ABSTRACT

Title of Dissertation: EFFECTIVE AND EFFICIENT SEARCH ACROSS LANGUAGES

Suraj Rajappan Nair
Doctor of Philosophy, 2023

Dissertation directed by: Professor Douglas William Oard
College of Information Studies
Department of Computer Science
Institute for Advanced Computer Studies

In the digital era, the abundance of text content in multiple languages has created a need to develop search systems to meet the diverse information needs of users. Cross-Language Information Retrieval (CLIR) plays an essential role in overcoming language barriers, allowing users to retrieve content in a language that differs from their query language. However, a challenge in designing retrieval systems lies in balancing their *effectiveness*, which reflects the quality of the ranked outputs, with their *efficiency*, which encompasses document processing latency at indexing time (indexing latency) and content retrieval latency at query time (query latency). This dissertation focuses on designing neural CLIR systems that offer a Pareto-optimal balance between the competing objectives of effectiveness and efficiency.

While neural ranking models that rely on query-document term interactions, such as cross-encoder models, are highly effective, they are computationally prohibitive for pro-

cessing large document collections in response to every query. One solution is to build a cascaded pipeline of multiple ranking stages, where a first-stage retrieval system generates a set of documents, which is then reranked by the cross-encoder. Ensuring that the first-stage retrieval system produces an accurate and rapid triage of large document collections is crucial for the success of the cascaded pipeline. This dissertation introduces BLADE, a first-stage system that strikes a better balance between retrieval effectiveness and indexing/query latency on the Pareto frontier by leveraging traditional inverted indexes. Once a smaller set of documents is generated, less efficient techniques can be applied to the output from the first stage. In addition, this dissertation introduces ColBERT-X, the best-known second-stage technique in terms of the balance between retrieval effectiveness and indexing latency on the Pareto frontier. To further tackle the efficiency challenges of cross-encoders, this dissertation introduces CREPE, an approach that optimizes the tradeoff between retrieval effectiveness and query latency.

While traditional CLIR methods rely on Machine Translation (MT) to address vocabulary mismatches between queries and documents, neural techniques match terms in a shared vector space, serving as a complementary source. Fusion techniques help leverage the synergies between these complementary methods by creating ensembles, and the design space of CLIR allows for multiple such ensembles. This dissertation highlights the complementary nature of BLADE and ColBERT-X with traditional CLIR approaches and demonstrates further effectiveness gains by ensembling them without adversely affecting the indexing-time efficiency. These results pave the way for the development of scalable CLIR systems with a better tradeoff between effectiveness and indexing speed.

EFFECTIVE AND EFFICIENT SEARCH MODELS ACROSS LANGUAGES

by

Suraj Rajappan Nair

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Dr. Douglas W. Oard, Chair/Advisor
Dr. Philip Resnik
Dr. Marine Carpuat
Dr. Jordan Boyd-Graber
Dr. Dawn Lawrie

Dedication

To my parents, **Su**dha and **Raj**appan,

without whom I am incomplete

Acknowledgments

I extend my heartfelt gratitude to everyone who has contributed to this dissertation and my graduate school experience.

First, I want to thank my advisor, Doug Oard, for the invaluable opportunity to be mentored by him. Throughout my journey, he gave me the freedom to explore research directions while providing the necessary guidance and asking the right questions to keep me on track. His valuable insights and constructive critiques have played an important role in shaping not only my work but also my development as a researcher. He continues to inspire me to become a better researcher and never ceases to amaze me with his sense of humor. I am deeply grateful for his time, effort, concern, and guidance.

I want to thank my committee members - Philip Resnik, Marine Carpuat, Tom Goldstein, and Dawn Lawrie for their valuable feedback, which helped improve this dissertation. My decision to join the CLIP lab was primarily influenced by Marine's CL1 course, which was further strengthened when I took Philip's CL2 course. Thanks to Vanessa Frias-Martinez for providing me with my first research experience as a budding graduate student. I owe special thanks to Philip for his invaluable mentoring and advice and for introducing me to my advisor, Doug. I also want to express my heartfelt thanks to Dawn Lawrie and James Mayfield for their exceptional mentorship and support. They provided me with all the resources I needed and allowed me to collaborate with folks from the Johns

Hopkins HLTCOE, which was an invaluable experience for me. I express my gratitude to the MATERIAL project for funding the first half of my PhD and to the HLTCOE for their continued support until my graduation. I am grateful to Prof. Rajiv Gandhi, whose encouragement led me to pursue a PhD, and I couldn't have done this without his support.

I was fortunate to be guided initially by senior CLIP lab members, including Sudha Rao, Yogarshi Vyas, Ahmed Elgohary, Weiwei Yang, Xing Niu, and Rashmi Sankepally. I enjoyed my interactions with several former and present CLIP lab members and fellow MATERIAL project members, including Petra Galuscakova, Han Chin-Shing, Joe Barrow, Elena Zotkina, Peter Rankel, Mahmoud Sayed, Xin Qian, Weijia Xu, Aquia Richburg, Jiahui Wu, Denis Peskov and many more. I want to especially thank Petra Galuscakova for unofficially mentoring me and being a constant source of support and endless fun/research discussions. I also enjoyed interacting with several HLTCOE folks, including Eugene Yang, Kenton Murray, Orion Weller, Kevin Duh, Paul McNamee, and Marc Mason, among others. I want to thank Eugene Yang for mentoring me, sharing his coding expertise, and for our fun late-night conversations, especially during deadlines.

I had the privilege of being part of two summer internships where I received invaluable mentorship. During my internship at Raytheon BBN, Damianos Karakos, Le Zhang, and Bonan Min provided me with excellent guidance and support. At Amazon, I had a fulfilling experience thanks to the exceptional mentorship provided by Alessandro Moschitti and Eric Lind. Furthermore, I fondly recall the great interactions and amazing moments spent with my colleagues, including Stefano Campese, Matteo Gabburo, Luca Di Liello, and Ivano Lauriola.

I want to thank my friends in the US - Jay Ghurye, Nidhi Shah, Jigar Bhati, Niyati

Mehta, Aakash Moni, Prashant Nagdeve, Deep Barot, Prakhar Joshi, Trevor Adriaanse, among others who made this journey so much more fun and enjoyable. I also want to express my gratitude to my cousins and friends in India, who always helped me to relax during my short visits. I could not thank enough my sister and my best friend, Kshama Doshi, who has continually remained by my side through all phases of my grad school. Most importantly, I would like to thank my girlfriend, Aakriti Mittal, for her patience and understanding, despite the sacrifices that came along with my (sometimes) busy schedule. She is the person I can count on, no matter what. Last but not least, I owe a huge debt of gratitude to my parents, Sudha and Rajappan, whose unwavering support and unconditional love have been a constant source of motivation.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BERT	Bidirectional Encoder Representations from Transformers
BPE	Byte Pair Encoding
CDF	Cumulative Distribution Function
CLEF	Cross-Language Evaluation Forum
CLIR	Cross-Language Information Retrieval
FAISS	Facebook AI Similarity Search
HMM	Hidden Markov Model
IDF	Inverse Document Frequency
IR	Information Retrieval
LSI	Latent Semantic Indexing
MAP	Mean Average Precision
mBERT	Multilingual Bidirectional Encoder Representations from Transformers
MPLM	Multilingual Pretrained Language Model
MS MARCO	Microsoft Machine Reading Comprehension
NeuCLIR	Neural CLIR
NMT	Neural Machine Translation
PLM	Pretrained Language Model
PMF	Probability Mass Function
PSQ	Probabilistic Structured Queries
PRF	Pseudo-Relevance Feedback
R@100	Recall @ 100
RoBERTa	Robustly Optimized BERT Approach
SMT	Statistical Machine Translation
SQuAD	Stanford Question Answering Dataset
SVD	Singular Value Decomposition
TF	Term Frequency
TREC	Text Retrieval Conference
XLM	Cross-Lingual Language Model
XLM-R	XLM-RoBERTa
WMT	Workshop in Machine Translation

Chapter 1: Introduction

Search plays a crucial role in our daily lives as we constantly strive to seek information to meet our needs. The availability of digital content in numerous languages accessible through different electronic devices has made this process easier. Building search systems that can effectively retrieve such relevant pieces of information in an efficient manner has been a long-standing goal of the Information Retrieval (IR) community. While much emphasis has been placed on monolingual retrieval, finding content in the same language as the query, with English being a particularly dominant language, a significant portion of the world's population is bilingual. Approximately 43% of people are fluent in two languages, thus highlighting the importance of supporting search systems that can bridge language barriers and provide relevant information in any language.[1] The general problem of finding content in multiple languages for a search query expressed in one language is called Multilingual Information Retrieval (MLIR). However, this dissertation focuses on a specific case of MLIR, known as Cross-Language Information Retrieval (CLIR), where the goal is to find content expressed in one language (e.g., French) using a query expressed in a different language (e.g., English).

There are currently 7,168 languages spoken worldwide [52], but only a fraction of

[1] `http://ilanguages.org/bilingual.php`

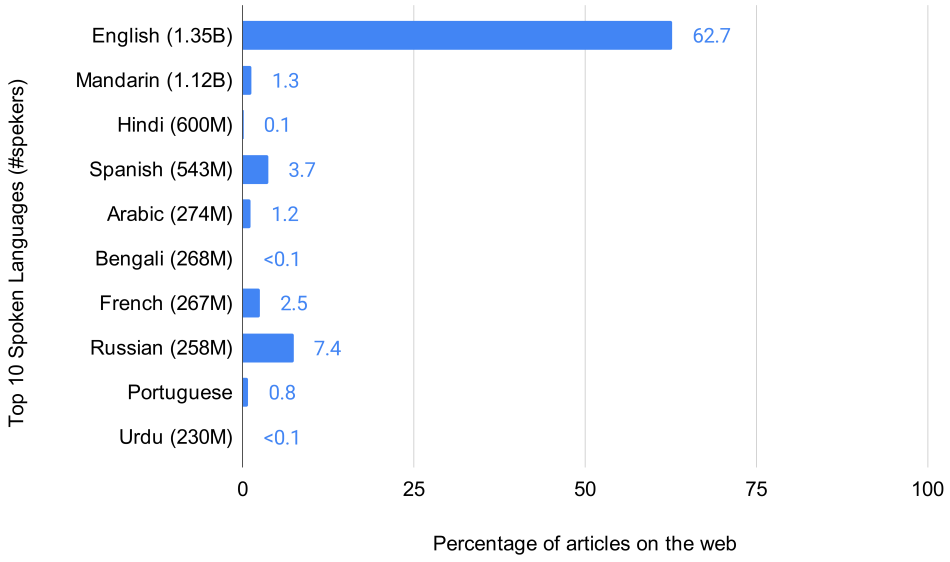


Figure 1.1: Percentage of the top 10M visited web articles in different languages as of September 2021. The languages are sorted in decreasing order based on the total number of speakers of both the first language (L1) and the second language (L2). Sources: `https://w3techs.com/technologies/overview/content_language`, `https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers`

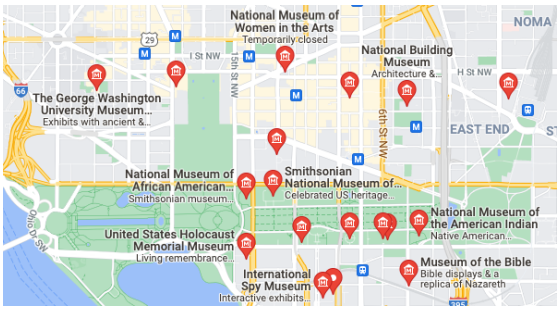
these languages are represented on the web or have well-documented resources, leading to a digital language divide.[2] Factors such as the demographic and socio-economic characteristics of the regional population contribute to this divide [67]. As shown in Figure 1.1, English, the most widely spoken language, has the majority of web content (~63%), while Mandarin Chinese and Hindi, the second and third most widely spoken languages, respectively, together account for less than 2% of the web content. Building human language technologies, such as Natural Language Processing (NLP) and IR systems, which serve languages other than English, has numerous benefits. Ruder [158] emphasizes the significance of building NLP systems that serve languages other than English by presenting six perspectives ranging from societal to cognitive. Additionally, Bender and Friedman [17]

[2] `http://labs.theguardian.com/digital-language-divide/`

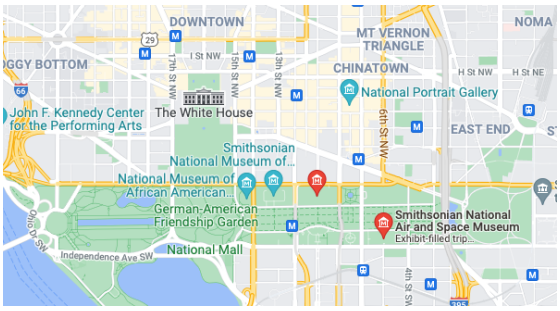
highlight the need for creating data statements to mitigate bias and enable fair scientific progress. This has led to the rise of the #BenderRule -"always state the language you're working on" [16]. Although these recommendations were initially made for NLP systems, they can be applied more broadly to all technologies that involve human language, including NLP, IR, and Automatic Speech Recognition.

Building search systems that can cross language barriers is crucial in reducing the language divide, and CLIR systems provide a means to achieve this goal. Relevant information can exist in different modalities, including text, speech, or even images, but in this dissertation, we will focus on text retrieval systems. The use cases of CLIR systems [62] can be broadly divided into two categories: (1) applications where the user does not speak the language of the documents, and (2) applications where the user can comprehend the document but wants to use a different language for their query. Here we describe application scenarios in each category where the CLIR systems can be useful.

- Search in many languages suffers from *information asymmetry* where much of the information is present in a limited number of languages, as shown in Figure 1.1 and from *information scarcity* due to lack of available content as shown in Figure 1.2 where the number of returned items for an English query are far more than for a query in Bengali. In such cases, CLIR can provide access to information that would otherwise be unavailable. An example of this is Cross-Lingual Question Answering (CLQA) [43, 65, 111, 157] where the system uses the information from relevant documents retrieved by a CLIR system to provide a direct answer to a question. One specific example of this, the Cross-lingual Open-Retrieval Question Answering



(a) English query "museum"



(b) Bengali query "জাদুঘর"

Figure 1.2: Google Maps search results for the query *museum* expressed in English and Bengali.

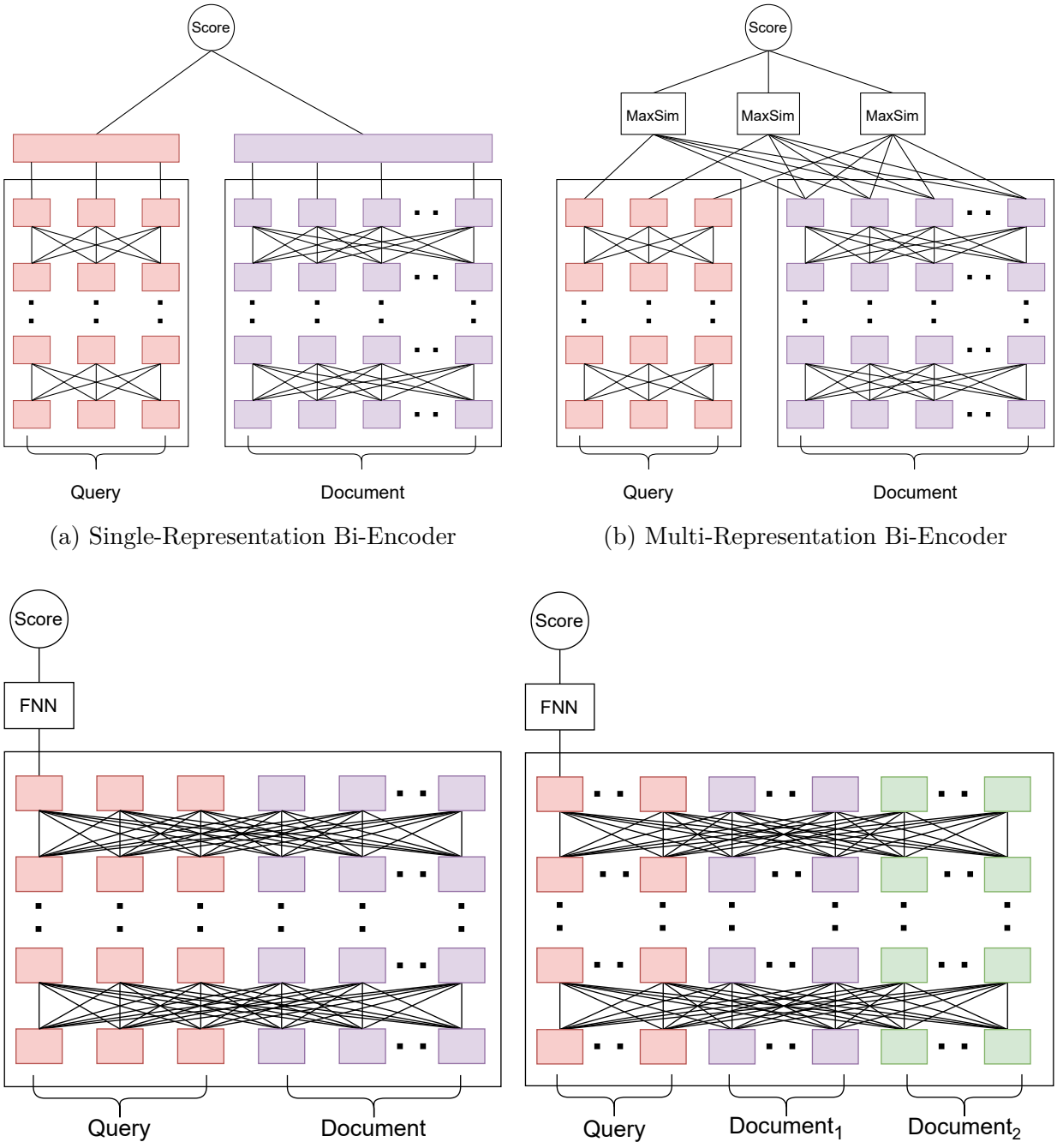
(XOR-QA) system [8], supports search for information in high-resource language collection (e.g., English Wikipedia articles) using queries expressed by users in language with limited resources. Another related example is Cross-Lingual Knowledge Grounded Conversation (CKGC) [174], where the system uses knowledge sources in high-resource languages to help conversational agents generate better responses in other languages that lack resources.

