Efficiency-Effectiveness Tradeoff of Probabilistic Structured Queries for Cross-Language Information Retrieval

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

Suraj Nair*
Amazon
Seattle, Washington, USA
srnair@cs.umd.edu

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

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

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

Kevin Duh
HLTCOE, Johns Hopkins University
Baltimore, Maryland, USA
kevinduh@cs.jhu.edu

ABSTRACT

Probabilistic Structured Queries (PSQ) is a cross-language information retrieval (CLIR) method that uses translation probabilities statistically derived from aligned corpora. PSQ is a strong baseline for efficient CLIR using sparse indexing. It is, therefore, useful as the first stage in a cascaded neural CLIR system whose second stage is more effective but too inefficient to be used on its own to search a large text collection. In this reproducibility study, we revisit PSQ by introducing an efficient Python implementation. Unconstrained use of all translation probabilities that can be estimated from aligned parallel text would in the limit assign a weight to every vocabulary term, precluding use of an inverted index to serve queries efficiently. Thus, PSQ's effectiveness and efficiency both depend on how translation probabilities are pruned. This paper presents experiments over a range of modern CLIR test collections to demonstrate that achieving Pareto optimal PSQ effectiveness-efficiency tradeoffs benefits from multi-criteria pruning, which has not been fully explored in prior work. Our Python PSQ implementation is available on GitHub[1] and unpruned translation tables are available on Huggingface Models.[2]

CCS CONCEPTS

• **Information systems** → **Multilingual and cross-lingual retrieval**; **Probabilistic retrieval models**; Retrieval effectiveness; *Retrieval efficiency*.

KEYWORDS

Cross-Language Information Retrieval, Statistical Machine Translation, Probabalistic Structured Queries

1 INTRODUCTION

While recent developments in IR have focused on neural approaches with dense vectors produced by pretrained language models, such as DPR [21] and ColBERT [22], studies have demonstrated that incorporating lexical retrieval models into the neural pipeline is still beneficial [26, 30, 41, 53]. Specifically for Cross-Language Information Retrieval (CLIR), Nair et al. [41] showed that combining rankings from sparse and dense retrieval models yields a better Pareto-optimal efficiency-effectiveness tradeoff. Among sparse models, Probabilistic Structured Query (PSQ) [10, 25, 52, 56] is extremely efficient compared to neural retrieval models. Therefore, combining any model with PSQ can potentially receive an effectiveness boost without increased latency, assuming a suitable parallel implementation.

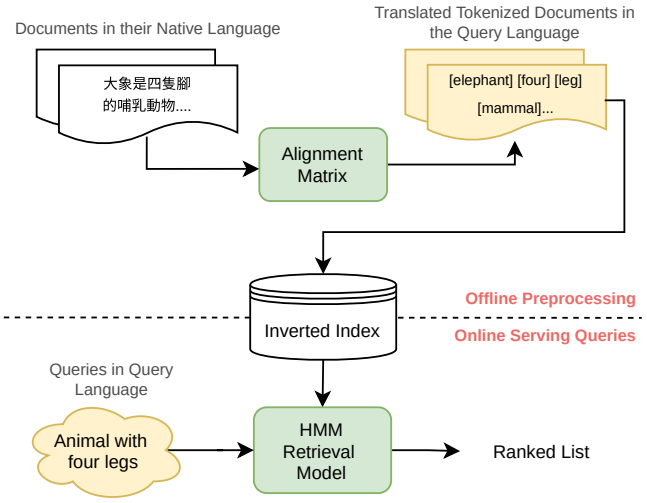


Figure 1: Indexing-time PSQ Retrieval Pipeline

PSQ is a statistical term translation approach designed for use with sparse retrieval models such as BM25 or Hidden Markov Model (HMM) to perform CLIR. It translates text from one language to another on a term-by-term basis, which can be performed in conjunction with other retrieval text preprocessing, such as stopword removal or stemming, to maximize retrieval effectiveness.

Because each term is treated independently, the translation mappings in PSQ can be computed either at query time or at indexing time. Most prior work on PSQ has focused on query-time implementation. This limits effort at indexing time, but at the cost of greater effort at query time, where a few query words could expand to

*Work done prior to Suraj joined Amazon.
[1]https://github.com/hltcoe/PSQ
[2]https://huggingface.co/hltcoe/psq_translation_tables

hundreds of tokens [10, 25, 52]. While Xu and Weischedel [56] proposed an indexing-time implementation of PSQ that aims for better query-time efficiency by offloading most computation to indexing time, their implementation did not focus on pruning the very large number of possible translations. Thus it was not compatible with an inverted index, limiting its practical use in large-scale applications. To further understand the tradeoff between retrieval effectiveness and efficiency, this work implements an efficient indexing-time PSQ. Figure 1 summarizes our process.

In statistical alignment models that estimate alignment probabilities based on token co-occurrence, each term has a long tail of alternate translations. While including more translations improves retrieval effectiveness, the resulting inverted index becomes dramatically larger, degrading efficiency.

Wang and Oard [52] experimented with three pruning strategies: Probability Mass Function (PMF) (a minimum threshold on the translation probability); Cumulative Distribution Function (CDF) (a maximum threshold on the cumulative probability included in the alignment for a given source token); and Top-k threshold (a maximum threshold on the number of alternative translations). The original paper studied the impact of applying each pruning strategy individually. Combinations of the three have been applied in practice to trim the alignment table [41], but a broad range of possible combinations has yet to be studied.

This reproducibility study focuses on revisiting PSQ and analyzing the efficiency and effectiveness tradeoff in a Pareto framework with multiple factors. We primarily focus on reproducing the findings from Wang and Oard [52] because of its potential impact in modern IR algorithms, such as SPLADE [15] and BLADE [41], that also exhibit an efficiency-effectiveness tradeoff in pruning the index. We revisit these pruning techniques with a modern technology stack, including more and cleaner parallel text, indexing-time processing, a public implementation that we will share, and larger evaluation collections. Specifically, we reimplement the indexing-time PSQ-HMM retrieval model introduced by Xu and Weischedel [56]. To further understand the tradeoff, we conduct additional analysis on applying a combination of pruning techniques with Pareto optimality.

The contributions of this work are: (1) an efficient indexing-time PSQ implementation in Python; (2) experimental results with modern, large CLIR evaluation collections and with more parallel text; and (3) a multi-factor analysis of the efficiency-effectiveness tradeoff for index pruning.

2 REVIEW OF PSQ DEVELOPMENT

Early work on CLIR for free text relied on comparable corpora [27, 49], dictionary-based techniques [3], or the use of rule-based Machine Translation (MT) [42]. However, CLIR researchers soon began to use parallel corpora, either directly [34] or through statistical MT [17, 54]. The basic approach to the direct application of parallel corpora was to first estimate translation probabilities from term alignments, and then use those probabilities to weight the contribution of each translation in the computation of the document scores on which ranking was based.

Early work with this technique used a smoothed unigram language model as the ranking function [56], although Darwish and Oard [10] later showed that the same approach could be used with other ranking functions. Drawing insight from Structured Queries [45], a dictionary-based CLIR technique, Darwish and Oard [10] gave the name Probabilistic Structured Queries (PSQ) to the general approach. One way to conceptualize PSQ is to think of the translation probability matrix as a lens through which to view a document-language term frequency vector. The result of this matrix-vector product is a vector of expected values for query-language term frequencies. That vector can then be used with any ranking algorithm that uses term frequencies, such as BM25 or unigram language models. This same idea has also been used for annotation projection, in which labels such as syntactic structures [19], semantic roles [2], or sentiment [47] are transferred from corpora in one language to another.

