieved at (or faster than) any specified indexing speed. As shown in Figure 2, PSQ-HMM and the 3-system RRF combination of BLADE-C, ColBERT-X, and PSQ+HMM are all on the Pareto frontier for each of the six languages. The 2-system combination of BLADE-C with PSQ-HMM is on the frontier for French, German and Spanish, and the 2-system combination of ColBERT-X with PSQ-HMM are on the frontier for the other three languages. The most striking point, however, is that DT-SPLADE alone is nowhere near the Pareto frontier in any language. Said another way, when indexing time matters, DT-SPLADE alone is never the best choice. A 4-system combination that includes DT-SPLADE with BLADE-C, ColBERT-X, and PSQ-HMM is also on the frontier, but the improvement in effectiveness over a 3-system combination is small relative to the additional cost in indexing time.

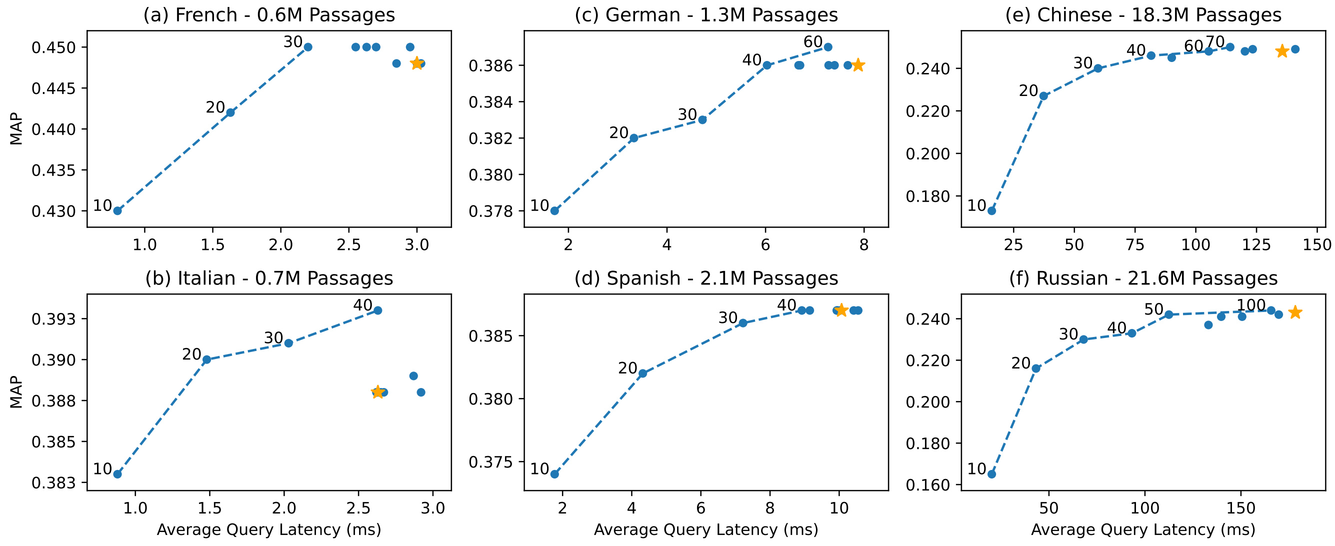

Figure 3: Average Query Latency vs MAP for BLADE-C model on the CLEF-03 and NeuCLIR collections using English queries with k ranging from 10..100 in intervals of 10. ⋆ denotes the BLADE-C model run with default k (1% of vocabulary size)

5.5 Query Latency

In a lexical expansion framework, query latency is affected by the number of terms in the expanded query. The experiments above set k to 1% of total vocabulary, which ranges from 330 to 380. Such values pose nearly no constraint on tokens output by BLADE, which usually outputs less than 100. However, the number of tokens affects query latency. Enforcing a tighter constraint on the output tokens trades between effectiveness and query latency. We vary k from 10 to 100 and plot the Pareto frontier of MAP and average query latency in Figure 3. We use PISA [45] on an AMD EPYC 7713 64-core processor with 256 GB CPU RAM to measure time to retrieve passages for a query set using BLADE-C.[15] We use PISA's multi-threaded processing with 32 threads to retrieve the top 10,000 passages for each query concurrently. With this large number of passages to retrieve, we use the MaxScore [65] dynamic pruning algorithm, as it has been shown to work well in such settings [46].