- CLIR systems offer a valuable solution for users who are proficient in multiple languages but prefer or feel more confident using a specific language to express their information needs, particularly in regions with multiple commonly used languages or countries with more than one official language such as Canada (English/French), Belgium (Dutch/French/German), and Spain (Spanish/Catalan/Basque). Moreover, with 24 official languages in the European Union and 23 official languages in India, the implementation of CLIR systems can bring significant benefits to these regions.

One of the central challenges in CLIR is the vocabulary mismatch between queries and documents stemming from differences in the language used to express them. To bridge

this vocabulary gap, multiple methods can be utilized to create representations of queries and documents that facilitate cross-language matching. For instance, traditional CLIR systems rely on machine translation (MT) systems' 1-best output to match query and document terms. Matching can be done in the query language using original queries and translated documents produced by MT or in the document language using MT queries and original documents. Probabilistic Structured Queries (PSQ) offer an alternative approach that involves mapping terms from one language to another using term-to-term probabilities from translation tables. These probabilities are estimated from parallel corpora consisting of sentences that are equivalent in meaning, using traditional statistical approaches [51, 70, 137]. By using translation tables, multiple alternatives for a given term can be considered during matching, which can be thought of as lexical expansion, with probabilities serving as weights. This form of matching with multiple alternatives provides an advantage over the 1-best MT output, which can be prone to mismatches due to translation errors.

Another method for bridging the vocabulary gap is to perform matching in a shared embedding space where queries and documents are represented as fixed-length vectors or embeddings. Recent improvements in language modeling techniques have resulted in the development of pretrained language models (PLMs) that can produce contextual term embeddings, which can change based on the surrounding terms or the context. These PLMs are initialized with several layers of transformers [180] and are pretrained on a large corpus of text in a self-supervised manner. To create a model tailored for the IR task, the most common approach is to fine-tune the off-the-shelf PLMs on large-scale retrieval collections, adjusting the PLM's parameters to learn the downstream relevance task. Figure 1.3 shows the different kinds of neural IR models, of which there are two main types:



(a) Single-Representation Bi-Encoder

(b) Multi-Representation Bi-Encoder

(c) Pointwise Cross-Encoder

(d) Pairwise Cross-Encoder

Figure 1.3: Neural IR models with varying degrees of query-document matching using Pretrained Language Model encoder denoted as ▭. The expressive power of the models increases from (a) to (d), which leads to a corresponding decrease in model efficiency, with each subsequent model scoring fewer documents for a fixed time unit. This dissertation focuses on Neural CLIR architectures based on Models (a), (b), and (c) while leaving the exploration of Model (d) to future work.

1. Bi-Encoder models separate the computation of query and document contextual term embeddings and allow for matching the representation using custom similarity functions. This offers two advantages. Firstly, matching can be carried out entirely in the embedding space, which can alleviate the vocabulary mismatch problem of keyword-based matching models. Secondly, document representations can be precomputed offline and stored in indexes specializing in nearest neighbor retrieval techniques for dense embeddings, such as FAISS [83], or in inverted indexes for sparse embeddings. Existing bi-encoder models can be classified into two categories. The first category is single-representation models, as depicted in Figure 1.3a. These models generate an aggregated vector for queries and documents, which reduces storage requirements. However, this approach sacrifices representation fidelity since it compresses information to a single vector. The second category is multi-representation models, as shown in Figure 1.3b. These models utilize additional signals from matching individual query and document term embeddings but require indexing multiple vectors, which comes at the cost of increased storage requirements.

2. Cross-Encoder models compute the full interaction between the query and document contextual term embeddings using multiple transformer layers. While this approach allows the model to exploit joint context in queries and documents, it comes at an additional computational cost due to the quadratic time complexity of self-attention in the transformer layer. Hence, these models are often employed in a retrieve-and-rerank pipeline, where a first-stage system, such as BM25 [155], retrieves a limited number of documents, which are subsequently reranked with a cross-encoder model.

The type of cross-encoder can be determined based on the number of documents fed to the model at query time. The pointwise cross-encoder model, as depicted in Figure 1.3c, generates a score for a single document given a query. In contrast, the pairwise cross-encoder model, as shown in Figure 1.3d, generates a score for a pair of documents. For CLIR, we investigate the application of pointwise cross-encoders using different first-stage CLIR systems.

The focus of this dissertation is to design neural CLIR systems that can handle cross-language texts using multilingual pretrained language models (MPLMs), such as mBERT [48] and XLM-R [40]. However, these MPLMs are limited by a sequence length constraint, allowing them to handle only a fixed number of terms (e.g., 512). This presents a challenge when processing longer documents, especially for cross-encoders that utilize concatenated sequences of queries and documents. To address this issue, common techniques include truncating documents to the first passage of a fixed length (≤ 512 or creating a set of overlapping passages/sentences with a fixed length and a stride. However, this approach can be viewed as a static choice between one or all passages for scoring, which represents two extremes of a broad spectrum, without leveraging the text signals in the underlying passages. As a result, this opens up an avenue to explore better passage selection strategies that can reduce the number of query-passage pairs for scoring with a cross-encoder and improve query latency.

The combination of cross-language input sequences and MPLMs creates a further challenge of learning a shared cross-lingual embedding space that can map query and document terms into a conformal representation. MPLMs are trained on monolingual texts in

multiple languages and, therefore, are not ideal for matching cross-language texts. Additionally, the lack of high-quality datasets in the CLIR setting poses a challenge to learning meaningful task-specific representations of queries and documents in the context of CLIR. In contrast, the availability of large-scale retrieval collections, such as MS MARCO, has facilitated the development of neural IR models for retrieving English content in the monolingual retrieval setting. With improvements in translation technologies, there is potential for building neural CLIR models using large-scale training collections for CLIR by translating the monolingual collections.

The primary objective of this dissertation is to design neural CLIR systems that strike an optimal balance between effectiveness and efficiency. These broad terms encompass various aspects related to the retrieval process. Specifically, effectiveness refers to the quality of the retrieval output produced by a CLIR system, which is assessed using established evaluation metrics. Efficiency is considered from two phases of the retrieval process: the indexing phase and the querying phase. Efficiency at the indexing phase is defined as the overall indexing latency, which involves tasks such as running machine translation (if necessary), generating bi-encoder representations, and storing them in FAISS or inverted indexes. Efficiency at the querying phase refers to the overall retrieval latency, the time taken to produce the retrieval outputs. Each neural CLIR system creates a trade-off space where indexing latency, query latency, and retrieval effectiveness are in tension. Achieving an optimal balance between these factors is crucial for building scalable CLIR systems, which can consist of one or more systems that can be deployed in real-world applications. To achieve this goal, we focus on Pareto-optimal CLIR systems that offer the best retrieval effectiveness for a given indexing latency or query latency. By exploring the Pareto frontier,

we can identify systems that provide the most favorable trade-offs between effectiveness and efficiency. This approach can assist search practitioners in selecting the best system for their specific application requirements.

1.1 Research Questions

In this dissertation, we focus on these broad research questions:

RQ1. Can we improve the query latency of a CLIR cross-encoder without degrading the retrieval effectiveness by better passage selection strategies?

RQ2. Can CLIR bi-encoders trained on translated retrieval collections improve retrieval effectiveness over traditional CLIR systems?

RQ3. Which of the neural CLIR systems lie on the Pareto frontier of retrieval effectiveness and indexing latency?

RQ4. Which of the neural CLIR systems lie on the Pareto frontier of retrieval effectiveness and query latency?

1.2 Contributions

The primary objective of this dissertation is to build neural CLIR systems that lie on the Pareto frontier of retrieval effectiveness and indexing or query latency. Our major contributions can be divided into three categories: System contributions (S), Data contributions (D), and Code contributions (C).

1.2.1 System Contributions

S1. We build a passage selection strategy, CREPE, which uses a first-stage CLIR system to find a Pareto-optimal balance between retrieval effectiveness and query latency of cross-encoders.

S2. We build a training dataset using the CREPE strategy for fine-tuning CLIR cross-encoders and demonstrate gains in retrieval effectiveness across different CLIR retrieve-and-rerank pipelines.

S3. We build a multi-representation bi-encoder, ColBERT-X, initialized with an XLM-R encoder and trained on machine-translated retrieval collections.

S4. We build a pseudo-relevance feedback model relying on term expansion within the embedding space of ColBERT-X to boost retrieval effectiveness further.

S5. We build two single-representation bi-encoders as first-stage systems, SPLADE-X and BLADE, initialized with a multilingual BERT and a pruned bilingual BERT, respectively, balancing retrieval effectiveness with query latency.

S6. We show a CLIR ensemble pipeline of BLADE, ColBERT-X, and traditional CLIR approaches achieve comparable retrieval effectiveness with lower indexing latencies compared to CLIR systems applied on translated documents.

1.2.2 Dataset Contributions

D1. We build new bilingual passage-aligned corpora from existing sentence-aligned parallel texts to provide additional context for modeling.

D2. We create PSQ translation tables in multiple language pairs and release them here: `https://github.com/hltcoe/PSQ/aligners/ttables/`.

1.2.3 Code Contributions

C1. We release the code of SPLADE-X and BLADE here: `https://github.com/hltcoe/BLADE`

C2. We release the code of ColBERT-X here: `https://github.com/hltcoe/ColBERT-X`

C3. We release the reference implementation of PSQ here: `https://github.com/hltcoe/PSQ`.

1.3 Outline

Chapter 2 provides the background for this dissertation and the related work. The remaining chapters are structured as follows. Chapter 3 focuses on building retrieve-and-rerank pipelines that use different first-stage CLIR systems to select the best passage(s) for scoring with a pointwise CLIR cross-encoder. Chapter 4 introduces ColBERT-X, a multi-representation bi-encoder model trained on a CLIR dataset created from translations of the MS MARCO retrieval collections. Chapter 5 presents two sparse bi-encoder models, SPLADE-X and BLADE, which leverage inverted indexes to balance retrieval effectiveness

with query latency. In Chapter 6, we explore different CLIR systems, including model ensembles, to find a Pareto-optimal balance between retrieval effectiveness and indexing latency. Finally, Chapter 7 concludes the dissertation by discussing the limitations of our work, future research directions, and the practical implications of our findings.

Chapter 2: Background

The primary goal of an Information Retrieval system is to return a ranked list of "documents" that satisfy the information need of an end-user expressed in the form of a "query". In the case of CLIR, the documents and the queries differ in the language in which they are expressed. First, we mention the evaluation measures commonly used to evaluate retrieval systems in Section 2.1. Section 2.2 provides an overview of the research history in CLIR. We then present core technologies that are used in CLIR systems, which include translation techniques (Section 2.3.1) and neural IR (Section 2.4). Finally, in Section 2.6, we present the Pareto-optimality framework that will be used throughout the dissertation to analyze the effectiveness-efficiency tradeoffs.

2.1 Evaluation

One of the main goals while building IR systems is to evaluate how well the system is performing on a given test collection. A test collection typically consists of a set of documents that could potentially contain the information sought by the user, a set of queries that represent the information needs expressed by the users, and a set of relevance judgments that indicate how relevant a document is with respect to a specific query. Relevance judgments are usually created by human annotators who evaluate documents and assign

labels in response to a query. These labels can be binary, indicating whether the document is relevant or not for the given query or graded, using a preference scale such as not relevant, somewhat relevant, or very relevant. Depending on the intended goal, different aspects of an IR system could be evaluated.

One aspect is to measure the quality of the ranked list of documents for a given query in satisfying the user's information needs. This aspect is also referred to as the *effectiveness* of the IR system. Based on the type of relevance judgments, there exist several measures that are commonly employed to evaluate the model's effectiveness. However, in this dissertation, we mostly use two measures for effectiveness, Recall (R) and Mean Average Precision (MAP). Recall computes the proportion of relevant documents that are retrieved. In other words, Recall measures how many relevant documents are retrieved. However, Precision computes the proportion of retrieved documents that are relevant. In short, Precision measures how many retrieved documents are relevant. Average Precision (AP) refers to the average of the precision values computed at different recall levels for a single query. AP measures the quality of a single query, considering the positions of relevant items in the ranked list of documents. Mean Average Precision calculates the mean of the different AP values for each query, providing a single score for the entire set of queries and thus measuring the overall effectiveness of the retrieval system. Instead of computing these measures on the entire ranked list of documents, these can be computed on a top-k list of documents (e.g., R@k). For a specific query, both Recall and Mean Average Precision range from 0 to 1, with higher values indicating better retrieval effectiveness. The choice of an evaluation measure depends on how well it correlates to the actual downstream task.

2.2 History of CLIR

In the 1960s, the earliest known efforts to build CLIR systems [160, 164] were made, which involved the expansion of General Inquirer [171] and SMART [161] document retrieval systems in retrieving content in German using English queries. During this time, both queries and documents were described using controlled vocabulary descriptors such as metadata and subject headings. These descriptors were organized hierarchically in a thesaurus, a reference database used to classify words based on their semantic relationship. Adding entry vocabulary in other languages to the thesaurus enabled cross-language search to be facilitated [136].

In the 1990s, the focus of CLIR systems shifted from using a thesaurus-based search with a limited vocabulary to allowing users to query with any vocabulary they choose. Retrieval was typically done by generating similar representations for both the queries and documents and computing their similarity using custom retrieval models. Two techniques used during this period included the Multilingual Vector Space Model [97, 109] and Machine Translation [150], combined with keyword-based matching models such as BM25 [155] or the Query Likelihood Model [147]. For more information on the development of CLIR systems during this time frame, we refer the readers to the CLIR survey [133].

The recent developments in IR include the emergence of neural ranking models to estimate query-document relevance. These models can be initialized with term embeddings that could be context-independent (e.g., Word2Vec [122, 124] or GLoVE [144]) or using pretrained language models (e.g., BERT [48] or RoBERTa [110]) to generate contextual embeddings. For more details, we refer the readers to the Galuščáková et al. [62] survey.

2.3 Translation techniques for CLIR

One of the most commonly used methods for building CLIR systems involves using translation tools to match query terms and document terms in different languages. The matching can occur in three spaces, i) mapping query terms to the document language, which is referred to as query translation, ii) mapping document terms to the query language, which is referred to as document translation, or iii) where both query and document terms are mapped to a common representation in a third language. While this addresses what to translate (query or document), the choice of "term", which is the unit of text that is translated or indexed by the system, can vary. The common choices include multi-word expressions [93], whitespace-separated words, word forms derived from stemming or lemmatization algorithms, overlapping character n-grams [120] or other subword units (e.g., Byte Pair Encodings (BPE), [165] SentencePieces [95]).

Once we choose which terms to translate, we can leverage different resources to generate the required translations for bridging the language barrier. One way to do this is by using a lexicon of term mappings (e.g., a bilingual dictionary) to search for a term in one language and replace it with the linked term(s) in another language. This approach is commonly referred to as dictionary-based translation. The usefulness of this approach crucially depends on how good the coverage of terms in the lexicon is. Another approach is to use parallel text (i.e., translation-equivalent texts, usually sentences) to learn translation mappings that include a probability of translating a term from one language to another. This can be done explicitly using 1) Statistical Machine Translation (SMT) models [195] that rely on statistical word aligners (e.g., GIZA++ [137] or BerkleyAligner [70]), 2) Neural

Machine Translation (NMT) models [14, 205] that leverage the low-dimensional representations of terms, or can be done implicitly by using a complete SMT or NMT system that produces 1-best or n-best results.

Once we have a list of translation alternatives with probabilities for a term, we can use them to perform retrieval in a manner similar to the monolingual setting. In general, IR systems function on three elements: (1) term frequency (TF) computed based on the number of occurrences of a term in a document, (2) inverse document frequency (IDF) computed based on the number of documents in which a term occurs, and (3) the length of the document. While in monolingual models, we could simply generate such features by counting the terms in a document, in a CLIR setting, we generate expected counts for each query term in a target document. This is the key idea behind the approach referred to as Probabilistic Structured Queries (PSQ) [47], where we estimate term counts for a query q in document d using the translation probabilities of a query word given the document terms.

2.3.1 PSQ

In this dissertation, we use a PSQ framework implemented using a Hidden Markov Model (HMM)[195], which is referred to as PSQ-HMM. Assuming the query language is English and the document language is non-English, PSQ-HMM estimates the relevance of a document in non-English given an input query in English. Viewing the model as a Hidden Markov Model [125], the first state θ_e generates English terms, whereas the second state θ_d generates non-English terms. Each English query q may consist of n terms $t_1, ..., t_n$. The

generation of query Q can then be expressed as:

$$p(q|d) = \prod_{n=1}^{N} \left[\alpha P(t_n|\theta_e) + (1-\alpha) \sum_{f \in doc} P(t_n|f) P(f|\theta_d) \right] \tag{2.1}$$

where f is a non-English term and α enables smoothing with a unigram language model. The probability of generating non-English terms f from state θ_d can be estimated from counts

$$P(f|\theta_d) = \frac{c(f, doc)}{\sum_{f'} c(f', doc)} \tag{2.2}$$

$P(f|\theta_d)$ can be viewed as the ratio of the expected term count of the English term and the length of the document. The probability of generating English terms $t_n^{(e)}$ from state θ_e is similarly estimated from counts in a large corpus of English (the Google one billion word corpus [34]).