This broad idea still leaves a large design space. Darwish and Oard [10] experimented with estimating Inverse Document Frequency (IDF) in a similar way, since BM25 requires IDF estimates; Montazeralghaem et al. [37] have subsequently shown that estimation can be improved. However, Nair et al. [39] found that estimating query-language IDF from a side collection such as the New York Times[3] works well; that is the approach we use in this paper. Axiomatic analysis has recently suggested that the use of IDF, as in BM25, may also have theoretical advantages over unigram language models that seek to achieve a similar effect using inverse collection frequency [46].

Early PSQ experiments (including its predecessors specific to unigram language models) relied on unidirectional GIZA++[43] alignments, and subsequent techniques for improving alignments [11, 13, 16] have been found to further improve PSQ [52].

Probabilities estimated from parallel text are typically pruned, both for practical reasons (dense translation matrices would be very large) and because small probabilities have small effects. Alignment systems typically suppress probability estimates below some value (10^{-6} in our work); we refer to this as filtering on the *Probability Mass Function (PMF)*. Wang and Oard [52] also experimented with retaining only enough alternate translations to reach some total probability; we call this filtering on the *Cumulative Distribution Function (CDF)*. A third alternative is to retain only some number k of the highest probability translations; we call this *top-k* filtering. Wang and Oard [52] found that each approach yielded similar results by their Mean Average Precision (MAP) measure, which rose as more translations were retained, and then, beyond some point, fell. Importantly, however, they renormalized after filtering; whether to do so is another design decision.

The popular focus on a query-time implementation of PSQ is often justified by claiming that it is more efficient than an indexing time implementation. While that is certainly true for experimentation with typical test collections that have many documents and few queries, there are also applications (e.g., Web search) in which the query volume can exceed even the volume of the indexed content. Moreover, indexing-time implementations offer the potential for some elasticity (e.g., servers can be allocated at lower-workload times); queries, on the other hand, must be responded to in real-time as they arrive. In this paper we therefore focus on indexing-time implementation of PSQ.

[3] https://catalog.ldc.upenn.edu/LDC2008T19

Neural CLIR methods have recently been shown to yield better rankings than PSQ [40]. Due to the computational cost of neural methods, they are frequently used in cascades in which sparse methods like PSQ generate initial results for neural reranking [41] and in system combinations, since PSQ and neural CLIR have some complementary strengths. PSQ thus lives on, even in a neural world.

3 GENERATING TRANSLATION PROBABILITIES

Word alignments are needed in many applications, most notably in statistical Machine Translation (MT). The seminal work of Brown et al. [7] introduced IBM Models 1-5 to estimate word alignments from parallel corpora. Using Expectation-Maximization, these generative models infer latent word alignments from document-aligned pairs of texts in source and target languages. IBM Model 1 computes the lexical translation table by aligning words independently. Model 2 allows the alignment to be dependent on the respective sentence lengths. Model 3 introduces a fertility model, which indicates how many target words a source word may translate to. Model 4 accounts for reordering, modeling varying word orders across languages. Like Model 4, Hidden Markov Model (HMM) alignment [51] captures first-order dependencies, where alignments depend on previously aligned tokens.[4] GIZA++ [43] is a toolkit that includes implementations of IBM Models 1-5 and the HMM model.

Using word alignments, we can generate lexical translation tables by aggregating the co-occurrence counts of aligned word pairs. GIZA++ performs alignments in two directions, source-to-target, and target-to-source. In the source-to-target direction, GIZA++ aligns words in the source language to words in the target language to estimate $P(target|source)$ and normalizes their sum to one. Similarly, in the target-to-source direction, GIZA++ aligns words in the source language to the words in the target language to estimate $P(source|target)$ and normalizes their sum to one. Wang and Oard [52] explored both possible normalization directions: the probability of a query-language term given a document-language term, or the probability of a document-language term given a query-language term. They found the best results from the first of those, and that is therefore the direction in which we normalize for this paper. Furthermore, we can concatenate the outputs of different word aligners to estimate a single lexical translation table, following Zbib et al. [57] and Nair et al. [38].

Using such combined word alignments, the co-occurrence counts of word pairs that appear in multiple aligners increase, subsequently increasing the likelihood of these word pairs being translations [55]. From here on, we refer to the lexical translation table produced by the word aligners as an alignment matrix.

4 PSQ-HMM

In this section, we summarize the overall PSQ-HMM retrieval model pipeline, which is an aggregation of the proposals by Xu and Weischedel [56], Darwish and Oard [10], Croft et al. [9], and Wang and Oard [52].

[4]We note that HMM alignment differs from HMM ranking. In HMM ranking (as used, for example, in PSQ-HMM) what has been called the query likelihood ranking model is implemented as an HMM.

Based on Xu and Weischedel [56] and Darwish and Oard [10], we translate the documents $D_{\mathcal{S}}$ in language $\mathcal{S}$ into query language $\mathcal{T}$ with an alignment table trained on parallel corpus $B_{\mathcal{S},\mathcal{T}}$. The probability that query language token $w_{\mathcal{T}}$ represents content in document $D_{\mathcal{S}}$ is expressed as

$$P(w_{\mathcal{T}}|D_{\mathcal{S}}) = \sum_{w_{\mathcal{S}} \in D_{\mathcal{S}}} P_B(w_{\mathcal{T}}|w_{\mathcal{S}}) P(w_{\mathcal{S}}|D_{\mathcal{S}})$$

where $P(w_{\mathcal{S}}|D_{\mathcal{S}})$ is the probability of token $w_{\mathcal{S}}$ in document $D_{\mathcal{S}}$, estimated from its term frequency; and $P_B(w_{\mathcal{T}}|w_{\mathcal{S}})$ is the alignment probability between tokens $w_{\mathcal{T}}$ and $w_{\mathcal{S}}$, estimated on B.

Theoretically, every token in the query language has a non-zero probability of aligning with every token in the document language, while only a few such alignments are meaningful. Here, we estimate such probabilities through a parallel corpus $B_{\mathcal{S},\mathcal{T}}$ (denoted as P_B), which limits the possible alignments of a given token $w_{\mathcal{S}}$ to the ones that co-occur in $B_{\mathcal{S},\mathcal{T}}$. For simplicity, we denote the estimated probability of a token $w_{\mathcal{T}}$ being in document $D_{\mathcal{S}}$ as $P(w_{\mathcal{T}}|D_{\mathcal{S}})$ without the B subscript.