For French and Italian, with fewer than 1M passages, we see query latency between 1 and 3 ms, and stronger sparsity constraints (smaller k) provide the best effectiveness/efficiency trade-off. Larger k values (points without numbers) are sometimes far from the Pareto frontier. For German and Spanish, with between 1M and 2M passages, we have longer query latency, between 2 and 12 ms. Again, we can achieve almost the same MAP with lower values of k than the unconstrained case (1% of the vocabulary). For Chinese and Russian, with between 18M and 21M passages, we see considerably higher query latency, between 50 and 200 ms. For large collections, allowing more tokens to be output by BLADE (larger k) contributes more to effectiveness than for smaller collections. To answer RQ5, we believe k can be better tuned to collection size. Larger collections benefit from more distinguishing power between documents, so allowing more tokens benefits retrieval more.

6 CONCLUSION AND FUTURE WORK

This paper introduces BLADE, a sparse lexical expansion model for CLIR using a bilingual encoder. Our experiments using English queries to search six other languages show that task-specific intermediate pretraining is helpful, although the relative benefit of parallel or comparable text, or sentences or passages, is less clear. Experimentation with state-of-the-art models shows that BLADE yields substantially better results than SPLADE-X, and that BLADE contributes complementary evidence to system combinations with techniques that are also efficient at indexing time. These system combinations result in Pareto optimal tradeoffs between retrieval effectiveness and indexing efficiency for each of our test collections. On average, 95% of the best presently achievable MAP can be achieved with a factor of four improvements in indexing speed, and both 4-system and 3-system combinations benefit from BLADE.

These results open several lines for future research. We have shown that query latency can be reduced with little effect on effectiveness by limiting query expansion. If similar reductions were made to expansion of indexed passages, indexing time could be further accelerated. ColBERT techniques for CLIR have diverged somewhat from monolingual practice, with substantial improvements to monolingual ColBERT's efficiency (first as ColBERTv2 [60], and then as PLAID [59]) that have yet to be applied for CLIR. Given the role of ColBERT-X (which is based on the original ColBERT) in Pareto optimal system combinations, work on efficient ColBERT implementations for CLIR is clearly called for. Our experiments highlight the importance of late fusion for achieving a Pareto optimal tradeoff between indexing efficiency and retrieval effectiveness, but we note that if the term segmentations were aligned, the query-language vocabulary of Probabilistic Structured Queries would be a subset of the bilingual BLADE vocabulary. This opens the option of early fusion [13, 47], which could result in further gains in efficiency, effectiveness, or both.

[15] We do not include the time it takes to rank documents from passage rankings, as that is done in memory and is thus fast relative to retrieval.

REFERENCES

[1] Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load What You Need: Smaller Versions of Multilingual BERT. In *SustaiNLP / EMNLP*.

[2] Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA Corpus: Building Parallel Language Resources for the Educational Domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, 1856–1862. http://www.lrec-conf.org/proceedings/lrec2014/pdf/877_Paper.pdf

[3] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020. Translation Artifacts in Cross-lingual Transfer Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7674–7684. https://doi.org/10.18653/v1/2020.emnlp-main.618

[4] Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. SparTerm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768* (2020).

[5] 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. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. https://doi.org/10.48550/ARXIV.1611.09268

[6] Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Association for Computational Linguistics, Florence, Italy, 1–61. https://doi.org/10.18653/v1/W19-5301

[7] Michael Bendersky and Oren Kurland. 2008. Utilizing passage-based language models for document retrieval. In *European Conference on Information Retrieval*. Springer, 162–174.

[8] Luiz Henrique Bonifacio, Israel Campiotti, Vitor Jeronymo, Roberto Lotufo, and Rodrigo Nogueira. 2021. mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset. *arXiv preprint arXiv:2108.13897* (2021).

[9] Martin Braschler and Carol Peters. 2004. CLEF 2003 Methodology and Metrics. In *Comparative Evaluation of Multilingual Information Access Systems*. Springer Berlin Heidelberg, 7–20.

[10] Eunseong Choi, Sunkyung Lee, Minijn Choi, Hyeseon Ko, Young-In Song, and Jongwuk Lee. 2022. SpaDE: Improving sparse representations using a dual document encoder for first-stage retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 272–282.