We estimate the translation probabilities $P(t_n|f)$ from the parallel corpus, as described previously. The indexing/query latency of PSQ is influenced by the number of translation alternatives to choose from for a specific document term. To manage the translation alternatives, common approaches include truncating the list of translations, using a Cumulative Distribution Function (CDF) threshold to clip the tail, or using a Probability Mass Function (PMF) threshold to discard probabilities that are less than a specific value. In each relevant chapter of this dissertation, we describe the method used to estimate the translation probability and the design choices made to limit the translation alternatives.

2.4 Neural IR

Recently, neural approaches to building ranking models have been gaining a lot of traction due to key innovations in attention-based neural architectures, coupled with large-scale relevance-annotated collections becoming available for training. The building blocks of these ranking models include: (1) some way of creating dense vector representations or embeddings in which terms with similar meaning are represented by similar embeddings, (2) the learned attention between the query and the document term representations encoded by the neural architecture, and (3) deep neural architectures that can be trained using a task-specific training loss.

2.4.1 Cross-lingual Embeddings

The initial work on creating dense vector representations for terms dates to the introduction of Latent Semantic Indexing (LSI) [61]. Specifically, a truncated Singular Value Decomposition (SVD) was used to compute a dense representation for each term (known as a singular vector). While initially created for monolingual texts, this approach was later extended to create bilingual embeddings that assigned similar dense representations to terms with similar meanings, regardless of their language [109]. In recent times, neural autoencoders that use non-linear models are used to compute a dense representation for each term [20, 183]. These dense representations capture an embedding of the original high-dimensional term space in a lower-dimensional vector space.

Three broad classes of techniques have been proposed for creating bilingual term embeddings: (1) projection-based [54, 97, 123, 192], (2) pseudo-bilingual [5, 66, 183], and

(3) unsupervised [6, 39, 75]. One problem with these learned embeddings is that each instance of a term is assigned the same representation irrespective of the context surrounding the term (e.g., the representation of the term "bank" for a river bank and for a financial bank would be the same). The advent of neural architectures made it possible to produce contextual term embeddings, with the embedding for *bank* differing, depending on its surrounding terms [48, 146] or the context. The resulting Bidirectional Encoder Representations from Transformers (BERT) architecture [48] has since been further extended (e.g., by RoBERTa [110], XLNet [200], ELECTRA [38] and XLM [96]), some of which support more than 100 languages. These encoders have become a de-facto standard for initializing neural models for ranking texts.

2.4.2 Interaction-based Neural Models

One of the ways of building effective neural models, commonly explored in monolingual IR, is to use embeddings as input to neural architectures that are optimized for relevance and that perform *full-collection neural ranking*. Specifically, query and document terms are encoded using a neural model initialized with low-dimensional term representations (e.g., Word2Vec [124]), and the model learns interactions between those representations to maximize a relevance objective (e.g., DRMM[68], KNRM[193] or PACRR[76]). This line of work is referred to as "interaction-based" due to the ability of models to leverage the interactions between the query and document terms.

The work of Yu and Allan [202] extends these interaction-based retrieval approaches to CLIR. The matching models are initialized with fastText embeddings aligned using a

dictionary [84], which is an example of the projection-based bilingual word embedding technique mentioned in the previous section. Subsequent research in monolingual retrieval has focused on switching from using context-independent Word2Vec embeddings to initialize the neural ranking models using contextual BERT-based representations mentioned above. These contextual Transformer-based architectures have achieved results superior to the best previously known techniques for ranking documents with respect to a query in monolingual applications [89, 103, 114].

Ranking models that leverage the interaction between each query term and every document term also referred to as cross-encoder, can be computationally expensive when ranking all of the documents in a large collection. This leads to cascade-based approaches, which involve running an efficient recall-oriented system first to get an initial set of documents, which is then re-ranked using a more computationally expensive model [44, 134, 201]. Zhang et al. [206] extended the cascade re-ranking approach to the cross-language setting and found similar improvements as in monolingual retrieval, especially in low-resource languages. Jiang et al. [81] fine-tuned a pretrained multilingual BERT model on cross-language query-sentence pairs constructed from parallel corpora in a weakly supervised fashion and applied this model to perform re-ranking for CLIR. Shi and Lin [167] used a transfer learning approach by applying a retrieval model trained on a large collection in English to retrieve content in other languages. However, the recall of these cascade re-ranking approaches is dependent on the quality of the initial set of documents produced by the first-stage systems. In the case of CLIR, the vocabulary mismatch between the queries and the documents adds additional challenges.

2.4.3 Representation-based neural models

In the case of representation-based neural models, the queries and documents are encoded separately into a shared vector space using a model commonly known as a dual encoder or a bi-encoder. Here, we represent the query encoder as η_q and the document encoder as η_d, though they could be the same ($\eta_q = \eta_d$). The matching is performed using some form of similarity function ϕ computed from the encoded query $\eta_q(q)$ and $\eta_d(d)$ document representation. The estimated relevance score of the document is computed as

$$\text{Rel(q,d)} = \phi(\eta_q(q), \eta_d(d)) \tag{2.3}$$

There are two main families of bi-encoder models that differ based on the encoded query/document representation: 1) single-representation and 2) multi-representation. In single-representation models, the query and document encoders encode the entire query or document to generate a single representation. These models have the advantage of reduced storage requirements due to using a single representation for each document, but the compression of the entire document to a single vector usually leads to loss of information. To avoid this, multi-representation models generate multiple representations of queries or documents or both either by generating a fixed amount of vectors (e.g., poly encoders [77] with a codebook of size k) or using the representation of every query and document term (e.g., ColBERT [89]).

A bi-encoder model can be trained in an end-to-end manner to optimize the ranking effectiveness. The document representations can be pre-computed using the trained model

and stored in an index that supports fast nearest-neighbor search operations. At query time, the queries are encoded using the trained bi-encoder, and the documents that are nearest to the encoded query are retrieved from the pre-computed index using approximate nearest neighbor techniques.

2.4.4 Training losses

Learning-to-rank approaches to IR preceded the current wave of neural ranking models. This framework introduced three main types of approaches to building and training models that vary based on the type of inputs: 1) pointwise, 2) pairwise, and 3) listwise.

Pointwise approaches take as input a single <query, document> pair to generate a score for the input document. The model could be a supervised classifier [101, 132] that predicts whether the document is relevant or not or a regression model [42, 166] that assigns a score to the document. The final ranked list is constructed by sorting the scores assigned by the model to each document.

Pairwise approaches, on the other hand, take a pair of documents for a given query and learn to assign scores such that the relevant document is ranked higher than the non-relevant document. The pairwise model learns to output a preference order among the documents, which can be used to build a ranked list with greedy approaches. Some examples of pairwise approaches include RankNet [30], RankBoost [59], RankingSVM [72, 82], LambdaRank [31], LambdaMART [188]

Listwise approaches operate on a ranked list of documents for a query with the goal of producing an optimal ordering of a list. Compared to the other two approaches, this ap-

proach closely models the actual ranking problem. It also allows the model to approximate existing IR measures, such as MAP to evaluate ranking effectiveness for the optimization task. Since these measures are non-differentiable, smooth continuous versions of these measures have been proposed as a basis for optimization. Some examples of pairwise approaches include SoftRank [175], SVM^{map} [203], AdaRank [196], ListNet [33], ListMLE [191].

In this dissertation, we focus on Pointwise loss to build neural ranking models.

2.5 Fusion techniques

There are diverse approaches to building IR systems, which provide the opportunity to improve the search results by combining the constituent systems. CLIR systems have even greater potential diversity than monolingual IR systems because of the additional potential for diversity that translation resources, and ways of using those translation resources, introduce. Combining multiple sources of evidence, more generally known as fusion [189], can be done in two ways, *early* fusion (in which evidence from multiple sources is combined by one or more components of the full system before results are generated) or *late* fusion, in which the ranked lists from separate IR systems are combined. Late fusion is often referred to as a system combination, and in this dissertation, we refer to the systems combined using late fusion as an ensemble. In general, system combination can be employed with systems that search different document collections, but here we focus only on cases in which all systems search the same collection.

System combination for CLIR has a long history, beginning when McCarley [119] found that neither query translation nor document translation was a clear winner and

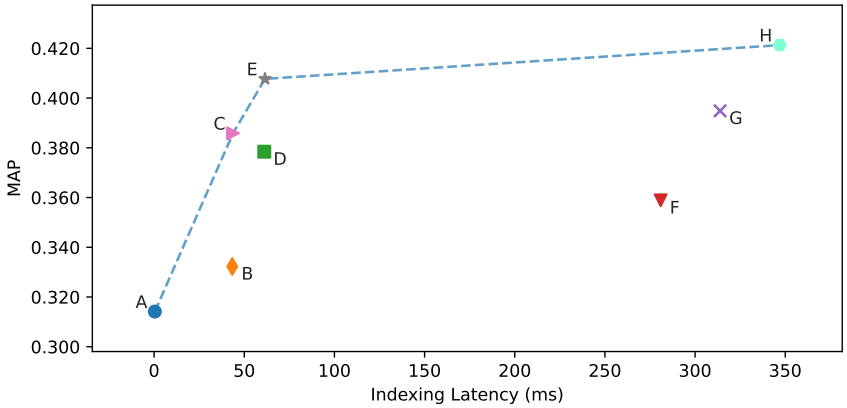


Figure 2.1: Figure illustrating the Pareto-frontier of Indexing Latency and MAP using dashed lines. Systems A, C, E, and H are Pareto-optimal.

that a late fusion combination between the two approaches yielded the best results. Later work [27, 91] further supported this claim by demonstrating the benefits of combining more diverse systems as well.

Early fusion of translation resources has also proven to be successful in CLIR. Perhaps the best-known case is the fusion of parallel sentences with translation pairs from a bilingual lexicon when training machine translation systems [195]. The architecture for this is simple; machine translation systems are trained on pairs of translation-equivalent sentences, and term pairs from a bilingual lexicon are simply treated as very short sentences. Because the term distribution in bilingual lexicons is not as sharply skewed in favor of common terms as is naturally occurring parallel text, this approach can help to reliably learn translations of relatively rare terms. This is also facilitated by the availability of bilingual lexicons in several languages, including Panlex [86] and MUSE [39].

2.6 Pareto-optimality

Pareto-optimality is a concept that originated in economics and has since been widely used in several fields. In this dissertation, we utilize the concept of Pareto-optimality to analyze the tradeoff between the effectiveness and the efficiency of different CLIR systems. A CLIR system is considered Pareto-optimal if no other system can offer better effectiveness without sacrificing efficiency. In situations where multiple CLIR systems are Pareto-optimal, we refer to them collectively as the Pareto-frontier. Depending on the specific application requirements, practitioners can choose among the different Pareto-optimal systems on the Pareto-frontier to achieve the best balance between effectiveness and efficiency. Figure 2.1 shows an illustration of the Pareto-frontier of efficiency during the indexing phase, measured using indexing latency, and the retrieval effectiveness, measured using MAP. The optimal outcome would be in the upper left corner of the figure, where the system achieves a low indexing latency and a high MAP score.

Chapter 3: Building Effective & Efficient Cross-Encoders for CLIR[1]

The effectiveness of neural ranking models, which compute the full interaction between query and document term embeddings (also known as interaction-based ranking models), depends on both the quality of the underlying term embeddings and the method used to compute the interactions. A specific type of interaction-based model, the Cross-Encoder, generates superior contextual representations by computing joint interactions between concatenated queries and documents through multiple layers of self-attention [180] from the transformer layers of pretrained language models (PLMs). However, this expressiveness comes at a higher computational cost due to the quadratic time and space complexity of self-attention in the transformer layers of PLMs. This has two implications.

First, the time complexity to rank documents in response to a given query increases linearly for a pointwise Cross-Encoder and quadratically for a pairwise Cross-Encoder as the size of the document collection grows. Therefore, retrieve-and-rerank pipelines have become widely adopted in monolingual retrieval applications to overcome this limitation. In these pipelines, the first stage involves using an efficient retrieval system, such as BM25 [155], to generate a set of top-k documents relevant to a query (often set to 1000). The second stage is a reranking process, in which a Cross-Encoder serves as a neural reranking model

[1]This chapter contains content from: **Suraj Nair**, Petra Galuščáková, Douglas Oard, Le Zhang, Dami-anos Karakos, and Bonan Min. "Rationale Training based Neural Re-ranking for Ad-hoc CLIR." In Preparation. [129]

to reorder the documents returned by the first-stage retrieval system. Training the Cross-Encoder with PLMs like BERT [48] or RoBERTa [110] has improved effectiveness beyond what traditional retrieval methods alone can achieve. In this chapter, we develop retrieve-and-rerank pipelines for ad-hoc document ranking in CLIR.

Second, BERT-style models are limited in their ability to handle long documents, resulting in a maximum context length that is often shorter than the length of a document. Given the widely used definition of relevance that a document is considered relevant if any part of that document is relevant, passage retrieval offers an elegant way of working around this limitation. To address this issue during the querying phase, two popular approaches consisting of FirstP and MaxP [44] include truncating documents to the first passage of n terms or creating a set of overlapping passages/sentences with a fixed length and stride, respectively. The final document score is either the score of the first passage (in the case of FirstP) or the highest passage score among the set of overlapping passages in the case of MaxP. However, this approach fails to leverage the text signals in the underlying passages and instead relies on a static choice between one or all passages for scoring, which represents two extremes of a broad spectrum of passage selection. Different passage selection strategies create a tradeoff between reducing query latency by limiting the number of passages to be scored by cross-encoder and improving retrieval effectiveness. Our goal in this chapter is to explore strategies that offer a Pareto-optimal balance between these contrasting objectives.

When using any passage selection strategy during the querying phase, a critical question arises about how to fine-tune the Cross-Encoder model on document-based training collections with relevance judgments during the training phase. If we knew which passage from the corresponding document in the training set should receive the highest score, it

would be a simple matter of transforming document retrieval to passage retrieval by replacing each document with its best passage and then fine-tuning on that dataset. Motivated by this idea, we introduce a simple yet effective approach called CREPE (Cross-language REtrived PassagE) for CLIR. This approach utilizes an efficient first-stage CLIR system to score individual passages from a document in response to a query and selects the highest-scoring passage or *CREPE* to create training samples for fine-tuning a pointwise cross-encoder. During the querying phase, using CREPEs can be regarded as a passage selection strategy that identifies the best passages to be scored by the cross-encoder. By doing so, this approach offers a solution to balance query latency with retrieval effectiveness.

The remaining sections of this chapter are structured as follows. First, we provide an overview of the general setup of the two stages of the retrieve-and-rerank CLIR pipeline in Section 3.1. Next, we introduce the key methodology of CREPE and its application during the training and querying phases in Section 3.2. We describe the experimental setup in Section 3.3 and then analyze the effect of CREPE in the querying and training phases in Sections 3.4 and 3.5, respectively. Finally, we conduct a detailed analysis with ablation studies in Section 3.6 and conclude the chapter with a summary in Section 3.7.

3.1 Retrieve-and-Rerank CLIR pipeline

In this section, we provide a general description of the two stages of the retrieve-and-rerank pipelines. We start by discussing the first-stage CLIR systems we use, followed by a description of the reranking setup with cross-encoders.

3.1.1 First-Stage CLIR systems

Previous work [20, 81, 202, 206, 207] involving the retrieve-and-rerank pipelines in CLIR mostly translate queries and perform matching in the document language [167, 168]. However, the CLIR problem presents various design choices for first-stage retrieval systems due to the distinct query and document languages. Matching term meaning across languages is a requirement for CLIR systems to bridge the vocabulary gap between queries and documents. Previous chapters (1 and 2) have introduced several CLIR systems that tackle this issue, such as utilizing MT or translation probabilities from the statistical alignment of parallel text. For the retrieve-and-rerank CLIR pipelines, we employ three different first-stage retrieval systems that differ based on the input format of queries and documents:

1. Probabilistic Structured Queries - This system utilizes queries and documents in their original form without using any off-the-shelf tool or machine translation system to translate them. We use a well-known approach in CLIR, Probabilistic Structured Queries (PSQ) [47], to estimate term counts for a query q in document d, using the translation probabilities of a query word given the document terms. Specifically, we use a PSQ-based HMM model (PSQ-HMM) as described in Section 2.3.1 to estimate the relevance of a document in a target language given an input query.

2. Query Translation - This system uses queries translated to the document language, and documents are in their original form. The first-stage retrieval is performed using a BM25 system in the document language.

3. Document Translation - This system uses queries in their original form and documents

translated to the query language. The first-stage retrieval is performed using BM25 in the query language.

Aside from being part of the retrieve-and-rerank pipeline, these first-stage CLIR systems serve as the model used to generate "CREPEs," which we will introduce in Section 3.2.