4.1 HMM Retrieval Model

Assuming a uniform prior on document relevance, the probability that document $D_{\mathcal{S}}$ is relevant (denoted "is R") to query $Q_{\mathcal{T}}$ can be expressed as

$$\begin{aligned} P(D_{\mathcal{S}} \text{ is R}|Q_{\mathcal{T}}) &= \frac{P(Q_{\mathcal{T}}|D_{\mathcal{S}} \text{ is R}) P(D_{\mathcal{S}} \text{ is R})}{P(Q_{\mathcal{T}})} \\ &\propto P(Q_{\mathcal{T}}|D_{\mathcal{S}} \text{ is R}) \end{aligned}$$

By modeling the relevance with an HMM [35] model, the overall relevance is the product of the probability of the document being relevant to each query term. Therefore, the scoring function of a query-document pair integrated with PSQ is the probability $P(Q_{\mathcal{T}}|D_{\mathcal{S}} \text{ is R})$ [56], which can be expanded as

$$\begin{aligned} &Score(Q_{\mathcal{T}}, D_{\mathcal{S}}) = P(Q_{\mathcal{T}}|D_{\mathcal{S}} \text{ is R}) \\ &= \prod_{w_{\mathcal{T}} \in Q_{\mathcal{T}}} \Big[\alpha P(w_{\mathcal{T}}|G_{\mathcal{T}}) + (1-\alpha) P(w_{\mathcal{T}}|D_{\mathcal{S}})\Big] \\ &\propto \sum_{w_{\mathcal{T}} \in Q_{\mathcal{T}}} \log\Big[\alpha P(w_{\mathcal{T}}|G_{\mathcal{T}}) + (1-\alpha) P(w_{\mathcal{T}}|D_{\mathcal{S}})\Big] \end{aligned} \tag{1}$$

where $P(w_{\mathcal{T}}|G_{\mathcal{T}})$ is the probability of $w_{\mathcal{T}}$ in a unigram language model and $\alpha \in [0, 1]$ is a Jelinek-Mercer smoothing hyperparameter to balance the two probabilities. $P(w_{\mathcal{T}}|G_{\mathcal{T}})$ can be estimated using an external corpus, as discussed in the next section. As Hiemstra [18] has noted, what we call here HMM Retrieval is mathematically equivalent to a smoothed unigram language model, which has also been called the query likelihood model.

However, direct implementation of the smoothing in Equation 1 would preclude using inverted index structures for efficient query processing because it gives a weight to every term in the vocabulary for every document. In sparse retrieval models, such as BM25, the score of each document is the sum of its score for each query term. In these models, the term score will be zero if the term does not appear in the document, and those many zeros are what make inverted indexing efficient. To achieve that sparsity with an HMM retrieval model, we modify the similarity function by subtracting the baseline

Table 1: Collection Statistics. We use the same alignment matrix for NTCIR-8 and NeuCLIR Chinese.

	CLEF 2003				NTCIR-8	NeuCLIR 2022		
	French	Italian	German	Spanish	Chinese	Chinese	Persian	Russian
# of Parallel Sentences	17.6M	3.6M	4.4M	15.7M	12.1M	12.1M	20.8M	14.5M
# of Docs	130K	158K	295K	454K	309K	3,179K	2,232K	4,628K
# of Topics	60	60	60	60	100	49	46	45
# of Topics w/ Rel Docs	52	51	56	57	100	49	46	45

score of a document with no overlapping query terms [9, Chapter 7.3.1]. This does not change the ordering of the documents but restores the sparsity. Specifically, we define the scoring function as

$$Score(Q_{\mathcal{T}}, D_{\mathcal{S}}) = \log P(Q_{\mathcal{T}}|D_{\mathcal{S}} \text{ is R}) - \sum_{w_{\mathcal{T}} \in Q_{\mathcal{T}}} \log(\alpha P(w_{\mathcal{T}}|G_{\mathcal{T}})) \quad (2)$$

Reorganizing Equation 2, the score of query term $w_{\mathcal{T}}$ for document $D_{\mathcal{S}}$ becomes zero if the term does not appear in the document. Therefore, we consider only overlapping terms between $Q_{\mathcal{T}}$ and the probabilistic translation of $D_{\mathcal{S}}$. The scoring function becomes the sum of each term weight, which we define as $v^{D_{\mathcal{S}}}_{w_{\mathcal{T}}}$ for convenience,

$$\begin{aligned} Score(Q_{\mathcal{T}}, D_{\mathcal{S}}) &= \sum_{w_{\mathcal{T}} \in I_{\mathcal{T}}} v^{D_{\mathcal{S}}}_{w_{\mathcal{T}}} \\ &= \sum_{w_{\mathcal{T}} \in I_{\mathcal{T}}} \log \left[\frac{\alpha P(w_{\mathcal{T}}|G_{\mathcal{T}}) + (1-\alpha) P(w_{\mathcal{T}}|D_{\mathcal{S}})}{\alpha P(w_{\mathcal{T}}|G_{\mathcal{T}})} \right] \\ &= \sum_{w_{\mathcal{T}} \in I_{\mathcal{T}}} \log \left[\frac{(1-\alpha) P(w_{\mathcal{T}}|D_{\mathcal{S}})}{\alpha P(w_{\mathcal{T}}|G_{\mathcal{T}})} + 1 \right] \end{aligned} \quad (3)$$

where $I_{\mathcal{T}} = Q_{\mathcal{T}} \cap \{w_{\mathcal{T}} | P(w_{\mathcal{T}}|D_{\mathcal{S}}) > 0\}$. With Equation 3, the collection can be indexed in an inverted index where the keys are query-language tokens. At query time, we gather documents with at least one query term by traversing postings lists, which is efficient with an inverted index.

We implement the sparse retrieval with SciPy CSC sparse matrices[5] where columns represent the tokens in the query language and rows represent the documents. This is mathematically identical to using an inverted index, since both approaches use the token as the lookup key for a list of documents that contain the token. It would be possible to further integrate our implementation with other sparse retrieval systems such as Lucene[6] and Pisa [33], which we leave for future work.

At indexing time, the alignment matrix is transformed from dictionary form to a sparse matrix with pruning in preparation for fast indexing. Each document is tokenized and vectorized into a sparse vector and multiplied with the alignment matrix to get a sparse vector in the query language. The resulting vector is modified based on Equation 3 to incorporate the query language unigram language model in its weights. Vectors are stored in chunks and then merged into a single inverted index, which is materialized by a CSC sparse matrix.

[5]https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csc_matrix.html
[6]https://lucene.apache.org/core/

At query time, the incoming query is preprocessed into a sparse vector and multiplied by the inverted index sparse matrix. Such multiplication is efficient and analogous to retrieving with an inverted index, except for various posting list skipping techniques.

4.2 Retrieval Efficiency

At indexing time, each document is tokenized, and its term frequency vector is then translated with the alignment matrix as a dot product between the sparse vector in the document language $\mathcal{S}$ and the sparse alignment matrix $\mathcal{S} \rightarrow \mathcal{T}$. This yields a sparse vector in the query language $\mathcal{T}$. For each translated term $w_{\mathcal{T}}$, we calculate its weight $v^{D_{\mathcal{S}}}_{w_{\mathcal{T}}}$ based on Equation 3. Documents are then indexed using the translated terms in the query language $\mathcal{T}$ as keys, and each postings list identifying documents represented using the translated term, ordered by their weights $v^{D_{\mathcal{S}}}_{w_{\mathcal{T}}}$.

Latency in serving the queries is strongly correlated with the length of the postings list for each query term under the same parallelism capability [32]. By performing PSQ at indexing time, the number of alternate translations for each document term directly affects the size of the translated documents, and, thus, the length of each postings list. However, limiting the number of alternate translations also reduces retrieval effectiveness [52]. This dilemma implies a clear tradeoff between retrieval effectiveness and efficiency.