[11] 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. arXiv:1911.02116 [cs.CL]

[12] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 758–759.

[13] Javid Dadashkarimi, Azadeh Shakery, Heshaam Faili, and Hamed Zamani. 2017. An expectation-maximization algorithm for query translation based on pseudo-relevant documents. *Information Processing & Management* 53, 2 (2017), 371–387.

[14] Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687* (2019).

[15] 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* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 985–988.

[16] 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.

[17] Tobias Domhan, Michael Denkowski, David Vilar, Xing Niu, Felix Hieber, and Kenneth Heafield. 2020. The Sockeye 2 Neural Machine Translation Toolkit at AMTA 2020. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. Association for Machine Translation in the Americas, Virtual, 110–115.

[18] Tobias Domhan, Eva Hasler, Ke Tran, Sony Trenous, Bill Byrne, and Felix Hieber. 2022. The devil is in the details: on the pitfalls of vocabulary selection in neural machine translation. *arXiv preprint arXiv:2205.06618* (2022).

[19] Andreas Eisele and Yu Chen. 2010. MultiUN: A Multilingual Corpus from United Nation Documents. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), Valletta, Malta. http://www.lrec-conf.org/proceedings/lrec2010/pdf/686_Paper.pdf

[20] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086* (2021).

[21] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2353–2359.

[22] 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.

[23] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation Degeneration Problem in Training Natural Language Generation Models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=SkEYojRqtm

[24] Luyu Gao, Xueguang Ma, Jimmy J. Lin, and Jamie Callan. 2022. Tevatron: An Efficient and Flexible Toolkit for Dense Retrieval. *arXiv preprint arxiv:2203.05765* (2022).

[25] Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. SOCKEYE 2: A Toolkit for Neural Machine Translation. In *EAMT 2020*. https://www.amazon.science/publications/sockeye-2-a-toolkit-for-neural-machine-translation

[26] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. arXiv:2010.02666 [cs.IR]

[27] Zhiqi Huang, Puxuan Yu, and James Allan. 2022. Improving Cross-lingual Information Retrieval on Low-Resource Languages via Optimal Transport Distillation. In *Proceedings of Web Search and Data Mining*. DOI: https://doi.org/10.1145/3539597.3570468.

[28] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 1–10. https://doi.org/10.3115/v1/P15-1001

[29] David Kamholz, Jonathan Pool, and Susan Colowick. 2014. PanLex: Building a Resource for Panlingual Lexical Translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, 3145–3150. http://www.lrec-conf.org/proceedings/lrec2014/pdf/1029_Paper.pdf

[30] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[31] 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.

[32] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[33] Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *Machine Translation Summit, 2005* (2005), 79–86.

[34] Carlos Lassance and Stéphane Clinchant. 2022. An efficiency study for SPLADE models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2220–2226.

[35] 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*.

[36] Yulong Li, Martin Franz, Md Arafat Sultan, Bhavani Iyer, Young-Suk Lee, and Avirup Sil. 2022. Learning Cross-Lingual IR from an English Retriever. 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, 4428–4436. https://doi.org/10.18653/v1/2022.naacl-main.329

[37] Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. Association for Computational Linguistics, New York City, USA, 104–111. https://aclanthology.org/N06-1014

[38] Jimmy Lin and Xueguang Ma. 2021. A few brief notes on DeepImpact, COIL, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv:2106.14807* (2021).

[39] Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2021. On Cross-Lingual Retrieval with Multilingual Text Encoders. *arXiv preprint arXiv:2112.11031* (2021).

[40] Joel Mackenzie, Antonio Mallia, Alistair Moffat, and Matthias Petri. 2022. Accelerating Learned Sparse Indexes Via Term Impact Decomposition. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2830–2842.

https://aclanthology.org/2022.findings-emnlp.205

[41] Joel Mackenzie, Andrew Trotman, and Jimmy Lin. 2021. Wacky weights in learned sparse representations and the revenge of score-at-a-time query evaluation. *arXiv preprint arXiv:2110.11540* (2021).

[42] Joel Mackenzie, Andrew Trotman, and Jimmy Lin. 2022. Efficient Document-at-a-Time and Score-at-a-Time Query Evaluation for Learned Sparse Representations. *ACM Transactions on Information Systems* (2022).