3.1.2 CLIR Reranker

We describe our approach to the reranking task by building on prior work [44, 134]. We adopt the same strategy of framing the task as a binary classification problem, in which our reranking model, initialized with a pre-trained language model known as a cross-encoder, predicts the relevance of a document D to a query Q. This formulation casts the cross-encoder in a pointwise setting, scoring each document independently for a given query. However, due to the maximum sequence length restriction of PLMs (e.g., 512 tokens), it is challenging to handle long documents. Existing works have addressed this issue by segmenting documents into either sentences [3], or overlapping passages [44] or using hierarchical models [100, 113] to build a document-level representation from its constituent passages. In this chapter, we follow the MaxP approach [44] and segment documents into overlapping passages with a window size of 150 words and a stride of 75 words.

We build separate neural rerankers for the three different first-stage CLIR systems introduced in Section 3.1.1. Figure 3.1 shows a pointwise cross-encoder model for the first-stage PSQ system, initialized with an mBERT encoder. The model takes as input the concatenation of the original query Q and passage P tokens, including [CLS] and [SEP]

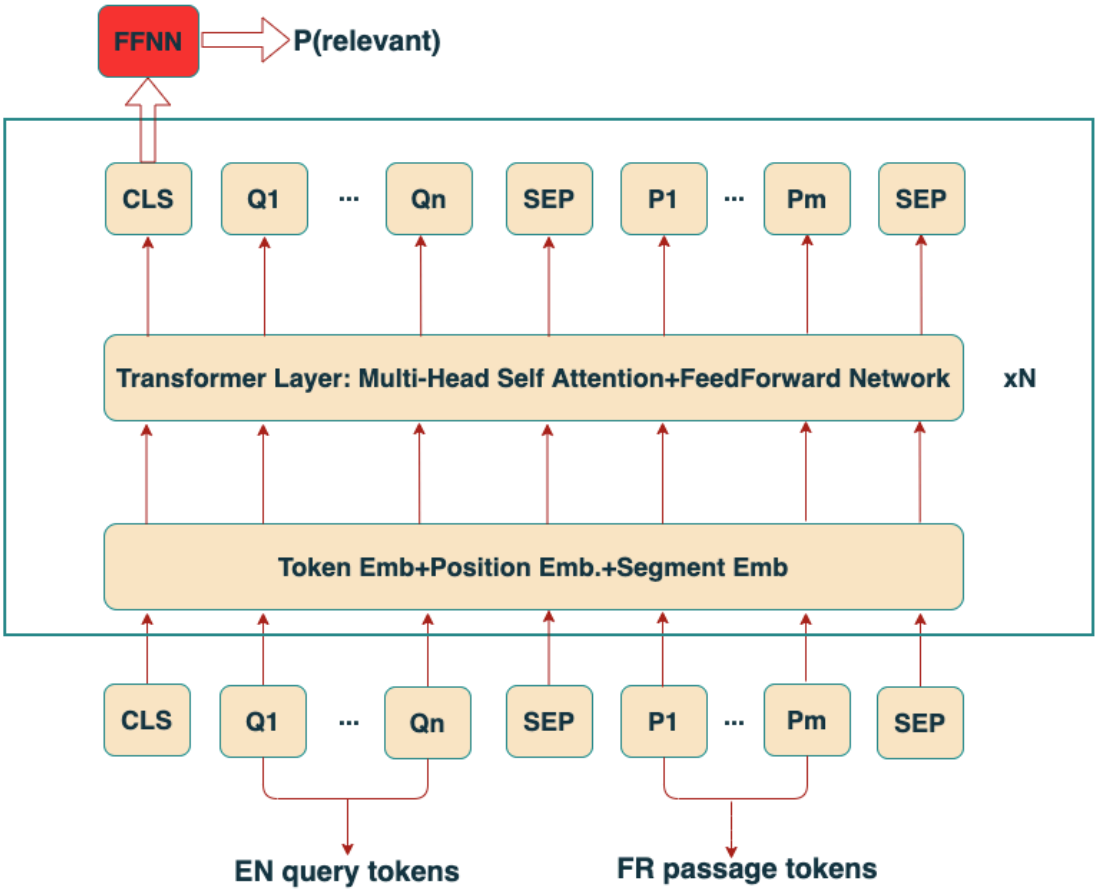


Figure 3.1: mBERT pointwise cross-encoder; English query, French passage.

tokens, as follows: [[CLS] Q [SEP] P [SEP]]. These tokens pass through multiple layers of transformers [180], with each layer producing fine-grained contextualized word representations. Finally, the [CLS] token output from the last layer is fed into a single-layer feed-forward neural network (FFNN) whose outputs are softmaxed to generate the probability of passage P being relevant to query Q. The reranking setup is the same for query translation and for document translation, except we feed the cross-encoder either a translated query Q' or translated passage P', respectively. For the first-stage PSQ and Query Translation CLIR systems, we utilize multilingual pretrained language models to initialize the cross-encoder. In contrast, for the Document Translation CLIR system, we employ a monolingual pretrained language model to initialize the cross-encoder. The cross-encoder is subsequently fine-tuned on the CLIR training collection generated by the respective first-stage CLIR system, as discussed in the subsequent section.

3.2 Cross-Language Retrieved Passages (CREPE)

In this section, we first present our CREPE approach to creating the training dataset for fine-tuning the cross-encoder for CLIR. We later introduce how CREPEs from the first-stage CLIR systems can be used in the querying phase as an alternative to the MaxP approach.

3.2.1 Training Phase

To effectively train a cross-encoder model that can accurately rerank passages, it is crucial to have supervision in the form of relevance judgments at the passage-level.

However, in practice, relevance judgments are typically only available at the document-level. Consequently, generating passage-level judgments from document-level judgments is a challenging task. This problem falls under the category of supervised machine learning, known as multiple-instance learning. In this type of learning, the model is presented with a set of labeled bags or documents, and each bag consists of a set of unlabeled instances or passages. A naive approach to address this problem is to assume that all passages from relevant documents are relevant and vice versa for non-relevant documents. However, this assumption is too strong, especially for relevant documents, as not all passages within a relevant document are necessarily relevant. To overcome this issue, a semi-random sampling approach was introduced by Dai and Callan [44], which involves selecting the first passage from every document, followed by randomly selecting 10% of the remaining passages.[2] The reasoning behind selecting the initial passage is that certain genres, such as news articles that typically follow an inverted pyramid style of writing [58], tend to focus on the main topic early on in the text, as observed by Wu et al. [190]. This approach to creating a dataset is referred to as "stochastic."

Although the stochastic approach may be a reasonable method for relevant documents, the passages sampled in this way from judged non-relevant documents may not provide the most discriminative signal. Therefore, we propose a new method called *CREPE*. CREPE addresses the problem of multiple-instance learning by creating a passage-level training dataset that selects the most discriminative passage from the document using signals from our first-stage CLIR system. Specifically, for a given query q_i and document d_j

[2]This detail is not mentioned in the original paper [44] but can be found in the reference implementation: `https://github.com/AdeDZY/SIGIR19-BERT-IR/blob/master/run_qe_classifier.py#L468-L471`

Algorithm 1 Dataset creation using CREPE

Input: Q: queries, C: document collection, R: binary relevance judgments, FS: first-stage CLIR system

$S \rightarrow \varnothing$
for $q_i \in$ Q **do**
  D = FS(Q, C)
  for $d_j \in$ D **do**
    P = overlapping_passages(d_j)
    if $R_{ij} == 1$ **or** $R_{ij} == 0$ **then**
      $p_j^k = \max_k$ FS(Q,P)
      $S = S \cup ((q_i, p_j^k), R_{ij}))$
    else {Unjudged document}
      p_j^k = random(P)
      $S = S \cup ((q_i, p_j^k), 0))$
    end if
  end for
end for

returned by the first-stage CLIR system FS, we rank all passages P in document d_j using the same system and select the passage p_j^k with the highest relevance score for training. This leads to two things:

- For a relevant document, we select the passage that contributes the most to the relevance of that document
- For a judged non-relevant document, we select the passage that has the most lexical overlap with the query, effectively choosing a *hard-negative* sample

We treat unjudged documents as non-relevant, and in such cases, we select a passage randomly from the document. The pseudo-code for creating the dataset using the CREPE approach is shown in Algorithm 1.

The motivation behind CREPE is twofold. First, we aim to use the best-scoring passage from every relevant document as a *strong-positive* passage to train the model. Second, we want to select the most discriminative passage from judged non-relevant documents as

a *hard-negative* passage. The closest approach to ours is by Rudra and Anand [159], which used a fine-tuned BERT to score passages and used a score-based threshold to determine which passages to select. However, our method differs in that we use efficient first-stage CLIR systems to rank passages instead of using BERT to score all passages, which can be computationally expensive. While their approach focuses on selecting the best positive samples, our approach of choosing hard negative samples from judged non-relevant documents helps to improve overall effectiveness. Furthermore, our experiments reveal that fine-tuning a cross-encoder using a dataset created using the CREPE approach outperforms the stochastic sampling of the first passage (which is often a strong baseline for news stories), even in more challenging scenarios, such as retrieval across languages.

3.2.2 Querying Phase

In the querying phase, the first-stage CLIR system returns documents that are segmented into overlapping passages.[3] A score is computed for each (query, passage) pair using the fine-tuned cross-encoder. MaxP aggregates the scores by selecting the highest score among the passages as the corresponding document score and uses it to rank the documents. On the other hand, FirstP chooses the score for the first passage as the corresponding document score. By default, MaxP requires the cross-encoder to score all the passages from the top-k documents returned by the first-stage CLIR system. However, since applying BERT-based cross-encoders is expensive, the reranking depth is typically adjusted to balance effectiveness and efficiency. We explore the use of CREPEs during the

[3]While directly retrieving passages could be an option, our initial experiments show it has lower effectiveness than retrieving documents from a first-stage system. Adjusting CLIR system hyperparameters to account for shorter passage lengths may be necessary.

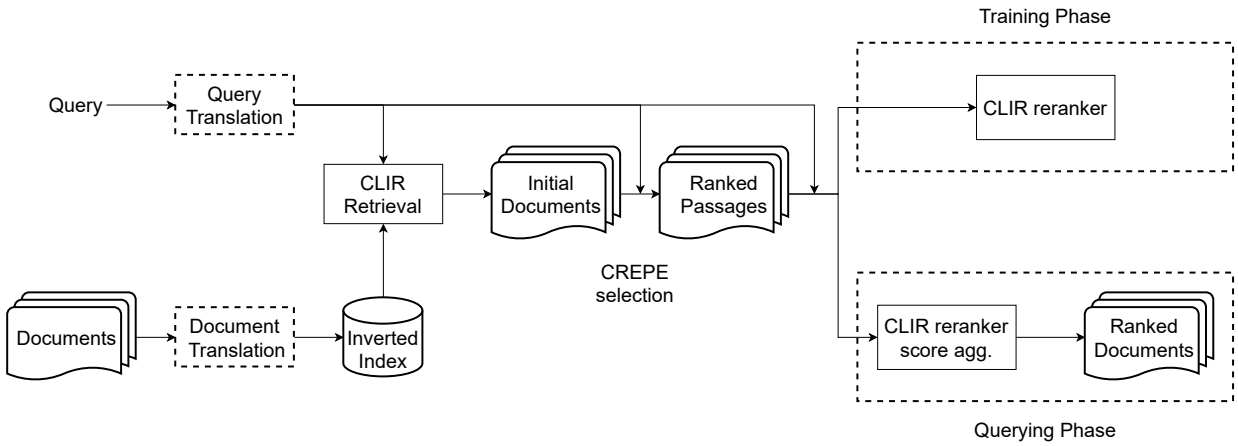


Figure 3.2: Retrieve-and-rerank CLIR pipeline with CREPE-based strategy

querying phase to reduce the number of passages to be scored by the cross-encoder and improve query latency while maintaining retrieval effectiveness. Specifically, instead of selecting all passages, we sub-select passages by choosing only the top-m passages indicated by the first-stage CLIR system. The passages coming from the first-stage CLIR system are referred to as CREPEs. We can also combine CREPEs with systematic passage selection strategies, such as FirstP, to create hybrid passage selection strategies. In Section 3.4, we analyze the impact of various passage selection strategies on the tradeoff between retrieval effectiveness and query latency. Figure 3.2 shows the overall retrieve-and-rerank pipeline using the CREPE approach.

3.3 Experiments

In this section, we provide details on the test collections, training data, and training setup of our retrieval and reranking models.

Test Collections. We evaluate our methods using CLIR test collections for six language pairs. The queries are in English (EN), and the documents are in Spanish (ES),

Italian (IT), Dutch (NL), Finnish (FI), German (DE), or French (FR) from CLEF evaluation campaigns [145]. The documents are news articles, and we use the collections from the CLEF's multilingual ad-hoc retrieval track, pooling the topics and judgments across the languages from the years 2000-2003 [24, 25, 26, 28]. The CLIR test collection statistics are presented in Table 3.1. We experiment with the title as queries, which resembles a typical web search query.

Text Preprocessing. We tokenize all text, including queries, documents, and translation resources, using Moses [94] and normalize it by converting all text to lowercase, removing punctuation, stripping diacritics from characters, and removing non-printable characters. We also remove stopwords from queries and documents using the NLTK [19] toolkit. This preprocessing is applied only for the first-stage retrieval systems. For reranking, we use the raw queries and documents as input and rely on the tokenizer included as part of the pretrained model.

3.3.1 First-stage retrieval setup

We provide details on the first-stage retrieval setup for different CLIR pipelines.

Query Translation (QT). As a baseline, translating the English queries to the target language is first done using Google Translate, and the translated queries are then used to perform monolingual retrieval in the target language. Specifically, we use the Anserini [199] toolkit to index the documents using Lucene's language-specific analyzer.[4] For retrieval, we use the BM25 model [155] with default hyperparameters (k1=0.9, b=0.4) from Anserini. We refer to this system as QT-BM25.

[4]We extended Anserini v0.10.1 to add the support for FI, NL, & IT

Document translation (DT). For translating documents to the query language (English), we use Opus-MT [178] models available as part of the EasyNMT[5] toolkit. Specifically, we use the xx-en models pretrained on OPUS [177] data using the MarianNMT [85] model. Here xx stands for the document language. Retrieval is performed using Anserini's BM25 model with default hyperparameters. We refer to this system as DT-BM25.

PSQ-HMM. To obtain the translation probabilities to be used in PSQ-HMM, we rely on the word alignment output from the GIZA++ [137] aligner. For training GIZA++, we use a combination of parallel sentences from Europarl [92] and Panlex [86] dictionaries for CLEF languages. For each language pair, we have approximately 2.5-3 million sentence pairs for training. We train the model for five iterations each for Model 1, HMM model, Model 3, and Model 4 in both language directions. Finally, we apply the grow-diag-final-and [93] heuristic to combine the forward and backward alignments and use it to generate the translation probabilities. Translation probabilities that are less than 1×10^{-5} are filtered out. It has been shown that selecting a single translation term often leads to reduced CLIR effectiveness [20, 202], instead, in this work, we use multiple translation alternatives. By instead using a broad range of translations, the recall of our first-stage retrieval system might be improved. The value of α is set to 0.1 in our experiments.

3.3.2 CLIR reranker setup

We investigate the use of different multilingual BERT-style models to initialize the cross-encoder for PSQ and Query Translation pipelines. The models we explored are briefly described below:

[5] `https://github.com/UKPLab/EasyNMT`

Table 3.1: CLIR test collection statistics: number of EN queries (#query), number of target documents (#docs), average number of relevant documents per query (#rel), average length of EN title queries (qlen), average length of target documents (dlen)

Target	ES	IT	NL	FI	DE	FR
#query	160	200	160	90	200	200
#docs	454,045	157,558	190,604	55,344	294,809	129,806
#rel	49.5	17.3	29.1	10.9	33.6	20.4
qlen	3.4	3.5	3.4	3.6	3.5	3.5
dlen	328.6	276.5	371.9	256	260.2	295.5

- **mBERT** [48] is built on top of the BERT-base architecture that includes several layers of transformers [180]. It is pretrained on concatenated Wikipedia texts in over 100 languages with the Multilingual Masked Language Model (MMLM) task.

- **XLM-R** [40] is trained on the larger CommonCrawl corpus [185] with the MMLM task. We use the large version of XLM-R, which has 2x more layers (24) as compared to mBERT (12) and 2x more vocabulary size.

For the Document Translation pipeline, we use monolingual BERT-style models to initialize the cross-encoder.

- **BERT** [48] is pre-trained on Wikipedia & BookCorpus [208] with Masked Language Modeling and Next Sentence Prediction tasks.

- **ELECTRA** [38] is trained on the same corpus and hyperparameters as BERT, however, using a Replaced Token Detection task. We use the large version of ELECTRA, which has 2x more layers (24) as compared to the BERT-base (12).

Rather than directly initializing the cross-encoder with off-the-shelf PLMs, we use publicly available BERT-style models [100], such as mBERT,[6] BERT-MS,[7] and ELEC-

[6] `https://huggingface.co/amberoad/bert-multilingual-passage-reranking-msmarco`
[7] `https://huggingface.co/Capreolus/bert-base-msmarco`

TRA,[8] which are fine-tuned on the MS MARCO passage retrieval task [11]. Since we do not have a version of the XLM-R model fine-tuned on MS MARCO, we instead use an XLM-R[9] model fine-tuned on the SQuAD question answering dataset [151].[10]

Training Setup. We rely on PyTorch [143] and Huggingface Transformers [187] to fine-tune the CLIR cross-encoder. For each target collection, we use the top-1000 documents from the first-stage retrieval system to create a passage dataset using either stochastic or CREPE approach as described in Section 3.2.1. For CREPE, we set m to 1, using a single passage from each document returned by the first-stage CLIR system. The input to the cross-encoder consists of query tokens concatenated with the passage tokens. To be consistent, we use the same query representation, title, that is used in the first-stage CLIR system for the cross-encoder. We create the passage representation by concatenating the title of the document with the passage content. We use the default hyperparameters from MaxP as follows: All the cross-encoder models use a fixed input size of 256 tokens. The models are trained with Adam optimizer [90] with weight decay set to 0.01 using Cross-Entropy loss. We set the learning rate to 1×10^{-5} for all the models except XLM-R, where we use 5×10^{-6} as the learning rate. We set the linear warmup rate to 10% of the initial training steps and train the model using 16-bit precision for a single epoch with a batch size of 32 training instances.