5 PRIOR FINDINGS ON PRUNING TRANSLATION ALTERNATIVES

Wang and Oard [52] explored three strategies for pruning the number of translation alternatives: minimum translation probability (*PMF Threshold*); maximum cumulative probability (*CDF Threshold*); and an integer threshold on the number of possible translations (*Top-k* filtering).

In their work, they evaluated the three strategies using query-side PSQ on CLIR tasks of French and Chinese documents searched with English queries. Wang and Oard [52] observed that applying CDF and PMF thresholds provides a broad range of efficiency (measured in average number of translated query tokens) and effectiveness (measured in percentage of monolingual MAP); in their opinion, these were ideal choices for the efficiency-effectiveness tradeoff. Top-k filtering, in contrast, requires more translation alternatives to perform reasonably well, although it demonstrates a higher effectiveness peak.

However, we argue that a broader range of efficiency and effectiveness values does not mean a better tradeoff. The quality of the efficiency-effectiveness tradeoff for a given pruning strategy

should be measured by Pareto-optimality [41]. One strategy with an higher Pareto frontier than another indicates being more effective with the same efficiency constraint. Therefore, we revisit the claim by comparing the three pruning strategies with a Pareto analysis framework.

Additionally, there are a number of limitations in Wang and Oard [52]:

(1) they only evaluated with a query-side PSQ implementation;
(2) they did not consider combinations of pruning techniques;
(3) they did not have access to as large a quantity of parallel text; and
(4) they did not have access to modern large CLIR test collections.

While influential work at the time, their conclusions may not hold with a more efficient implementation and more rigorous evaluation. In the next section, we describe our modern experiment setup for PSQ.

6 EXPERIMENT SETUP

This section describes the design of our experiments. We first describe the text alignment. Then we enumerate the hyperparameters used. Next we discuss baselines, and finally we introduce our evaluation measures.

6.1 Parallel Text Alignment

To evaluate PSQ with modern resources, we train the alignment matrix with modern collections of parallel text, which have a far larger scale, and a far greater diversity, than what was available to Wang and Oard [52]. We use multiple sources from OPUS [50], including EuroParl [23], GlobalVoices,[7] MultiUN [14], NewsCommentary,[8] QED [1], TED [48], and WMT-News [4]. We also use bilingual Panlex dictionaries [20] as an extra source of parallel text. The number of parallel sentences for each language is summarized in the top row of Table 1. To generate word alignments, we use three aligners: bidirectional GIZA++ [43], BerkeleyAligner [31], and Eflomal [44]. These individual word alignments are concatenated to create a unified alignment file. Using this concatenated file, we proceed to create the unidirectional alignment matrix (i.e., the probability of a query-language (in our experiments, English) term given a document-language (non-English) term through maximum likelihood estimation.[9] For bidirectional GIZA++, we run 5 iterations of Model 1 and HMM each, followed by 3 iterations of Model 3 and Model 4 each, using the Moses toolkit. The word alignments are estimated using the grow-diag-final-and heuristic from the bidirectional word alignments [24]. For BerkeleyAligner, we obtain the word alignments by running 5 iterations of Model 1 followed by 5 iterations of HMM, both trained jointly in the bidirectional setting. For Eflomal, we use the default settings and estimate word alignments using the grow-diag-final-and heuristic, similar to GIZA++.[10] We use the Google one-billon word corpus [8] to estimate the query-language unigram language model.

6.2 Evaluation Collections

For retrieval evaluation, we use French, Italian, German, and Spanish collections from CLEF 2003 [6], Simplified Chinese from the NTCIR-8 ACLIA Task [36], and Chinese, Persian, and Russian from TREC NeuCLIR 2022 [28]. The NeuCLIR collections are much larger than those from CLEF and NTCIR. Since our Chinese parallel text uses simplified characters, we mapped traditional characters in the NeuCLIR Chinese collection to simplified characters to match the alignment matrix.[11] Collection statistics are summarized in Table 1. Our experiments use the English title field of the topics (short, keyword descriptions of the information need) as queries.

For both parallel text and the information retrieval test collections, we tokenize the text with Moses Tokenizer[12] and remove punctuation, diacritics, stopwords (provided by NLTK [5]), and casing to normalize the text. For Chinese, we segment sentences into words that may comprise more than one character using Jieba.[13] Matching text preprocessing across the two steps ensures consistency when applying alignment matrices to retrieval documents, and in our experience failure to do so is one of the most common mistakes made by people who are new to PSQ.

6.3 Hyperparameters

For our main results, we create the inverted index without unnecessary pruning. We simply impose an initial PMF threshold of 10^{-6} to ensure the resulting inverted index will fit in memory. To investigate the tradeoff between retrieval effectiveness and index size, we experiment with three hyperparameters for trimming the alignment matrix: minimum PMF, maximum CDF, and the k for top-k filtering. All three approaches limit index size, but they operate differently. The experiment involves all combinations of six minimum PMF thresholds (0, 0.05, 0.01, 0.001, 0.0001, and 0.00001), eight values for top-k (2, 4, 8, 16, 32, 64, 128, and unrestricted) and ten CDF thresholds (0.800, 0.900, 0.950, 0.960, 0.965, 0.970, 0.975, 0.980, 0.990, and 1.000), resulting in 480 models per collection. We aim to cover values that one might consider using in a practical application, and also values that might catastrophically damage the retrieval results.

We also experimented with other pruning methods such as minimum document term weight $v^{D_S}_{W_T}$ and maximum number of translated tokens in a document. We omit those results from this paper as they generally offer a worse tradeoff between index size and retrieval effectiveness.

6.4 Baselines

We evaluate PSQ-HMM against BM25 ($k_1 = 0.9, b = 0.4$) with the use of one-best machine translation of either the query or the document, which are both strong CLIR baselines [28, 40]. In prior work [anonymized], we had tried both HMM and BM25 retrieval models with one-best machine translation results, finding that the BM25 results were numerically slightly better. We therefore report results with BM25 as being representative of what can be achieved using one-best machine translation. For query translation (QT), we

[7] https://casmacat.eu/corpus/global-voices.html
[8] https://data.statmt.org/news-commentary/v16/
[9] http://www2.statmt.org/moses/?n=FactoredTraining.GetLexicalTranslationTable
[10] https://github.com/robertostling/eflomal/tree/7b97f19187c8b1bc1f21aefd77fc1b87575d1c00
[11] https://github.com/NeuCLIR/download-collection/blob/ad53164a90dbe0b1e6a9fc7e960e643f4cab7f4a/convert_chinese_char.py
[12] https://github.com/luismsgomes/mosestokenizer
[13] https://github.com/fxsjy/jieba

Table 2: Effectiveness, title queries, no index pruning. ‡and †indicate that PSQ-HMM is significantly different from QT-BM25 or DT-BM25, respectively, at $p < 0.05$ by a two-tailed paired t-test. Individual collection results have Bonferroni correction for 8 tests. Significance tests for the Average column are paired over the full set of 456 topics.