[43] Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1723–1727.

[44] Antonio Mallia, Joel Mackenzie, Torsten Suel, and Nicola Tonellotto. 2022. Faster learned sparse retrieval with guided traversal. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1901–1905.

[45] Antonio Mallia, Michal Siedlaczek, Joel Mackenzie, and Torsten Suel. 2019. PISA: Performant Indexes and Search for Academia. In *Proceedings of the Open-Source IR Replicability Challenge co-located with 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, OSIRRC@SIGIR 2019, Paris, France, July 25, 2019.* 50–56. http://ceur-ws.org/Vol-2409/docker08.pdf

[46] Antonio Mallia, Michał Siedlaczek, and Torsten Suel. 2019. An experimental study of index compression and DAAT query processing methods. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41.* Springer, 353–368.

[47] Suraj Nair, Petra Galuscakova, and Douglas W Oard. 2020. Combining contextualized and non-contextualized query translations to improve CLIR. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1581–1584.

[48] 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 *European Conference on Information Retrieval.* Springer, 382–396.

[49] Suraj Nair, Eugene Yang, Dawn Lawrie, James Mayfield, and Douglas W Oard. 2022. Learning a Sparse Representation Model for Neural CLIR. In *Proceedings of the Third International Conference on Design of Experimental Search & Information REtrieval Systems, San Jose, CA.*

[50] Rodrigo Nogueira and Jimmy Lin. 2019. *From doc2query to docTTTTTquery.* Technical Report. University of Waterloo. https://cs.uwaterloo.ca/~jimmylin/publications/Nogueira_Lin_2019_docTTTTTquery-v2.pdf

[51] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[52] Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics* (2003).

[53] Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics* (2016).

[54] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics.* 311–318.

[55] Biswajit Paria, Chih-Kuan Yeh, Ian E.H. Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. Minimizing FLOPs to Learn Efficient Sparse Representations. In *International Conference on Learning Representations.* https://openreview.net/forum?id=SygpC6Ntvr

[56] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining.* 813–823.

[57] Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics. https://arxiv.org/abs/2004.09813

[58] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *NIST Special Publication Sp* 109 (1995), 109.

[59] 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.

[60] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488* (2021).

[61] Xing Shi and Kevin Knight. 2017. Speeding Up Neural Machine Translation Decoding by Shrinking Run-time Vocabulary. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* Association for Computational Linguistics, Vancouver, Canada, 574–579. https://doi.org/10.18653/v1/P17-2091

[62] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316* (2021).

[63] Shuo Sun and Kevin Duh. 2020. CLIRMatrix: A massively large collection of bilingual and multilingual datasets for Cross-Lingual Information Retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* 4160–4170.

[64] Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS.. In *Lrec*, Vol. 2012. Citeseer, 2214–2218.

[65] Howard Turtle and James Flood. 1995. Query evaluation: strategies and optimizations. *Information Processing & Management* 31, 6 (1995), 831–850.

[66] Vered Volansky, Noam Ordan, and Shuly Wintner. 2015. On the features of translationese. *Digital Scholarship in the Humanities* 30, 1 (2015), 98–118.

[67] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

[68] 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.

[69] Eugene Yang, Suraj Nair, Ramraj Chandradevan, Rebecca Iglesias-Flores, and Douglas W. Oard. 2022. C3: Continued Pretraining with Contrastive Weak Supervision for Cross Language Ad-Hoc Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22).* Association for Computing Machinery, New York, NY, USA, 2507–2512. https://doi.org/10.1145/3477495.3531886

[70] Jheng-Hong Yang, Xueguang Ma, and Jimmy Lin. 2021. Sparsifying Sparse Representations for Passage Retrieval by Top-k Masking. *arXiv preprint arXiv:2112.09628* (2021).

[71] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17).* Association for Computing Machinery, New York, NY, USA, 1253–1256.

[72] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 497–506.

[73] Shengyao Zhuang and Guido Zuccon. 2021. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *arXiv preprint arXiv:2108.08513* (2021).

[74] Shengyao Zhuang and Guido Zuccon. 2021. TILDE: Term independent likelihood moDEl for passage re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1483–1492.

[75] Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16).* 3530–3534.