[8] `https://huggingface.co/Capreolus/electra-base-msmarco`

[9] `https://huggingface.co/deepset/xlm-roberta-large-squad2`

[10] The version of the XLM-R model fine-tuned on MS MARCO was not available at the time of the experiments.

3.3.3 Baselines

We report our three first-stage retrieval systems (PSQ-HMM, QT-BM25, DT-BM25) as baselines for our reranking pipelines, and we report the following as additional baselines:

- Human Translation (**HT**) - we use the document-language queries[11] provided as part of the test collection to perform monolingual retrieval in the target language. We use Anserini's BM25 model with default hyperparameters to do monolingual retrieval.
- Deep Relevance Matching Model (**DRMM**) [68] - This is a pre-BERT neural matching model that learns patterns from the interaction between the query terms and document terms. We do not reimplement but rather report results from Yu and Allan [202]. The DRMM model is initialized with non-contextual cross-lingual word embeddings from fasttext [84].

For each cross-encoder, we report two results, the first for a cross-encoder fine-tuned on the dataset created using the stochastic approach from the MaxP framework (as a strong neural reranking baseline) and the second (following a slash) fine-tuned on the dataset created using our CREPE approach.

3.3.4 Evaluation

To evaluate the effectiveness of our CLIR model, we follow a 5-fold cross-validation setup for each CLIR test collection. For each document collection, we split the queries into

[11] In CLEF, the generation of original queries was divided across languages, so in some cases, these are original queries; in other cases, they are the result of reexpression of those queries in other languages by human translators who were instructed to express the query in a form that would be natural in that language.

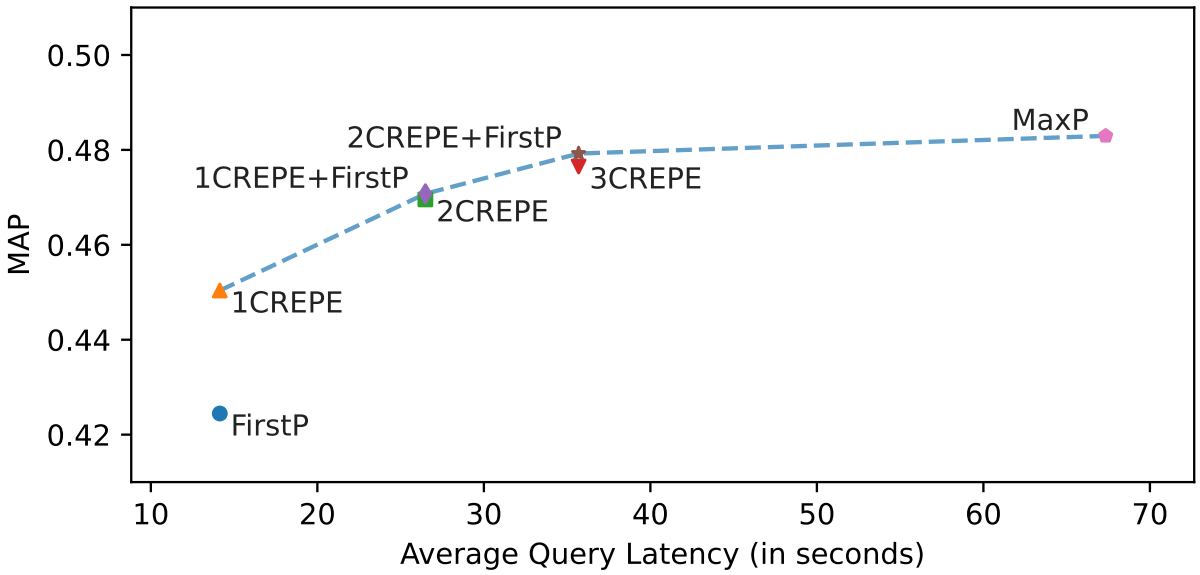


Figure 3.3: Average Query Latency (in seconds) vs. MAP for different passage selection strategies using PSQ-HMM system with XLM-R cross-encoder macro-averaged across the six CLEF collections. The number preceding CREPE denotes the number of passage(s) selected by the PSQ-HMM system. The dashed line indicates the Pareto frontier of retrieval effectiveness and query latency.

five disjoint folds, train the reranker on four-folds, and apply the model on the remaining fold. We re-rank the top 1000 documents returned by the first-stage retrieval system. We report Mean Average Precision (MAP) as the evaluation measure using the trec_eval[12] toolkit. Differences in the means are tested for significance using a two-tailed paired t-test ($p < 0.05$) with Bonferroni correction.

3.4 Effect of CREPE at Querying Phase

Figure 3.3 illustrates how different passage selection strategies used during the querying phase affect the average results across six CLEF collections. The analysis is conducted using the PSQ pipeline, where PSQ-HMM is the first-stage CLIR system, and a fine-tuned

[12] `https://github.com/usnistgov/trec_eval`

XLM-R cross-encoder is the reranker. Each strategy creates a tradeoff between the number of <query, passage> pairs that need to be scored by the cross-encoder, which affects query latency, and the effectiveness of the ranked list of documents, which is measured using MAP. Query latency for the XLM-R cross-encoder is calculated based on the concatenated query and passage text sequences of length 256, run on a single V100 GPU with a batch size of 1024. It's worth noting that the query latencies for the XLM-R large cross-encoder are too extensive for interactive applications. In practice, a distilled version of the cross-encoder is employed for interactive purposes.

The Pareto frontier of retrieval effectiveness and query latency is depicted in Figure 3.3. MaxP produces the highest MAP score but at the expense of the highest query latency, as it calculates scores over all the passages from a document. FirstP, which is widely used due to its simplicity, has the least query latency and only requires scoring one passage per document. However, it has the lowest MAP score among all the strategies.

When comparing FirstP with the top passage selected by PSQ-HMM (denoted as 1CREPE), we observe a 5% relative increase in MAP for the same cross-encoder query latency between the two strategies. However, more importantly, we establish that the FirstP strategy is not Pareto-optimal, and we can switch to 1CREPE, which is Pareto-optimal. 1CREPE can achieve 93% of the effectiveness of MaxP while requiring only a fraction of the passages to be scored by the cross-encoder, leading to an average reduction of 4.8x in query latency. Furthermore, we can enhance the retrieval effectiveness by adding more passages. Using the top-2 (2CREPE) and top-3 (3CREPE) passages can achieve about 97% and 99% of the effectiveness of MaxP, respectively. It is worth noting that 2CREPE and 3CREPE require only a fraction of the MaxP query latency, 0.4x and 0.55x,

respectively, to achieve a comparable retrieval effectiveness of MaxP This can potentially enable us to increase the reranking depth without doing additional work by selecting fewer passages to score per document.

We also investigate the effectiveness of a hybrid strategy that combines FirstP with CREPE from PSQ-HMM (1CREPE and 2CREPE).[13] As expected, we find that this hybrid strategy performs almost as well as the CREPE approach, with slightly higher effectiveness for the same amount of query latency. It is worth noting that both hybrid strategies fall on the Pareto frontier of retrieval effectiveness and query latency. This emphasizes the importance of CREPE since FirstP by itself was not Pareto-optimal, and it is only after combining CREPE with FirstP that the strategy becomes Pareto-optimal.

Figure 3.4 displays a breakdown of query latency with retrieval effectiveness for each individual CLEF collection. Most of our earlier findings hold when examining individual collections. Occasionally, we observe some changes, such as in Finnish and French, where 2CREPE is Pareto-optimal instead of the hybrid strategy of 1CREPE+FirstP. In French, 3CREPE is Pareto-optimal compared to 2CREPE+FirstP. Nevertheless, the differences in MAP are minor enough not to yield a statistically significant difference.

[13] If the first passage is also selected by PSQ-HMM, we add the second highest scoring passage from PSQ-HMM.

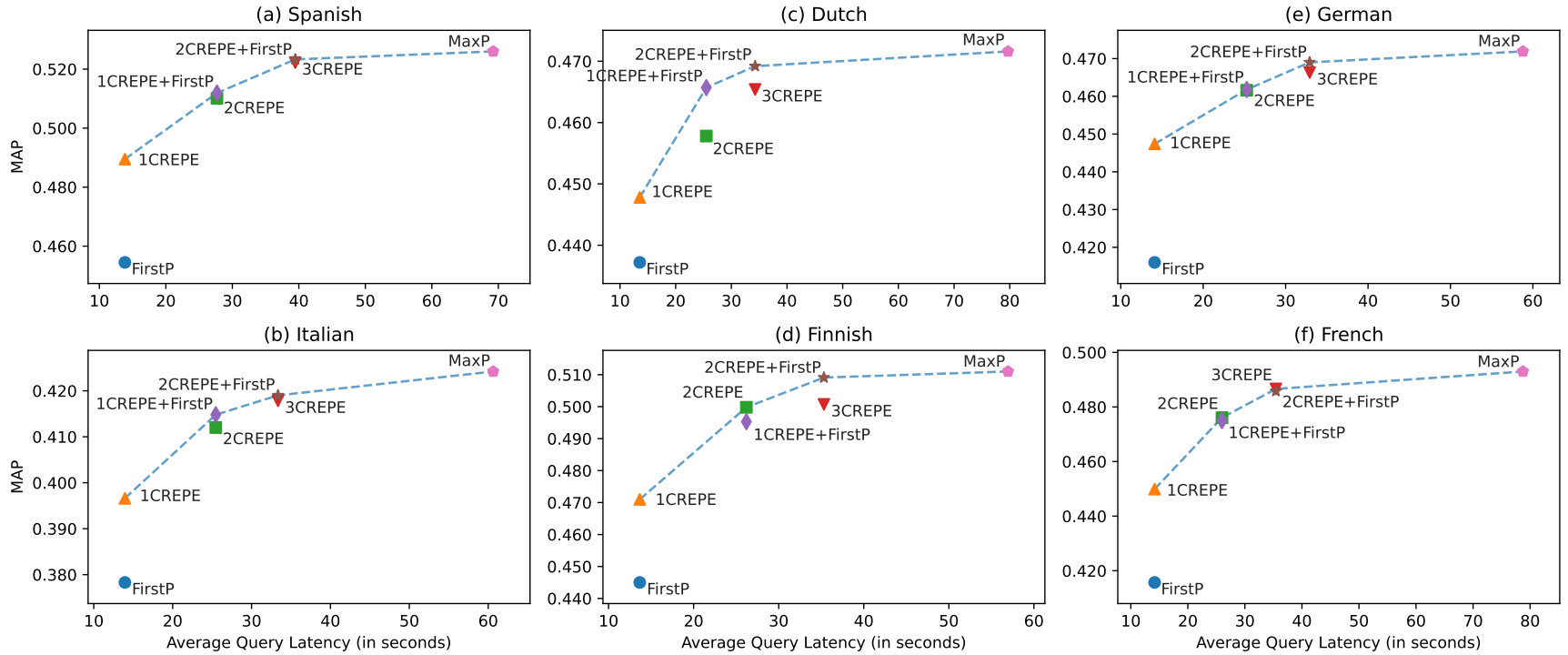


Figure 3.4: Average Query Latency (in seconds) vs. MAP for different passage selection strategies using PSQ pipeline with XLM-R cross-encoder for each CLEF collection. The number preceding CREPE denotes the number of passage(s) selected by the PSQ-HMM system. The dashed line indicates the Pareto frontier of query latency and retrieval effectiveness.

3.5 Effect of CREPE on Training Phase

Table 3.2 presents our findings comparing the performance of different CLIR models using the title field, simulating a typical web search query. We rerank the documents for each first-stage CLIR system using a cross-encoder fine-tuned on the dataset created using either the stochastic or the CREPE approach. Our primary objective is to analyze whether reranking based on CREPE enhances effectiveness over traditional first-stage methods and compare it with the stochastic approach for CLIR.

We begin by comparing the best cross-encoder, trained with CREPE, with the first-stage retrieval system and observe robust relative MAP improvements in all document collections. This holds true for the PSQ, query translation, and document translation pipelines, with 44%, 34%, and 30% relative MAP improvements averaged across all the document collections over the first-stage retrieval system, respectively.

Next, we compare CREPE with a pre-BERT neural reranker, DRMM [68]. We chose the DRMM model since it outperforms other pre-BERT rerankers, as demonstrated in Yu and Allan [202]. To have a fair comparison, we compare it to the PSQ reranking pipeline since Yu and Allan [202] used raw queries and documents with no translation involved. Our best reranking model trained with CREPE manages to outperform the DRMM reranker for the Spanish, Italian, Dutch, and Finnish document collections. The DRMM model is initialized with aligned fasttext monolingual embeddings, which is why it performs poorly in comparison to the contextualized embeddings that are a part of the pretrained models. Nevertheless, training DRMM with contextualized representations has shown to perform better than static non-contextual embeddings for monolingual retrieval in English [113].

We observe that the cross-encoder trained with CREPEs from the first-stage CLIR system typically outperforms the stronger baseline of training with the stochastic approach. Comparing the best cross-encoder within each CLIR pipeline, we consistently see MAP improvements when switching from a stochastic to a CREPE approach. This finding is particularly significant since the CREPE approach trains on a dataset that contains at most one passage from every document, whereas the stochastic strategy uses at least one passage from every document and possibly more depending on length. Thus, we showcase consistent effectiveness gains with a monolingual cross-encoder trained with CREPEs from the document translation CLIR system or a multilingual cross-encoder trained with CREPE using either PSQ or query translation CLIR system across collections in multiple languages.

Table 3.2: MAP for different CLIR retrieval models using title field and MaxP passage selection strategy for scoring. The first two rows are query translation baselines, the third row is document translation baseline. Reranking results are presented in the order of stochastic/CREPE training. The highest value in each pipeline is marked as bold. Statistically significant improvements over the first-stage system and reranker trained with the stochastic approach are marked with † and ‡ resp.

Retrieval stage	**Reranker**	**EN-ES**	**EN-IT**	**EN-NL**	**EN-FI**	**EN-DE**	**EN-FR**
HT-BM25	-	0.452	0.334	0.371	0.350	0.304	0.403
QT-BM25	-	0.420	0.304	0.324	0.342	0.275	**0.411**
DT-BM25	-	0.447	0.327	**0.408**	**0.430**	**0.375**	0.387
PSQ-HMM	-	0.402	0.281	0.352	0.321	0.313	0.362
DRMM [202]	-	**0.462**	**0.352**	0.374	0.304	-	-
PSQ Pipeline							
PSQ-HMM	mBERT	0.440/0.482†‡	0.340/0.362†‡	0.374/0.417†‡	0.379/0.414†‡	0.396/0.433†‡	0.409/0.452†‡
	XLM-R	0.496/**0.526**†‡	0.409/**0.424**†‡	0.450/**0.472**†‡	0.501/**0.511**†	0.447/**0.472**†‡	0.479/**0.493**†‡
Query Translation Pipeline							
QT-BM25	mBERT	0.467/0.496†‡	0.361/0.387†‡	0.367/0.405†‡	0.405/0.421†	0.359/0.390†‡	0.442/0.503†‡
	XLM-R	0.507/**0.525**†‡	0.417/**0.422**†	0.420/**0.428**†	0.435/**0.461**†	0.391/**0.406**†‡	0.509/**0.526**†‡
Document Translation Pipeline							
DT-BM25	BERT	0.477/0.500†‡	0.387/0.410†‡	0.454/0.487†‡	0.494/0.531†	0.4335/0.453†‡	0.458/0.486†‡
	ELECTRA	0.514/**0.530**†‡	0.417/**0.447**†‡	0.505/**0.526**†‡	0.548/**0.554**†	0.468/**0.492**†‡	0.502/**0.526**†‡

3.5.1 Comparing retrieve-and-rerank CLIR pipelines

We begin by comparing the first-stage CLIR systems. PSQ-HMM performs similarly to the QT-BM25 and DT-BM25 baselines. We attribute this to the query expansion effect in the PSQ-HMM system, which considers more than one translation alternative. In contrast, QT-BM25 uses Google Translate, and DT-BM25 uses an off-the-shelf neural MT tool to perform retrieval using the single best translation alternative. While PSQ-HMM and QT-BM25 (except for French) fall short of the human translation baseline (HT-BM25), document translation outperforms human translation in three out of six languages. We speculate that the additional context available in the documents might be aiding the NMT system to generate better translation alternatives, as opposed to the query translation performed by humans.

Next, we compare the two mPLMs, mBERT and XLM-R, used to initialize the cross-encoder in the PSQ pipeline. We find that the XLM-R cross-encoder, with its larger vocabulary and twice as many layers, consistently outperforms the mBERT cross-encoder for the PSQ and query translation pipelines across all document collections. This is consistent with the finding that the depth of the model is crucial for better cross-language generalization [88]. Additionally, for the document translation pipeline, the ELECTRA cross-encoder yields better retrieval effectiveness than the BERT cross-encoder in all document collections.