		CLEF 2003				NTCIR-8	NeuCLIR 2022			Average
		French	Italian	German	Spanish	Chinese	Chinese	Persian	Russian	
R@100	QT-BM25	0.694	0.600	0.438	0.559	0.311	0.484	0.449	0.428	0.477
	DT-BM25	0.735	0.673	0.629	0.567	0.441	0.479	0.475	0.436	0.546
	PSQ-HMM	0.808	0.657	‡0.677	0.620	‡†0.523	0.456	0.488	0.467	‡†0.585
MAP	QT-BM25	0.403	0.334	0.274	0.327	0.165	0.242	0.205	0.263	0.267
	DT-BM25	0.381	0.348	0.376	0.336	0.262	0.264	0.209	0.245	0.302
	PSQ-HMM	0.470	0.336	0.420	0.378	‡0.318	0.238	0.212	0.257	‡†0.332

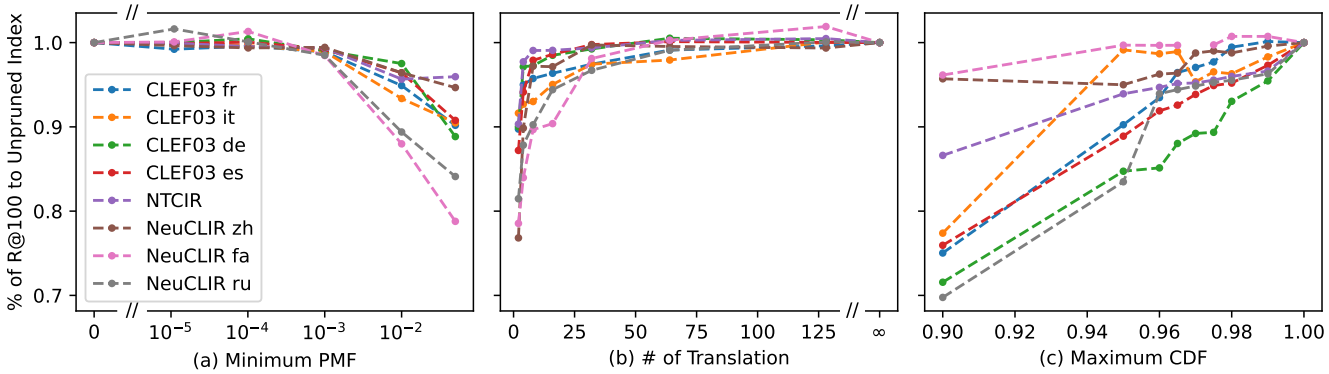


Figure 2: Relative R@100 score to the unpruned index of each collection. The x-axis of the PMF threshold graph is in log scale, and the unpruned index (0) is marked at the far left.

translated the queries using Google Translate. For document translation (DT), we use public sets of translated documents provided in prior work [29, 40], which were generated by AWS Sockeye V2 [12] trained on general domain parallel text.

6.5 Evaluation Measures

We measure retrieval effectiveness using mean average precision (MAP) and recall at 100 (R@100), which have been commonly reported in prior work on these test collections [40, 41, 46, 52]. We use R@100 to characterize retrieval effectiveness when used as the first stage of a neural reranking pipeline, and MAP to characterize standalone use of a technique as a single-stage ranking model. Following prior work, we only consider topics with at least one known relevant document when reporting effectiveness measures, since neither of our measures can distinguish systems on topics for which no relevant documents are known to exist. For retrieval efficiency, we could use wall clock time to measure query latency, but that could be affected by concurrent processes on the same machine. To avoid that source of noise, we instead use index size as a proxy for query latency. Query latency is linear in the length of the postings lists for the query terms, and the index size is dominated by the postings lists (the vocabulary is small enough to fit in memory). Thus the total index size is a good proxy for query latency [32].

7 RESULTS AND ANALYSIS

In this section, we first compare the effectiveness of PSQ-HMM on our test collections with baselines using modern neural machine translation models, which situates the performance of PSQ with respect to modern sparse CLIR models. We then revisit the claim of Wang and Oard [52] by conducting a similar analysis, applying each pruning technique individually. Finally, we present experiments applying a combination of techniques to further analyze the efficiency-effectiveness tradeoff with Pareto optimality.

7.1 Effectiveness Compared to Baselines with Modern Machine Translation

As summarized in Table 2, PSQ-HMM without index pruning numerically outperforms both the use of Google for query translation (QT-BM25) and the use of Sockeye for document translation (DT-BM25) in both MAP and R@100. Focusing first on R@100, without index pruning PSQ outperforms DT and QT on six of the eight collections (the exceptions are Italian and NeuCLIR Chinese), although the only statistically significant improvements are QT-BM25 for German and NTCIR Chinese and DT-BM25 for NTCIR Chinese. As Table 1 shows, the number of topics in each individual collection is limited, but across the eight collections, we have 456 topics. That larger number is sufficient to detect statistically significant

improvements over both QT-BM25 and DT-BM25 in the microaveraged R@100 across the eight collections, weighting each collection in proportion to the number of topics that it contributes to the total.

Looking now at the two languages for which we saw numerical decreases in R@100, we focus first on NeuCLIR Chinese. We transformed all traditional Chinese characters in that collection to simplified characters, so character mismatch between the collection and the alignment matrix cannot explain the problem. However, there are also vocabulary differences to be considered. The vocabulary in the document collection, which includes a substantial number of documents from outside China, differs markedly from the vocabulary in the alignment matrix that was principally trained on parallel text from China. Such a vocabulary mismatch could explain some systematic underperformance of PSQ on this collection. Notably, for the NTCIR Chinese collection, which contains only documents from China, PSQ statistically significantly outperforms both QT-BM25 and DT-BM25. For Italian, we note that it is only the DT-BM25 model that numerically outperformed PSQ-HMM. Because this difference is not statistically significant, this may simply be a random effect.

The MAP results at the bottom of Table 2 show similar patterns. Again, the microaverage over all 456 topics indicates that PSQ-HMM is statistically significantly better than either QT-BM25 or DT-BM25. Again we see numerical underperformance of PSQ-HMM in Italian (below DT-BM25) and NeuCLIR Chinese (below QT-BM25 and DT-BM25), and we now also see NeuCLIR Russian slightly below QT-BM25. Overall, we conclude that MAP and R@100 yield similar results when PSQ-HMM is used without index pruning.

We note that PSQ is far more efficient at indexing time than DT-BM25 when using a neural translation model, since PSQ-HMM requires only a sparse table lookup to get translation probabilities and then a sparse matrix multiplication to create sparse query-language vectors. At query time, all models use a sparse inverted index. However, PSQ-HMM indexes with an unpruned alignment matrix are far larger than an index built in the document language would be; indeed, such an index can be as much as 20 times larger than the original document language text! Pruning the alignment matrix reduces not only index size but also query latency, by limiting the number of documents with matching tokens.

7.2 Efficiency-Effectiveness Tradeoff

According to Wang and Oard [52], CDF and PMF thresholding result in a better tradeoff between efficiency and effectiveness compared to top-k filtering due to their larger range of possible outcomes. In this section, we revisit this finding and include in the analysis combinations of pruning methods.

As demonstrated in Figure 2, holding everything else constant, manipulating the maximum CDF has a larger effect on R@100 than using a PMF threshold or top-k filtering. This differs from what Wang and Oard [52] observed for MAP, the measure on which they focused. With a moderate degree of pruning (the middle parts of the graphs), both a PMF threshold and top-k filtering provide roughly 90% of the R@100 of the full index.

However, similar to experiment results in Wang and Oard [52], retrieval results suffer when limiting top-k to a small number of translations (Figure 2(b)). Smaller fan-out more closely approximates one-best translation, which is essentially a statistical MT model without a target language model. SMT generally underperforms neural MT by translation quality measures, and those differences are reflected in retrieval effectiveness. However, using more translation alternatives offsets that, achieving R@100 within 5% of that of the full index. In this case, we observe very similar results for MAP.

Wang and Oard [52] conclude that a pruning strategy leading to a larger dynamic range conveys a preferable efficiency-effectiveness tradeoff. Our experiment results also suggest CDF thresholding has a larger range in R@100; however, we conclude it actually introduces more risk of worse effectiveness. Thus, our analysis reveals that it is not preferable.

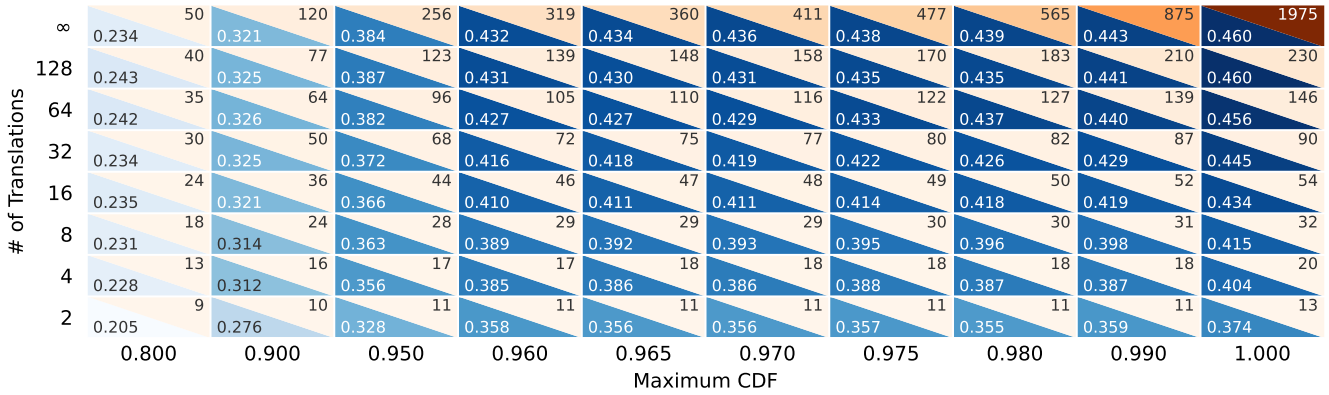
While pruning the alignment matrix can degrade retrieval effectiveness, the resulting inverted index is smaller than an unpruned index, with different pruning techniques resulting in different drops in effectiveness. Figure 3 shows that different techniques also affect index size differently. At the upper right corner of Figure 3(a), where the number of translations is unlimited, and there is no thresholding on CDF, PSQ provides the highest possible R@100 (0.460, in darkest blue), but also the largest index (1,975 GB, in darkest brown). We would prefer high effectiveness (dark blue) with a small index (light brown); top-k filtering demonstrates this. In the rightmost column (using top-k filtering without a CDF threshold), R@100 gradually drops with index size, resulting, for example, in an index of size of 54GB and an R@100 of 0.434 on this Russian collection. Top-k filtering provides a better tradeoff between index size and effectiveness than just a CDF threshold (top row), which requires an index of 319GB to reach 0.432 R@100. Applying both techniques (the cells in the middle), always admits a setting without using a CDF threshold that achieves at least equal R@100 with a smaller index; this indicates that CDF thresholding, contrary to conclusions in Wang and Oard [52], is less suitable for controlling the index size vs. R@100 tradeoff space.

Comparing PMF thresholding to top-k filtering, both of which lead to smaller effectiveness drops than CDF thresholding in Figure 2, top-k filtering is slightly better. With no PMF threshold (bottom row) and allowing at most eight alternate translations, we achieve 0.415 R@100 with an index size of 32 GB. On the other hand, with no limit on translation alternatives and a PMF threshold of 0.01, we only get 0.411 R@100 with a larger 34 GB index. We observe a similar trend in MAP.

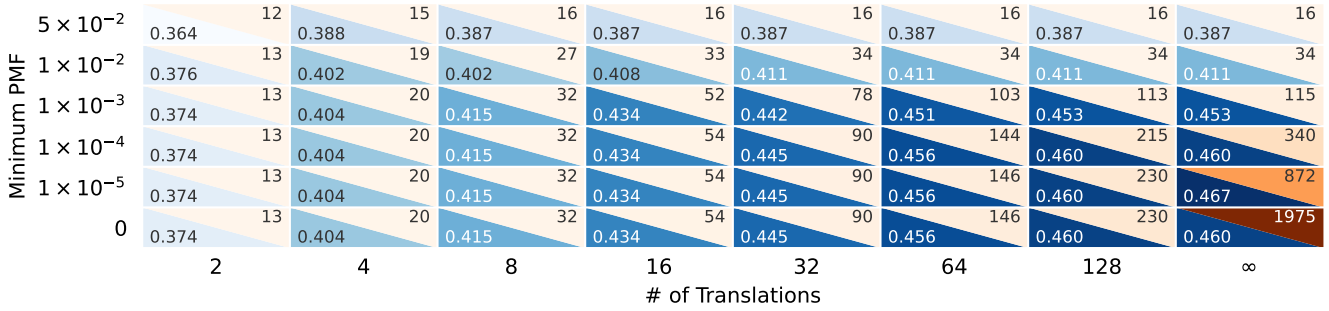
Other collections demonstrate similar trends to those seen on NeuCLIR Russian in Figure 2, scaled according to the size of the collection. We picked the largest collection in our experiment as an example to demonstrate the effect more clearly.

7.3 Pareto Optimality

Finally, we plot all 480 pruning settings as a scatter plot for a Pareto analysis in Figure 4. Pareto-optimal settings (those that achieve the highest R@100 without increasing index size) are starred; connecting them forms the Pareto frontier (dotted lines). On the frontier, all settings are Pareto optimal. An implementer can then choose one setting over another by defining a utility function on that tradeoff.



(a) Top-k Filtering vs. CDF Thresholding



(b) PMF Thresholding vs. Topk-k Filtering

Figure 3: R@100 and index size (GB) on NeuCLIR Russian. Blue and brown represent R@100 and index size, respectively; darker indicates larger values. The ideal is light brown (small index) and dark blue (high R@100).

For all three NeuCLIR languages, there is a diminishing return with increased index size. With a large index, R@100 can be lower than some settings with a smaller index (this is seen in Persian and Russian). That indicates a regularization effect, stemming from a mild PMF or top-k filtering, that trims noise from translations with very low probability. However, such effects are far less dramatic than the sharp falls observed in Wang and Oard [52].

For the CLEF 2003 collection (Figure 4(d-g)), especially in German and Spanish, where the alignment matrices are well-trained, the graphs demonstrate a fan-shaped pattern; this indicates that there is also a diminishing return phenomenon among the 48 models with a given CDF threshold. For French, the graph demonstrates a small regularization benefit when imposing a light CDF threshold, resulting in a Pareto improvement (i.e., pushing the Pareto frontier outward) over those without a CDF threshold. However, indexes without a CDF threshold are still very close to the Pareto frontier, which aligns with our observations and conclusions from the large NeuCLIR collections.