When comparing the various CLIR pipelines, we find that the document translation pipeline outperforms the PSQ and query translation pipelines in terms of CLIR effectiveness. This can be attributed to the first-stage system using document translation

performing better than the other two first-stage systems. The PSQ pipeline is the next best-performing pipeline, as it outperforms the query translation pipeline in all languages except French. This finding is noteworthy, as it shows that reasonable CLIR performance can be achieved by using queries and documents in their original form. However, it is important to consider the feasibility of using document translation pipelines, especially in scenarios where content is rapidly generated, such as in streaming applications, as each new document needs to be translated. We explore this aspect in detail in Chapter 6. Overall, in the absence of external translation systems, the PSQ pipeline provides a reasonable alternative for CLIR, and from here on, we conduct our experiments using it as the baseline.

3.6 In-depth Analysis of CREPE

In this section, we aim to understand why CREPE works and the impact of different first-stage CLIR systems on the CREPE approach.

3.6.1 Ablating CREPE

To analyze the impact of CREPE on CLIR effectiveness, we conduct an ablation study. We choose the best reranker, XLM-R, using title queries trained with CREPEs from the PSQ-HMM CLIR system as the default condition. We then subsequently create new training datasets by switching from CREPE to the stochastic approach for either the positive samples (relevant documents) or the negative samples (judged non-relevant documents). More specifically, in the case of ablating positive samples, we swap the passage returned by the PSQ-HMM with the first passage of the document, keeping the negative

Table 3.3: Analyzing CREPE: The default model is XLM-R fine-tuned with the dataset created using the CREPE approach. Δ denotes the average relative increase (+) or decrease (-) compared to the default model.

Reranker	POS	NEG	EN-FR	EN-ES	EN-IT	EN-FI	Δ
XLM-R - **Default**	PSQ-HMM	PSQ-HMM	0.4930	0.5260	0.4242	0.5110	-
Ablating CREPEs (Section 3.6.1)							
-negative	PSQ-HMM	First	0.4649	0.5007	0.3770	0.4866	-6.4%
-positive	First	PSQ-HMM	0.3956	0.4394	0.3653	0.4154	-17.1%
Tuning positive CREPEs (Section 3.6.2)							
2CREPE	PSQ-HMM	PSQ-HMM	0.4907	0.5171	0.4106	0.5024	-1.76%
1CREPE+FirstP	PSQ-HMM+First	PSQ-HMM	0.4834	0.5007	0.4117	0.5141	-2.27%
3CREPE	PSQ-HMM	PSQ-HMM	0.4801	0.5164	0.4091	0.4819	-3.42%
Choice of CREPEs (Section 3.6.3)							
Swapping both	GT	GT	0.4965	0.5274	**0.4288**	0.5354	+1.90%
	HT	HT	0.4997	0.5278	0.4238	**0.5467**	+0.07%
Swapping positive	GT	PSQ-HMM	**0.5032**	0.5231	0.4279	0.5319	+1.80%
	HT	PSQ-HMM	0.4962	**0.5284**	0.4245	0.5176	+0.08%
Swapping negative	PSQ-HMM	GT	0.4933	0.5140	0.4073	0.5279	-0.06%
	PSQ-HMM	HT	0.4249	0.5180	0.4140	0.5165	-1.96%

samples unchanged. For unjudged documents, we use the same passage as used in the training of the default XLM-R model. We then fine-tune the reranking model with the ablated dataset using the same setup as described in Section 3.3. Alternatively, we replace the negative samples only and fine-tune the model again.

The first group of rows in Table 3.3 analyzes the impact of swapping the training strategies. Both cases, involving positive and negative samples, see a drop in MAP compared to the default setting; this is observed across the four languages used in this study. However, the drop from swapping out CREPEs for positive samples (17.1%) is twice as high as the drop from swapping out CREPEs for negative samples (6.4%). This suggests that training the reranker with positive CREPEs from the PSQ-HMM system is particularly useful.

3.6.2 Tuning positive CREPE

Instead of using the best single passage from the PSQ-HMM for training, we try selecting more passages and see how that affects MAP. Specifically, we select top-m passages from the PSQ-HMM system and add them to the training set. We also try hybrid strategies of adding the best passage(s) from the PSQ-HMM system and the first passage from the document, which we explored in Section 3.4 We refrain from adding more than one negative sample (per document). By design, we have more non-relevant than relevant documents, and this choice avoids increasing that skew. The second group of rows in Table 3.3 present the results of tuning the positive CREPEs. In comparison to the default model, we see no improvements in increasing the number of positive instances using both top-m passages and the hybrid strategy, with a drop in effectiveness as more instances are added. This suggests that an integer count-based threshold to select passages might not be the best choice; instead, a score-based approach to select passages that exceed a certain threshold might work well, as observed in Rudra and Anand [159]. However, the challenge lies in computing a global score threshold given the unnormalized PSQ-HMM scores.

3.6.3 Choice of CREPE

Next, we analyze whether we get further improvements by swapping PSQ-HMM to query translation using either Google or human-created monolingual queries to produce better CREPEs. Specifically, we use our baseline PSQ-HMM retrieval system to retrieve a set of documents for a given query. After that, we score the passages from those retrieved documents using the monolingual BM25 system by using the translated query produced by

either Google Translate (GT) or using a human-translated query (HT). We then pick the highest scoring passage as per the monolingual retrieval system and then follow the same process of creating the dataset as listed in Section 3.2.1. Once the dataset is created, we train the model using the same setup as described in Section 3.3

The third group of rows in Table 3.3 presents the results of switching between PSQ-HMM, GT, and HT for generating CREPE. We observe MAP improvements over the default PSQ-HMM using the CREPE from GT or HT. Except in Finnish, CREPEs using GT outperform those produced by HT. However, the improvements are fairly modest compared to the default PSQ-HMM, except in Finnish, where they improve the results substantially. This demonstrates that the training of the model using CREPEs from PSQ-HMM is fairly robust.

Similarly, we conduct a fine-grained analysis by swapping the first-stage system for either the positive or negative sample, keeping the other constant (PSQ-HMM). Looking at Table 3.3, it is clear that, except for Finnish, essentially all of the gains are coming from using CREPE to sample positives. This might also be related to the finding from the ablation study in Section 3.6.1, where we observed a sharp drop in MAP on switching from CREPE to stochastic approach for positive samples. However, except for Spanish, MAP decreases when using query translation to sample CREPE negatives. This leads us to conclude that for sampling negatives, using signals from an improved lexical system instead of the underlying PSQ-HMM model doesn't necessarily help. Rather than relying on a different system, it may be better to use the same lexical system to sample passages that it finds most confusing. Whereas for positive samples, an improved lexical system can find training passages that the reranker can use to learn a better model.

3.7 Chapter Summary

In this chapter, we introduce the CREPE approach, which leverages the first-stage CLIR systems in both the training and querying phases of retrieve-and-rerank CLIR pipelines. By using CREPE, we can subselect passages to be scored by a cross-encoder during the querying phase, balancing retrieval effectiveness and query latency. We propose several Pareto-optimal selection strategies that combine the FirstP strategy with CREPE. During the training phase, we integrate signals from the first-stage system using CREPE to fine-tune the contextual embeddings, leading to better effectiveness than traditional strategies that do not use text signals. We evaluate our approach on CLEF document collections in several languages, and our results show significant improvements over the first-stage retrieval system and over neural rerankers trained using the stochastic approach.

Chapter 4: Transfer Learning for Neural CLIR[1]

In Chapter 3, we examined several retrieve-and-rerank pipelines. These pipelines involve first-stage CLIR systems whose output is fed to interaction-based pointwise cross-encoders that utilize PLMs acting as neural rerankers. While using these cross-encoders for reranking has proven effective, a major challenge arises when reranking multiple documents in response to a query due to the linear growth in time complexity as the document collection increases in size. As a result, the number of documents to be reranked must be tuned to strike a balance between query latency and retrieval effectiveness. The primary cause of this increase in time complexity is that the cross-encoder computes contextual embeddings over the concatenated sequences of query and document tokens, which is then coupled with quadratic time and space complexity of self-attention in the PLM's transformer layers. On the other hand, a different neural ranking model, known as the representation-based model, computes the query and document embeddings separately, providing two distinct advantages over retrieve-and-rerank pipelines.

Firstly, it allows for precomputing the embeddings for the entire document collection as part of the indexing phase, and storing them in specialized indexes, enabling fast com-

[1]This chapter contains content from: **Suraj Nair**, Eugene Yang, Dawn Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W Oard. "Transfer learning approaches for building cross-language dense retrieval models." In Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022 [127]

putation of approximate nearest neighbors [69, 83, 115] with sublinear time complexity in response to a query. In contrast, the document embeddings for cross-encoders must be computed for each query during the querying phase since the cross-encoder works with concatenated sequences of queries and documents. Secondly, representation-based models perform matching of the query and document representations in a shared vector space that captures the notion of relevance using custom similarity functions such as cosine similarity. Matching in a vector space thus helps partially mitigate vocabulary mismatches typically found in first-stage keyword-based lexical systems, which are further exacerbated in the case of CLIR due to translation errors.

Representation-based models that use PLMs as their underlying encoder, commonly referred to as bi-encoders, have become increasingly prominent in monolingual retrieval applications [87, 152, 194]. The term "bi" refers to the individual encoders used for processing queries and documents, which can either be shared between the two or kept separate. These bi-encoder models can be broadly classified into two variants: single-representation and multi-representation. Single-representation bi-encoders [87, 105, 106, 107, 108, 194] encode queries and documents to generate a fixed-size single aggregated representation. While the compression of document and query embeddings into single vectors results in a lower indexing footprint and reduced query latency, this comes at the cost of reduced expressiveness, which can impact retrieval effectiveness. This is primarily due to the fact that queries, which are usually shorter in length than documents, lack the joint context in bi-encoders that cross-encoders utilize to generate better contextual embeddings. Moreover, compressing the representations of both queries and documents into single vectors can result in information loss that adversely affects retrieval effectiveness.

In contrast, multi-representation bi-encoders [77, 89] generate multiple vectors for queries and documents, built on top of individual term embeddings for matching. While using multiple vectors has the potential to recover some of the information loss in single vectors, thereby increasing modeling expressivity, it also leads to an increase in indexing footprint and query latency. Thus, designing bi-encoders that can balance the tradeoff between retrieval effectiveness and indexing/query latency becomes crucial. Currently, ColBERT [89], a multi-representation bi-encoder that computes the similarity between each query term representation and each document term representation, provides the best tradeoff in monolingual retrieval applications. In this chapter, we explore a generalization of the ColBERT approach that supports CLIR.

The generalization of ColBERT to CLIR is not trivial and poses two problems. First, the bi-encoder must be able to process the query and document languages to perform CLIR. Second, finding appropriate CLIR resources for training the bi-encoder model is challenging due to the lack of large-scale collections like the MS MARCO [11] dataset, which is widely used for training monolingual neural retrieval models. To address these challenges, this chapter introduces *ColBERT-X*, a generalization of ColBERT for CLIR. We utilize XLM-R [40], an MPLM, to initialize the bi-encoder model, enabling us to perform CLIR. We employ cross-lingual transfer learning techniques to train two variants of ColBERT-X: zero-shot, where the model is trained on MS MARCO in English, and translate-train, where we use machine-generated translations of MS MARCO passages paired with English queries as the training resource.

The remaining sections are in the following order. In Section 4.1, we introduce the details of ColBERT-X and the process by which we train the model for CLIR. Section 4.2

introduces the experimental setup used to evaluate the ColBERT-X model. We present the effectiveness of ColBERT-X in Section 4.3, followed by improvements to ColBERT-X in Section 4.4. We conduct several analyses in Section 4.5 and conclude the chapter with a summary in Section 4.6.

4.1 ColBERT-X

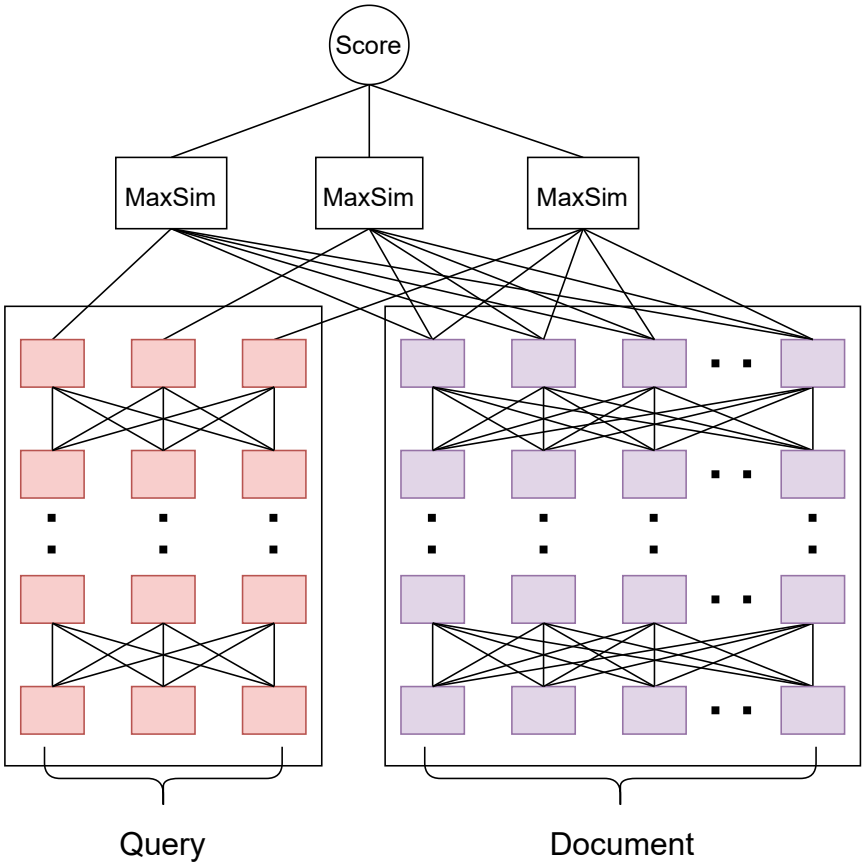


Figure 4.1: ColBERT-X multi-representation bi-encoder architecture

ColBERT is a bi-encoder model that utilizes monolingual BERT [48] to encode both query and document terms.[2] The model follows the single-representation bi-encoder archi-

[2]In this setting, the query and the document encoders share the parameters as they are initialized with the same BERT models. Note that this is distinct from the DPR [87] model, where the query and the document encoder do not share the parameters.

tecture, which computes contextual term embeddings separately for queries and documents. Additionally, inspired by the cross-encoder architecture, ColBERT adds a single layer of interaction called the "late-interaction," which operates on the contextual term embeddings computed from the previous step. By combining the strengths of the two architectures, ColBERT aims to create a unified model.

The key design choice of ColBERT is the function used in the late-interaction step since multiple options are available. While a single transformer layer is a straightforward choice from cross-encoders, ColBERT uses MaxSim, a heuristic inspired by term matching in the vector space, as shown in Figure 4.1. MaxSim finds the most similar document term for a given query term by utilizing a custom similarity function applied to the contextual query and document term embeddings. This setup closely follows those found in keyword-based systems, which reward the lexical match between the query and document terms, but in a vector space. However, the main benefit of MaxSim lies in its ability to function as a first-stage system that helps triage the set of documents for the late-interaction step.

Although ColBERT allows for the separate computation of query and document embeddings, the time complexity of the late-interaction step grows linearly as the document collection scales, the same as in the cross-encoder. To address this issue, ColBERT operates in two modes. The first mode is similar to the retrieve-and-rerank architecture introduced in Chapter 3, where a first-stage retrieval model produces the initial set of documents, which are then fed to the ColBERT late-interaction step, functioning as the reranker. This mode is referred to as "reranking." One disadvantage of this mode is that the overall recall of the system is limited to the recall of the initial set. In the context of CLIR systems, we face the additional complexity of crossing the language barrier, which further affects recall.

In the second mode, ColBERT utilizes the MaxSim heuristic to create the initial set of documents by employing nearest-neighbor techniques. For each query term embedding, the model identifies the k nearest document terms using fast Approximate Nearest Neighbor (ANN) methods. These terms are then mapped to their respective document ids, and the initial (unordered) set of documents is generated by computing the union of these ids. Subsequently, the late-interaction step reranks this set, as in the reranking mode. The advantage of MaxSim over traditional keyword-based matching systems is that the contextual term embeddings for queries and documents can be fine-tuned on a training collection with relevance judgments, thus improving recall beyond lexical matches. Equation 4.1 shows how the final score of the document is computed as the sum of individual query term contributions.

$$s_{q,d} = \sum_{i=1}^{|q|} \max_{j=1..|d|} \eta(q_i) \cdot \eta(d_j)^T \tag{4.1}$$

Here, η denotes the monolingual BERT encoder, and the similarity function is chosen as the dot product between the two vectors.

To generalize ColBERT to CLIR, we replace monolingual BERT with XLM-R. We call the resulting model ColBERT-X. Initializing the encoder to a multilingual model allows retrieval in any language supported by the embeddings. However, these models must be trained before they can be used for CLIR.

4.1.1 CLIR Training Strategies

ColBERT was trained using pairwise cross-entropy loss on MS MARCO [11] triples, which consists of an English query, a relevant English passage, and a non-relevant English

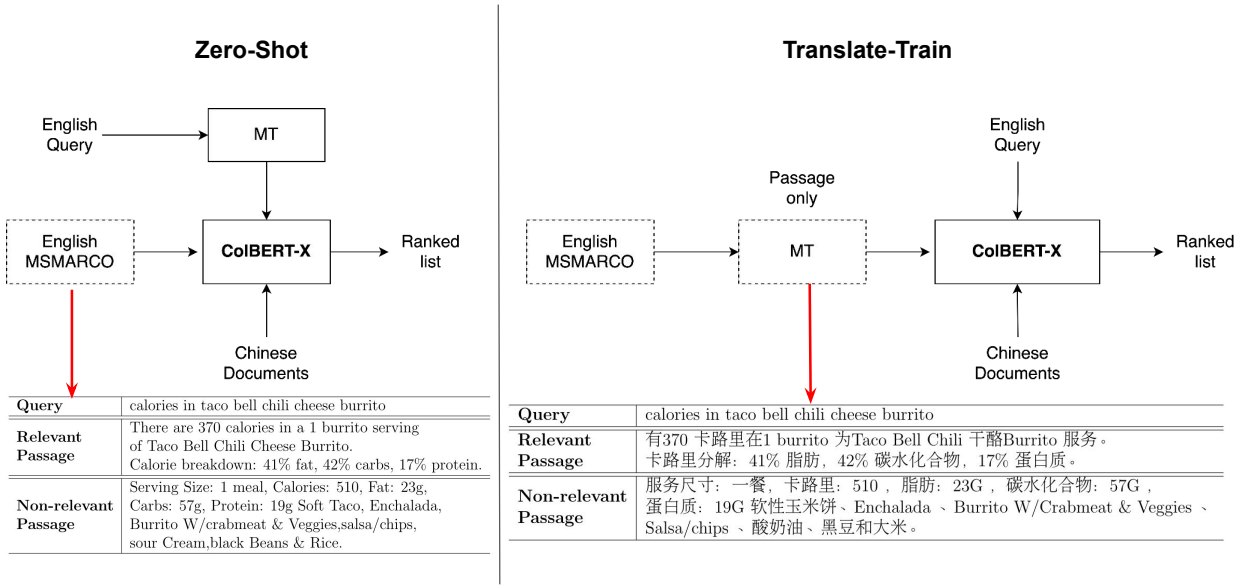


Figure 4.2: Two ColBERT-X Transfer Learning Pipelines: Zero-Shot (left) and Translate-Train (right). Dashed boxes denote the components used during the training step. In the zero-shot scenario, ColBERT-X trained on English MS MARCO is applied directly to the translated queries. With the translate-train setting, the training set consists of translated passages to enable ColBERT-X to cross the language barrier.

passage. To train ColBERT-X for CLIR, we explored two strategies from the cross-language transfer learning literature:

1. Zero-Shot: This is a common technique in which a multilingual model (e.g., mBERT or XLM-R) is trained in a high-resource language (usually English) and then applied to the document language. In this chapter, we first train a ColBERT-X model initialized with an XLM-R encoder on English MS MARCO passage ranking triples. At query time, we use machine translation (MT) to translate the English query to the document language, and use the trained ColBERT-X model to perform retrieval in the document language using Equation 4.2. $\hat{q}$ is the translated query. Multilingual language models have demonstrated good cross-language generalization in many

other natural language processing tasks; we hypothesized it would also work well for CLIR. Notably, the proposed zero-shot with translated queries is different from the actual zero-shot setting, where the queries and documents would be in their respective native languages. We observe higher effectiveness when using translated queries compared to using queries and documents in different languages. One reason for this could be the absence of explicit cross-language supervision during the pre-training of multilingual models, which can affect their generalization during zero-shot transfer, as noted by Karthikeyan et al. [88].

$$s_{\hat{q},d} = \sum_{i=1}^{|\hat{q}|} \max_{j=1..|d|} \eta(\hat{q}_i) * \eta(d_j) \tag{4.2}$$

2. Translate-Train: In this setting, an existing high-resource language (e.g., English) collection is translated into the document language. As in zero-shot training, we choose training triples from the MS MARCO passage ranking collection and use a trained MT model to translate them. Since our focus here is on using English queries to retrieve content in non-English languages, we pair the original English queries with machine translations of relevant and non-relevant MS MARCO passages to form new triples. We then train ColBERT-X on these newly constructed triples in the same manner as ColBERT.

Figure 4.2 shows these two pipelines. The key difference is that in the zero-shot setting, we have a single ColBERT-X model for a given query language (in this case, English) that is used for retrieval in multiple document languages. In the translate-train setting, we train a ColBERT-X model for each query-document language pair.

4.1.2 Retrieval

While we train ColBERT-X on passages, our goal is to rank documents. We split large documents into overlapping passages of fixed length with a stride. During indexing, we use the trained ColBERT-X model to generate term representations from these passages. These representations are stored in a FAISS-based ANN index [83], and are saved to disk for subsequent MaxSim computation. At query time, we generate a ranked list of passages for each query and then use a document's maximum passage score as its document score.

4.2 Experiments

In this section, we describe the following: the test collections, the training and retrieval design choices for ColBERT-X, the MT systems utilized, the baselines, and the evaluation measures.

4.2.1 Collection Statistics.

Table 4.1 provides details for the test collections used in our experiments. We worked with several languages from the 2000 to 2003 Cross-Language Evaluation Forum (CLEF) evaluations [145], using news collections for French, German, Italian, Russian, and Spanish. We also conducted experiments using the new CLIR Common Crawl Collection (HC4) [99], where the documents are newswire articles from Common Crawl in Chinese or Persian. Throughout, English queries are used to search documents in a non-English language. We experiment with title and description queries. The MS MARCO [11] passage ranking

dataset, which we use for training ColBERT-X, consists of roughly 39M training triples, spanning over 500k queries and 8.8M passages.

4.2.2 ColBERT-X Training and Retrieval.

Our two ColBERT-X model strategies, zero-shot (ZS) and translate-train (TT), are trained using mostly the same hyperparameters used to train the original ColBERT model.[3] We replaced the BERT encoder with the XLM-RoBERTa (large) encoder provided by the HuggingFace transformers [187] library (but see Section 4.5.2 for mBERT results). To generate passages from documents, we use a passage length of 180 tokens with a stride of 90 tokens. We index these passages using the trained ColBERT-X model in the same way as the original ColBERT model setting.[4]

4.2.3 Machine Translation.

For CLEF document languages, we use MS MARCO passage translations[5] from Bonifacio *et al.* [22], and the same MT model to translate queries. For the HC4 languages, we use directional MT models built on top of a transformer base architecture (6-layer en-

[3] We increase our batch size from 32 to 128
[4] `https://github.com/stanford-futuredata/ColBERT#indexing`
[5] `https://github.com/unicamp-dl/mMARCO`

Table 4.1: Test collection statistics for the CLEF and HC4 newswire collections.

Collection	HC4 Chinese	HC4 Persian	CLEF French	CLEF German	CLEF Italian	CLEF Russian	CLEF Spanish
#documents	646K	486K	129k	294k	157k	16k	454k
#passages	3.6M	3.1M	0.7M	1.6M	0.8M	0.1M	2.7M
#queries	50	50	200	200	200	62	160

coder/decoder) using the Sockeye toolkit. [49] To produce translations of MS MARCO, the original passages were split using *ersatz* [186], and sentence-level translation was performed using the trained MT model.

4.2.4 Baselines.

We compare these two strategies with several lexical and neural reranking baselines, grouped as follows:

- Human Translation: Monolingual retrieval using Anserini BM25 [199] with the document-language queries provided in the test collection.
- Query Translation: BM25 retrieval using translated queries produced by a specific MT model and original documents in the target language.[6]
- Reranking: We rerank query translation baseline results using the publicly available multilingual T5 reranker[7] trained on translated MS MARCO in 8 languages [22].

4.2.5 Evaluation.

We evaluate ranking using Mean Average Precision (MAP). Differences in means are tested for significance using a paired two-tailed t-test ($p < 0.05$) with Holm-Bonferroni multiple test correction.

[6] We use the same MT model to translate the queries as the one used to translate the MS MARCO passages.

[7] `https://huggingface.co/unicamp-dl/mt5-base-multi-msmarco`

Table 4.2: Effectiveness results (MAP) for CLIR HC4 and CLEF collections using title queries. Statistically significant improvements over the query translation and reranking baselines are marked with * and † respectively.

Collection(→) Model	HC4 Chinese	HC4 Persian	CLEF French	CLEF German	CLEF Italian	CLEF Russian	CLEF Spanish
human translation							
BM25	0.301	0.276	0.403	0.304	0.350	0.452	0.452
ColBERT-X (ZS)	0.510	0.343	0.401	0.360	0.328	0.479	0.418
query translation							
BM25	0.237	0.211	0.387	0.263	0.275	0.377	0.405
reranking							
BM25+mT5-multi	0.312	-	0.333	0.297	0.279	0.303	0.370
our methods							
ColBERT-X (ZS)	**0.450***†	0.297*	0.382†	0.328*†	0.272	**0.418**†	0.379
ColBERT-X (TT)	0.408*†	**0.310***	**0.422**†	**0.397***†	**0.339***†	0.410†	**0.415**†

4.3 Retrieval Effectiveness of ColBERT-X

Table 4.2 compares the effectiveness of our models to the baselines. Our main finding is that both ColBERT-X variants perform better than BM25 query translation baselines in general. ColBERT-X (ZS) trained using English MS MARCO alone performs better than the query translation baseline BM25 and fine-tuning the ColBERT-X (TT) on translated MS MARCO data helps improve the effectiveness further. These gains are statistically significant in both HC4 collections and many of the CLEF collections.

We also compare the ColBERT-X variants to the multilingual T5 reranker that reranks the query translation baseline output. In each of the collections, ColBERT-X (with 550M parameters) performs consistently and significantly better than the reranker (580M parameters). This is particularly interesting in CLEF collections since both the

mT5 reranker and ColBERT-X (TT) was trained on the same MS MARCO translations. However, training the reranker on a combined dataset in 8 languages highlights the curse of multilinguality [40]. This refers to the degradation in the system's performance as the number of supported languages increases.

When we compare the two variants of ColBERT-X, we observe that, on average, translate-train often does better than zero-shot, but these differences are only significant in CLEF collections except Russian and not in HC4 collections. The difference is likely a result of using different MT models in CLEF and HC4 collections, so we conduct this analysis later.

4.4 Improving ColBERT-X effectiveness: Pseudo-Relevance Feedback

Pseudo-relevance feedback (PRF) is a form of query expansion that adds discriminative terms extracted from a set of "feedback" documents. While PRF has been explored for pre- and post-translation query expansion [121], here we choose cross-language expansion terms using the ColBERT-X term representation, as recently suggested by Wang et al. [184]. First, the feedback documents (fb-docs) are selected from the top of a ColBERT ranked list. Next, embeddings of terms from the feedback documents are clustered into m distinct clusters.

The top-ranked centroids of these m clusters[8] are then used as feedback embeddings (fb-embs) for query expansion. These fb-embs are added to the original query terms, and the ColBERT MaxSim heuristic is applied to the resulting queries to produce the final

[8]Each cluster centroid representation is mapped to the nearest document token using the ANN index, with the document token IDF as the score

Table 4.3: MAP for query translation BM25 and ColBERT-X translate-train, with and without PRF. * and † denote significant improvements over BM25 with PRF and ColBERT-X (TT) respectively

Retrieval Model	CLEF French	CLEF German	CLEF Italian	CLEF Spanish
baseline				
BM25	0.387	0.263	0.275	0.405
ColBERT-X (TT)	0.422	0.397	0.339	0.415
with PRF				
BM25	0.410	0.321	0.320	**0.438**
ColBERT-X (TT)	**0.459***†	**0.406***†	**0.371***†	0.436†

ranked list. We generalize this approach to the ColBERT-X CLIR setting.

To better understand the effect of PRF, we compare ColBERT-X (TT) and query translation BM25, with and without PRF. For BM25, we use Anserini's RM3 implementation to perform PRF with default hyperparameter values. For ColBERT-X PRF, we extend Terrier's [139] implementation[9] with default hyperparameters in the ranking setting. Table 4.3 shows the effect of PRF on ColBERT-X translate-train performance using MAP. Except in Spanish, applying PRF to ColBERT-X significantly improves effectiveness compared to ColBERT-X without PRF as well as BM25 with PRF.

4.5 Detailed Analysis

This section considers several aspects of ColBERT-X. First, different machine translation models are compared using both MT and CLIR measures. Second, effects of different multilingual encoders are explored. Third, the impact of pseudo-relevance feedback is examined. Then the influence of query length on performance is considered. Finally,

[9] `https://github.com/terrierteam/pyterrier_colbert`

ColBERT-X costs in terms of index size are noted.

4.5.1 Effect of Machine Translation

ColBERT-X utilizes machine translation in two different ways depending on whether it is trained using the zero-shot strategy or the translate-train strategy. In the zero-shot strategy, the queries are translated to the document language at query time, while the translate-train strategy requires an MT system to translate the monolingual training corpus (in this case, the MS MARCO passages) to the document language. The MT systems used to produce translations include:

- OpusMT – bidirectional MT model(s) with MarianNMT as the base architecture, [10] released by the Helsinki NLP group from Bonifacio *et al.* [22].
- SockeyeMT1 – MT model built on top of a transformer base architecture (6-layer encoder/decoder) trained on bitext. Depending on language, these include publicly available bitext such as OpenSubtitles, UN Corpus, Europarl, and WMT. The model is trained using AWS Sockeye v2 [49].
- SockeyeMT2 – identical model architecture to SockeyeMT1 but trained with 2x - 3x more bitext. The number of training sentence pairs for MT1 v.s. MT2 are, respectively, 51M v.s. 120M for Russian, 36M vs 85M for Chinese, and 6M v.s. 11M for Persian.

Table 4.4 provides an intrinsic comparison of the systems translating from English on a translation task using BLEU scores [140]. BLEU is a metric used to evaluate the quality

[10] `https://huggingface.co/Helsinki-NLP`

Table 4.4: BLEU scores for translation systems using WMT'19 newstest for Chinese and Russian, and TICO-19 (from OPUS[11]) for Persian. These are computed on test sets distinct from the CLIR collections, so the absolute BLEU score is not an exact reflection of the quality of translations in CLIR experiments. Nevertheless, the relative comparison of BLEU scores among MT systems is meaningful.

Language Benchmark	Russian *newstest'19*	Chinese *newstest'19*	Persian *tico-19*
OpusMT	26.3	14.6	-
SockeyeMT1	32.1	25.8	4.4
SockeyeMT2	**35.9**	**38.6**	**20.2**

of machine translations, which measures the similarity between the machine-generated translation and one or more human-generated reference translations. For Russian and Chinese, we evaluated using newstest'19 dataset from the shared task in Workshop in Machine Translation (WMT); for Persian, we evaluated with TICO-19, a collection of about 3000 sentences about COVID-19, as no WMT test data is available. Scores were calculated with *sacrebleu* [148] using the lowercase i.e., `-lc` setting. The table reveals that SockeyeMT outperforms OpusMT and that exposing SockeyeMT to more training data improves the BLEU score.

Table 4.5 shows that improving BLEU scores likely leads to improvements in CLIR

[11] `https://opus.nlpl.eu/`

Table 4.5: Effect of different MT models for ColBERT-X at query time and training time on the downstream CLIR task, measured using MAP scores.

MT model	CLEF Russian	HC4 Chinese	HC4 Persian
OpusMT	0.418	0.411	-
SockeyeMT1	0.442	0.391	0.230
SockeyeMT2	**0.461**	**0.450**	**0.297**

(a) ColBERT-X zero-shot

MT model	CLEF Russian	HC4 Chinese	HC4 Persian
OpusMT	0.410	0.365	-
SockeyeMT1	**0.459**	0.389	0.287
SockeyeMT2	0.456	**0.408**	**0.310**

(b) ColBERT-X translate-train

Table 4.6: MAP scores for ColBERT-X (TT) initialized with the mBERT and XLM-R encoders, and trained on SockeyeMT1 MS MARCO translations.

Multilingual Model	CLEF Russian	HC4 Chinese	HC4 Persian
mBERT	0.341	0.284	0.173
XLM-R	**0.459***	**0.389***	**0.287***

for both training strategy. Table 4.5a shows the results of translating queries in the zero-shot strategy. While BLEU improvements tend to be realized downstream, this is not seen for HC4 Chinese where OpusMT has a better MAP score than SockeyeMT1. It should be noted that asking MT systems to translate title keyword queries may not align well with how the systems were trained with complete sentences.

Table 4.5b shows results for using different translation models on MS MARCO triples, and the effect this has on ColBERT-X retrieval as measured using MAP. Again, we see that the MAP scores tend to improve with improved BLEU; however, in this case the improvement in Russian BLEU from Table 4.4 between SockeyeMT1 and SockeyeMT2 does not carry over to ColBERT-X, where the performance is essentially the same. Generally, one can expect that improving MT quality will lead to improved effectiveness of ColBERT-X.

4.5.2 Effect of Multilingual Language Models

Comparing different multilingual encoders to initialize ColBERT-X, we observe that XLM-R performs significantly better than mBERT, as shown in Table 4.6. While this might be unsurprising given that the XLM-R model is twice as large and was pretrained on more data than mBERT, tokenization differs across the languages. Considering the

Table 4.7: MAP results for ColBERT-X (TT) model using different query representations. * and † denote significant improvements over title and description queries respectively.

Query Representation	CLEF French	CLEF German	CLEF Italian	CLEF Spanish
title	0.422	0.397	0.339	0.415
description	0.434	0.410	0.380	0.456
title+description	**0.507***†	**0.466***†	**0.424***†	**0.500***†

case of Chinese, mBERT tokenization produces character-level tokens, whereas the XLM-R tokenizer generates subwords (sentencepieces). This also implies that mBERT indexes are larger than XLM-R indexes, resulting from the term-level storage requirements of the ColBERT-X model.