Most settings with no CDF threshold are on or near the Pareto frontier, indicating that changing the index size by top-k filtering and a PMF threshold suffices for Pareto optimality. The left of each graph, where the index is small, shows a steep drop in R@100; such regions are unlikely to be used in practice due to poor effectiveness. It is possible, and in many use cases preferable, to achieve higher effectiveness with slightly larger indexes.

Figure 5 directly shows a comparison of the Pareto frontier between tuning with all three pruning techniques (480 models) against using only a PMF threshold and top-k filtering, keeping the CDF threshold at 1.0 (48 models). By the definition of Pareto optimality, since the set of the models using all pruning techniques is a superset of a set that only uses two, the Pareto frontier when using all techniques (blue dashed lines) provides a tradeoff between index size and R@100 no worse than that using just two techniques. However, applying only PMF and top-k filtering (orange solid lines) is very close to, if not the same as, using all three.

8 SUMMARY

This paper has systematically studied indexing-time PSQ, a strong sparse CLIR method aggregating the salient works by Xu and Weischedel [56], Darwish and Oard [10] and Wang and Oard [52]. We revisit the three pruning techniques studied by Wang and Oard [52]. Index pruning experiments and Pareto analysis show that CDF thresholds provide a sub-optimal effectiveness-efficiency tradeoff, but that PMF thresholds and top-k filtering provide Pareto-optimal operating points evidenced by both R@100 and index size.

While PSQ was developed before the rise of neural IR models, the effectiveness-efficiency tradeoff is similar when pruning models.

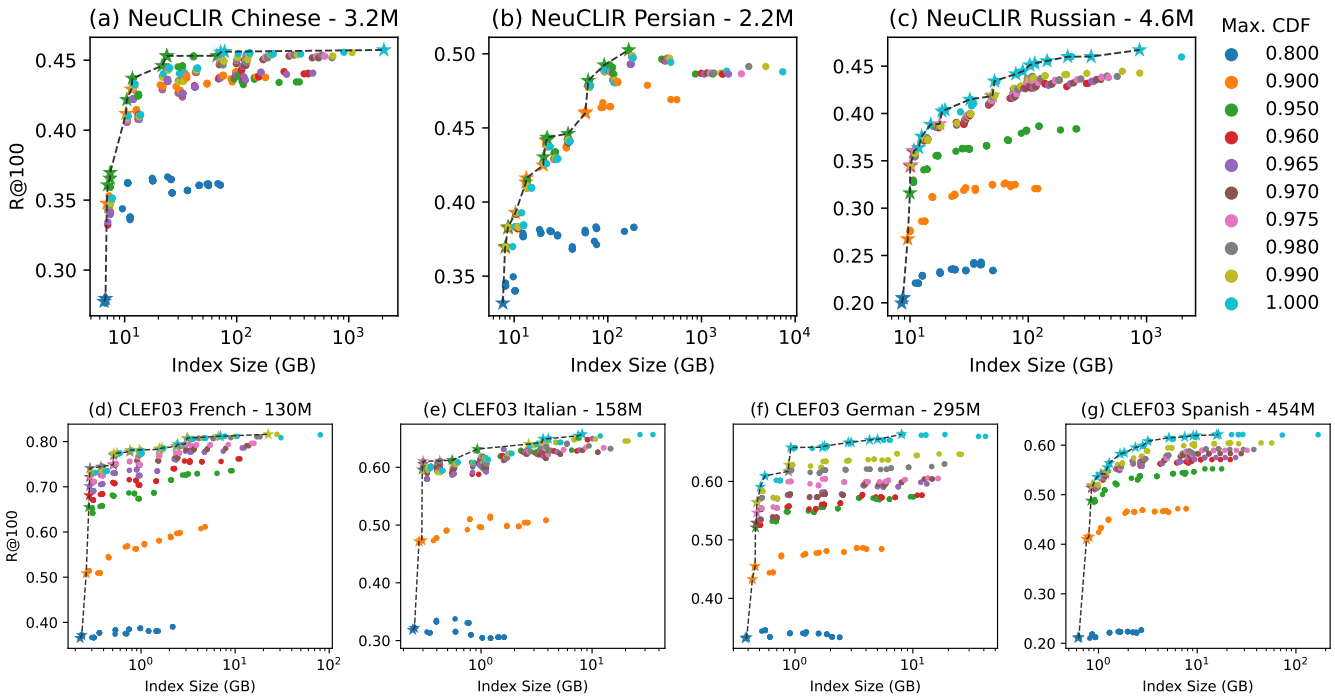


Figure 4: Pareto graphs. Stars indicate Pareto-optimal runs, i.e., those on the Pareto frontier.

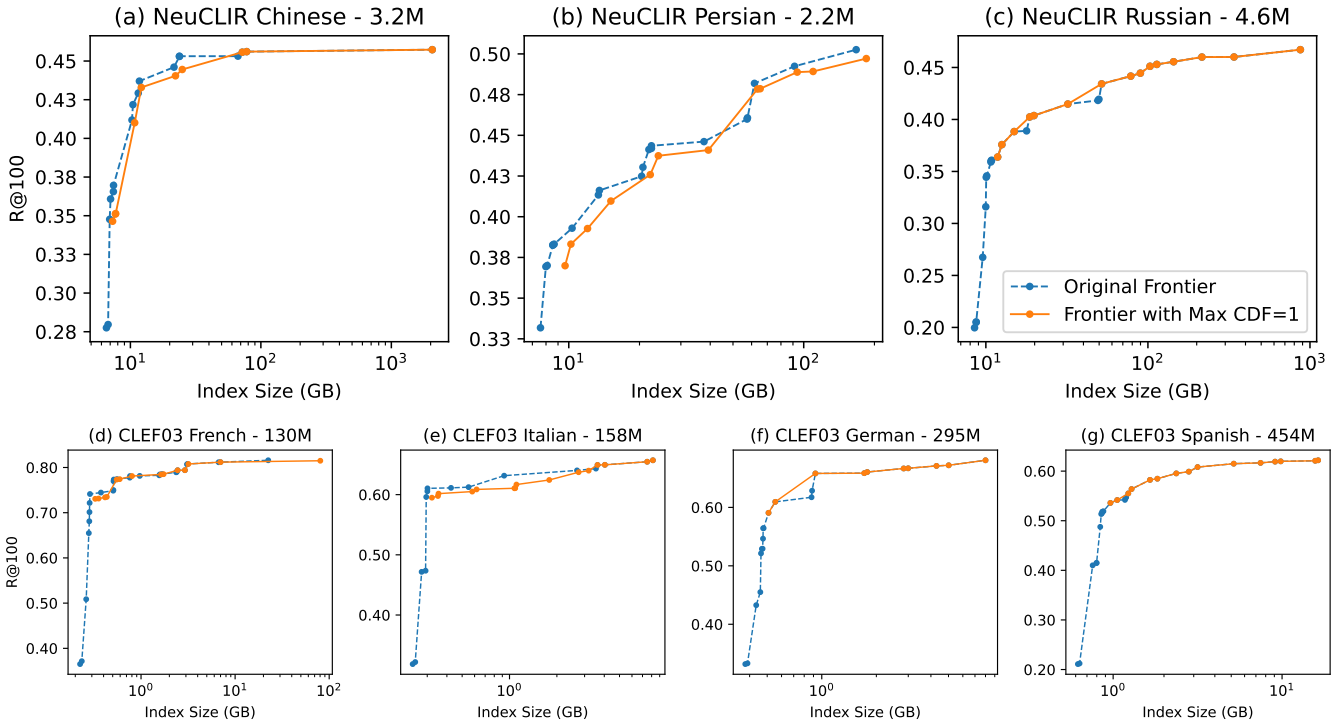


Figure 5: Comparison of Pareto Frontier between using and not using CDF thresholds.

Our findings and analytical methods can inform further tuning of neural IR models, such as BLADE [41], which conceptually also transform documents into query language vectors at indexing time.

REFERENCES

[1] 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

[2] Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2019. Cross-Lingual Transfer of Semantic Roles: From Raw Text to Semantic Roles. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*. Association for Computational Linguistics, Gothenburg, Sweden, 200–210. https://doi.org/10.18653/v1/W19-0417

[3] Lisa Ballesteros and W Bruce Croft. 1997. Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 84–91.

[4] 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

[5] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly.

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

[7] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19, 2 (1993), 263–311. https://aclanthology.org/J93-2003

[8] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005* (2013).

[9] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*. Vol. 520. Addison-Wesley Reading.

[10] 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.

[11] John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 17–24.

[12] 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.

[13] Zi-Yi Dou and Graham Neubig. 2021. Word Alignment by Fine-tuning Embeddings on Parallel Corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 2112–2128.

[14] 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

[15] 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.

[16] Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*. 771–779.

[17] Benjamin Herbert, György Szarvas, and Iryna Gurevych. 2011. Combining query translation techniques to improve cross-language information retrieval. In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*. Springer, 712–715.

[18] Djoerd Hiemstra. 2000. *Using Language Models for Information Retrieval*. Ph.D. Dissertation. University of Twente.

[19] Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering* 11, 3 (2005), 311–325. https://doi.org/10.1017/S1351324905003840

[20] 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

[21] 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).

[22] 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.

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

[24] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*. 177–180.

[25] Wessel Kraaij, Jian-Yun Nie, and Michel Simard. 2003. Embedding web-based statistical translation models in cross-language information retrieval. *Computational Linguistics* 29, 3 (2003), 381–419.

[26] Hrishikesh Kulkarni, Sean MacAvaney, Nazli Goharian, and Ophir Frieder. 2023. Lexically-Accelerated Dense Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 152–162.

[27] Thomas K. Landauer and Michael L. Littman. 1990. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*. 31–38.

[28] Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldaini, and Eugene Yang. 2023. Overview of the TREC 2022 NeuCLIR Track. In *Proceedings of the 31st Text REtrieval Conference* (Gaithersburg, Maryland). https://arxiv.org/abs/2304.12367

[29] Dawn Lawrie, Eugene Yang, Douglas W. Oard, and James Mayfield. 2023. Neural Approaches to Multilingual Information Retrieval. In *Proceedings of the 45th European Conference on Information Retrieval (ECIR)*. https://arxiv.org/abs/2209.01335

[30] Jurek Leonhardt, Koustav Rudra, Megha Khosla, Abhijit Anand, and Avishek Anand. 2022. Efficient neural ranking using forward indexes. In *Proceedings of the ACM Web Conference 2022*. 266–276.

[31] 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

[32] Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2012. Learning to predict response times for online query scheduling. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 621–630.

[33] Antonio Mallia, Michal Siedlaczek, Joel Mackenzie, and Torsten Suel. 2019. PISA: Performant indexes and search for academia. *Proceedings of the Open-Source IR Replicability Challenge* (2019).

[34] J Scott McCarley. 1999. Should we translate the documents or the queries in cross-language information retrieval?. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. 208–214.

[35] David R H Miller, Tim Leek, and Richard M Schwartz. 1999. A hidden Markov model information retrieval system. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 214–221.

[36] Teruko Mitamura, Hideki Shima, Tetsuya Sakai, Noriko Kando, Tatsunori Mori, Koichi Takeda, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, and Cheng-Wei Lee. 2010. Overview of the NTCIR-8 ACLIA Tasks: Advanced Cross-Lingual Information Access. In *Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-8, National Center of Sciences, Tokyo, Japan, June 15-18, 2010*, Noriko Kando, Kazuaki Kishida, and Miho Sugimoto (Eds.). National Institute of Informatics (NII), 15–24. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings8/NTCIR/01-NTCIR8-OV-CCLQA-MitamuraT.pdf

[37] Ali Montazeralghaem, Razieh Rahimi, and James Allan. 2019. Term discrimination value for cross-language information retrieval. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 137–140.

[38] 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* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1581–1584. https:

//doi.org/10.1145/3397271.3401270

[39] Suraj Nair, Anton Ragni, Ondrej Klejch, Petra Galuščáková, and Douglas Oard. 2020. Experiments with cross-language speech retrieval for lower-resource languages. In *Information Retrieval Technology: 15th Asia Information Retrieval Societies Conference, AIRS 2019, Hong Kong, China, November 7–9, 2019, Proceedings 15*. Springer, 145–157.

[40] 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*. Springer, 382–396.

[41] 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)*.

[42] Douglas W Oard. 1998. A comparative study of query and document translation for cross-language information retrieval. In *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas AMTA'98 Langhorne, PA, USA, October 28–31, 1998 Proceedings 3*. Springer, 472–483.

[43] Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29, 1 (2003), 19–51. https://doi.org/10.1162/089120103321337421

[44] Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics* (2016).

[45] Ari Pirkola. 1998. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 55–63.

[46] Razieh Rahimi, Ali Montazeralghaem, and Azadeh Shakery. 2020. An axiomatic approach to corpus-based cross-language information retrieval. *Information Retrieval Journal* 23 (2020), 191–215.

[47] Mohammad Sadegh Rasooli, Noura Farra, Axinia Radeva, Tao Yu, and Kathleen McKeown. 2018. Cross-lingual sentiment transfer with limited resources. *Machine Translation* 32, 1/2 (2018), 143–165. http://www.jstor.org/stable/44988391

[48] 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

[49] Páraic Sheridan and Jean Paul Ballerini. 1996. Experiments in multilingual information retrieval using the SPIDER system. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. 58–65.

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

[51] Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*. https://aclanthology.org/C96-2141

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

[53] Xiao Wang, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2022. An inspection of the reproducibility and replicability of TCT-ColBERT. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2790–2800.

[54] Dan Wu, Daqing He, Heng Ji, and Ralph Grishman. 2008. A study of using an out-of-box commercial MT system for query translation in CLIR. In *Proceedings of the 2nd ACM workshop on Improving non english web searching*. 71–76.

[55] Hua Wu and Haifeng Wang. 2005. Improving Statistical Word Alignment with Ensemble Methods. In *Second International Joint Conference on Natural Language Processing: Full Papers*. https://doi.org/10.1007/11562214_41

[56] 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.

[57] Rabih Zbib, Lingjun Zhao, Damianos Karakos, William Hartmann, Jay DeYoung, Zhongqiang Huang, Zhuolin Jiang, Noah Rivkin, Le Zhang, Richard Schwartz, et al. 2019. Neural-network lexical translation for cross-lingual IR from text and speech. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 645–654.