4.5.3 Effect of Longer Queries

Table 4.7 analyzes the effect of different query representations on the ColBERT-X translate-train. We compare three representations: *title* (t), which usually corresponds to a short Web search query; *description* (d), a well-formed sentence describing the information need, and *title+description* (td), the concatenation of the two. Longer queries pose a problem for ColBERT-X, however, since the model only supports queries up to 32 tokens long. To mitigate this problem, we use a list of "stop structures"[4] consisting of phrases (e.g. find documents on, reports of, etc.), which have been shown to work in the past, removing them from the td queries. We observe that td with stop structures removed leads to significant improvements over t or d alone.

4.5.4 Indexing Space Footprint

In addition to the FAISS-based ANN index, ColBERT-X requires access to the representation of each passage term to compute MaxSim. With each term embedded as a 128-dimensional vector and each embedding dimension represented using 16-bits, we would need 256 bytes of storage per term. These are onerous storage requirements, with the index sizes increasing with the collection size. Table 4.8 provides collection-specific statistics on the disk space required to store the document collections. An important design artifact that affects the index size is the way passages are generated from the documents. Since we employ a sliding window of document tokens, this means most of the tokens have two term representations generated.

Table 4.8: Collection-specific memory footprint.

Collection	HC4 Chinese	HC4 Persian	CLEF French	CLEF German	CLEF Italian	CLEF Russian	CLEF Spanish
#passages	3.6M	3.1M	0.7M	1.6M	0.8M	0.1M	2.7M
Disk Space	154GB	134GB	33GB	70GB	36GB	4.7GB	117GB

4.6 Chapter Summary

In this chapter, we introduce ColBERT-X, a cross-language generalization of ColBERT, which uses a multilingual query and document encoder to improve CLIR effectiveness beyond what traditional systems such as BM25 can achieve. To train ColBERT-X, we used MT systems to translate MS MARCO and create CLIR collections. We have shown that performing cross-language expansions using ColBERT-X model with PRF can lead to

significant gains in retrieval effectiveness. Furthermore, we have analyzed the impact of MT on the downstream CLIR task.

Chapter 5: Efficient First-Stage Sparse Bi-Encoders for CLIR[1]

First-stage retrieval systems play an essential role in multi-stage retrieval architectures by identifying top-k documents from a large collection while focusing on high recall, i.e., finding as many relevant documents as possible for a given query. However, it is equally important that these systems process the documents efficiently to maintain the desired query latency through the subsequent expensive stages of the retrieval architecture, including reranking. In Chapter 3, we explored traditional first-stage CLIR systems that are part of the retrieve-and-rerank architecture. These systems relied on the outputs of translation models, which impacted recall while using inverted indexes to facilitate fast retrieval. In Chapter 4, we explored a vector-similarity-based first-stage CLIR system that leveraged contextual embeddings from multilingual bi-encoders fine-tuned on translated retrieval collections, contributing to its high recall while utilizing techniques from ANN systems for fast retrieval. This chapter introduces another set of first-stage CLIR systems that use contextual embeddings from multilingual bi-encoders to learn sparse vectors.

[1]This chapter contains content from following papers:

- **Suraj Nair**, Eugene Yang, Dawn Lawrie, James Mayfield, and Douglas W Oard. "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, 2022 [128], and
- **Suraj Nair**, Eugene Yang, Dawn Lawrie, James Mayfield, and Douglas W Oard. "BLADE: Combining Vocabulary Pruning and Intermediate Pretraining for Scaleable Neural CLIR." In Preparation. [130]

The primary motivation is to harness the knowledge of PLM stored in their language modeling (LM) head to build a sparse representation for queries and documents. The core idea is to represent queries and documents in a high-dimensional space defined by PLM's vocabulary, where only a few dimensions (corresponding to vocabulary terms) have non-zero values. The advantage of sparse representation is that these non-zero document term weights can then be stored in an inverted index, allowing the efficiency of traditional sparse retrieval approaches to be exploited. In addition, the sparsity of the representation allows for inspecting the model outputs and building explainable models. Furthermore, this framework allows for query or document expansion by generating weights for terms that did not appear in either the queries or documents but plausibly could have. This approach thus could potentially help in partially mitigating the vocabulary mismatch issue faced by the bag of words models such as BM25 [60, 154].

The use of PLMs (e.g., BERT [48]) to learn such sparse representation models has become increasingly popular in monolingual information retrieval applications, particularly for English content. With the availability of large-scale training collections such as MS MARCO [11] that have been translated into multiple languages [22, 127] and an increasing variety of sparse representation models for monolingual retrieval [10, 45, 55, 56, 104, 118, 204, 210], a natural question is whether extending these ideas to CLIR involves anything more than simply replacing a monolingual pretrained model (e.g, BERT) with a multilingual model (e.g., mBERT [48] or XLM-R [40])? However, we identify two main challenges when it comes to building sparse bi-encoder models that utilize the vocabulary space of the underlying PLM.

Firstly, the vocabulary size of MPLMs such as mBERT and XLM-R is roughly 3 to

7 times larger than the size of monolingual BERT vocabulary, resulting in a larger dimensionality of sparse vectors. This increased dimensionality can adversely impact the model's efficiency during the training, indexing, and querying phases. This vocabulary selection problem is similar to that in the MT setting, where the trade-off between translation latency and MT output quality, measured using automatic metrics such as BLEU [141], has been well studied [50, 78, 169]. Similarly, in the CLIR setting, the vocabulary selection generates a trade-off between training time, indexing latency, query latency, and retrieval effectiveness. Secondly, the similarity of representations for terms from different languages with similar meanings may be inadequate. We addressed this issue in Chapter 4 by learning cross-language term associations from translated MS MARCO passages paired with English queries using a translate-train approach. However, this setting is prone to translationese, where the passages generated by the MT system contain translation artifacts that might not be present if the passages were expressed in their natural forms.

To address these challenges, we first propose SPLADE-X, a bi-encoder model initialized with mBERT to generate sparse vectors for CLIR. The design of SPLADE-X is inspired by its monolingual cousin, SPLADE [56], which has been shown to generalize well in multiple retrieval scenarios. SPLADE-X employs a vocabulary reduction technique that restricts the multilingual vocabulary space to the terms corresponding to the query language only. This choice forces the model to learn query expansion to potentially include terms that were not present in the query and, more importantly, cross-language lexical expansions for the non-English documents, which roughly corresponds to an encoder-only translation task. We subsequently propose BLADE, a bi-encoder model initialized with a pruned bilingual PLM, where the output dimensions now correspond to the terms in both

query and document languages. This allows the model to learn bilingual term expansions for both queries and documents. Finally, we propose an intermediate pretraining step using aligned text pairs expressed in their natural form to reduce the impact of translationese present in translated collections.

The remainder of this chapter is structured as follows. In Section 5.1, we first give a brief overview of SPLADE and the key terminology associated with it that we use throughout the chapter. Section 5.2 introduces our proposed sparse CLIR models, SPLADE-X and BLADE, describes the key changes in the modeling design and the training objectives going from SPLADE to SPLADE-X to BLADE and compares these models to existing traditional approaches that rely on MT. We detail our experimental setup in Section 5.3, present our results focusing on retrieval effectiveness in Section 5.4, and query latencies in Section 5.5. We conclude the chapter with a summary in Section 5.6.

5.1 SPLADE

Pre-BERT sparse neural retrieval models [204] generated query and document vectors using L1 regularization, which enforces sparsity in the vectors and permits inverted indexing that is efficient during the querying phase. The advent of BERT led to different forms of neural ranking models, including those that generate sparse weights for query and document terms [10, 37, 45, 55, 56, 104, 118, 210]. We can group the existing models into two categories: a) exact-match, where weights are changed for terms that occur in queries or documents, but no nonzero weights are added for any additional terms; or b) lexical expansion, in which the number of terms with nonzero weights is still limited in some

way, but some terms that did not appear in the original query or the original document can be given non-zero weights. In the case of CLIR, the exact-match approach will not work (or at least it will not work very well!) because the queries and documents are expressed in different languages, generally using different words. Thus our natural point of comparison should be lexical expansion. In a monolingual lexical expansion setting, existing approaches either rely on off-the-shelf document expansion models such as doc2query [135] or TILDEv2 [209] to generate additional terms, or they use the vocabulary space of the PLMs for expansion [10]. Among the models built off of the latter framework, SPLADE [55, 56] 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 cousins, SPLADE-X and BLADE.

SPLADE [55, 56, 57, 98] is a lexical-expansion-based bi-encoder 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 encoder η, computes the similarity score $s(q, d)$ between them as:

$$s(q, d) = \eta(q)^T \eta(d) \tag{5.1}$$

Here, the query and the documents encoders are initialized with the same PLM.

Let $V_{\mathcal{T}}$ denote the output vocabulary space of the SPLADE model and $V_{\mathcal{Q}}$ and $V_{\mathcal{D}}$ be the subword vocabularies of the query and document languages, respectively. In the case of a monolingual SPLADE model initialized with a BERT encoder, $V_{\mathcal{T}}$ is equal to the size of the BERT vocabulary, i.e., $V_{\mathcal{T}} = V_{\mathcal{Q}} = V_{\mathcal{D}}$ with $|V_{\mathcal{T}}| = 30522$. 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:

$$w_{ij} = \phi(h_i)^T e_j + b_j \tag{5.2}$$

where ϕ is a combination of a linear layer with GeLU [71] activation, with LayerNorm [9] 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:

$$w_j = \max_i \log\left(1 + \text{ReLU}(w_{ij})\right) \tag{5.3}$$

A similar explanation follows for generating aggregate query vectors. Given the aggregate query and document vectors, the similarity score can be computed using Eq. (5.1). Sparsity is enforced in the document and query representations by combining ReLU [131] activation and FLOPS [142] regularization.

SPLADE [56] uses a contrastive ranking loss to train the 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:

$$\mathcal{L}_{\text{rank}} = -\log \frac{e^{s(q_i,d_i^+)}}{e^{s(q_i,d_i^+)} + e^{s(q_i,d_i^-)} + \sum_j e^{s(q_i,d_j^-)}} \tag{5.4}$$

In SPLADE, the contrastive ranking loss was trained on MS MARCO [11] training triples. We refer to this step as task-specific fine-tuning. Subsequent versions [55, 57] introduced distillation loss and a hard negative mining step. SPLADEv2 [55] leverages Margin-MSE loss [74] 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.

The choice of using LM head allows the SPLADE model to take advantage of the knowledge acquired during the pretraining phase of PLM, which involves self-supervised learning. Additionally, the model can learn to expand both queries and documents to include related terms that may not have been present in the original text. Sparsity plays a crucial role in this setting, as it helps control the expansion factor of queries and documents, preventing the expanded queries and documents from becoming too large and negatively impacting the query latencies and inverted index size.

5.2 Sparse Bi-Encoders for CLIR

Building a document expansion model that generalizes well beyond English is already a challenging problem [36, 168], one that becomes even more challenging given the explosive growth in the vocabulary size of multilingual pretrained models such as mBERT (110k) and

XLM-R (250k). These are 3 to 7 times the size of the monolingual BERT vocabulary (35k). The increased dimensions impact efficiency, as the larger vocabulary $|V|$ leads to increased memory use during training and higher latencies during the indexing and querying phases. It is these two factors, the need to generalize across languages and the potential benefits of limiting the vocabulary size, that distinguish CLIR applications of lexical expansion methods from their monolingual cousins. In this section, we introduce two sparse CLIR models, SPLADE-X and BLADE, and discuss their key design choices and the training process.

5.2.1 SPLADE → SPLADE-X

Generalizing the SPLADE model for CLIR applications, we first introduce SPLADE-X, which uses an mBERT encoder to generate aggregate term vectors similar to SPLADE, given a query and a document in different languages. To address the vocabulary selection problem, we limit the expansions to query-language terms, i.e., $V_{\mathcal{T}} = V_{\mathcal{Q}}$ for SPLADE-X. This essentially makes SPLADE-X an encoder-only MT model that translates (or expands) document language terms to query language terms, albeit with overgeneration and without a target language model. To limit the expansion factor of queries and documents, we use a top-l masking [198] scheme instead of the FLOPS regularization part of the original SPLADE model. This technique only preserves the dimensions corresponding to the top-l terms with the highest weights and sets the remaining weights to zero.

Like ColBERT-X in Chapter 4, we employ the translate-train approach for training SPLADE-X to learn term associations necessary for CLIR matching. Specifically, English

queries are paired with translated passages from English to the document language, and a contrastive ranking loss is learned as in Eq. (5.4) using these pairs as part of the CLIR task-specific fine-tuning. Furthermore, we introduce 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, we minimize a KL-divergence loss to match the probability distribution coming from the teacher and student models, as introduced in Yang et al. [198]. We omit SPLADEv2's hard negative mining step and only use in-batch negatives for SPLADE-X training.

5.2.2 SPLADE-X → BLADE

The architecture of BLADE is derived from its monolingual counterpart, SPLADEv2 [55], and its cross-language variant, SPLADE-X. We preserve many of the modeling choices from SPLADE-X, but we modify a) vocabulary selection; and b) intermediate pretraining.

5.2.2.1 Vocabulary Selection

In the original SPLADE, the output vocabulary was the same as that of the monolingual BERT language model. In the case of SPLADE-X, we restricted the vocabulary space of mBERT only to include query language terms for query and document expansion. For BLADE, we opt instead for a pruned bilingual language model [1], $\text{mBERT}_{\text{en-xx}}$. This choice offers two advantages. First, the bilingual model consists of a pruned mBERT vocabulary corresponding to the subword terms in both the query and document languages, i.e., $V_{\mathcal{T}} = V_{\mathcal{Q}} \cup V_{\mathcal{D}}$. This allows for bilingual term expansion as the model can choose related

terms in either language for both queries and documents. Second, the bilingual model contains only the embeddings corresponding to the pruned vocabulary of query and document languages; all the remaining embeddings corresponding to subwords from other languages are discarded. This reduces the model's size, as most parameters of PLMs are stored in the input/output embedding matrix. Across the six document languages we use for evaluation in this chapter, 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, this amounts to a reduction of the total training time by 30%. With a batch size of 64 on one V100 GPU, the reduction in inference time of the BLADE model averages 55%.

5.2.3 Intermediate Pretraining

In CLIR, the vocabulary mismatch between queries and documents poses a significant challenge for multilingual PLMs. To match terms in different languages, MPLMs need to generate similar representations of words with the same meaning across languages. To address this, ColBERT-X and SPLADE-X use *translate-train* to learn cross-language term associations using translated mMARCO pairs. However, this approach is not without its limitations. Specifically, machine-generated translations can introduce a phenomenon known as *translationese* [182] that has been shown to affect cross-language transfer performance due to translation artifacts [7]. Using a translate-train approach, the model learns term associations exclusively from translated document texts rather than from what would have been their natural written forms. To address this limitation, we propose an interme-

diate pretraining step that uses aligned text pairs in the query and document languages, expressed in a more natural writing style. We investigate two sources of aligned text: (1) parallel texts, which are direct translations of one another; and (2) comparable texts, which convey similar meanings but may not be direct translations. We describe these sources in detail in the experimental setup in Section 5.3.

Consider a list of aligned text pairs $[(P_1^{\mathcal{Q}}, P_1^{\mathcal{D}}), (P_2^{\mathcal{Q}}, P_2^{\mathcal{D}}), \ldots, (P_n^{\mathcal{Q}}, P_n^{\mathcal{D}})]$ in languages $\mathcal{Q}$ and $\mathcal{D}$. We compute the contrastive ranking loss similarly to Eq. (5.4). Treating $P_i^{\mathcal{Q}}$ as the query, $P_i^{\mathcal{D}}$ as the relevant document, and a set of in-batch documents $P_j^{\mathcal{D}}$ that we treat as non-relevant to the query, we model the loss as:

$$\mathcal{L}_{\text{CO}}^{\mathcal{QD}} = -\log \frac{e^{s(P_i^{\mathcal{Q}}, P_i^{\mathcal{D}})}}{e^{s(P_i^{\mathcal{Q}}, P_i^{\mathcal{D}})} + \sum_j e^{s(P_i^{\mathcal{Q}}, P_j^{\mathcal{D}})}} \tag{5.5}$$

The similarity score s is computed using Eq. (5.1). 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 $\mathcal{L}_{\text{WWM}}^{\mathcal{Q}}$ and $\mathcal{L}_{\text{WWM}}^{\mathcal{D}}$, respectively. WWM masks all subwords for a given word, in contrast to the common choice that only masks subwords which sometimes are only part of a whole

word.[2] Our overall pretraining loss $\mathcal{L}_{\text{pretrain}}$ is:

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{CO}}^{\mathcal{QD}} + \mathcal{L}_{\text{WWM}}^{\mathcal{Q}} + \mathcal{L}_{\text{WWM}}^{\mathcal{D}} \tag{5.6}$$

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.[3]

5.2.4 Connection to PSQ

Given English as the query language, SPLADE-X takes a sequence of terms from a non-English document as input and outputs a corresponding set of term weights for (only) English terms. In contrast, BLADE would output terms in both English and non-English languages. SPLADE-X exhibits the same behavior we would expect from PSQ, which resembles a statistical machine translation system that lacks a language model for the generated English. Section 2.3.1 shows how PSQ can be used to generate partial term counts in English from a non-English document by mapping term frequency vectors from non-English to English using a matrix of translation probabilities. This results in a sparse document representation, which contains nonzero term weights only for plausible translations of terms that appear in the document. Because SPLADE-X and PSQ models generate conformal representations, we can experiment with either early fusion, where

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

[3]We also tried updating all the model's parameters, but that did not provide any downstream effectiveness gain.

the term weights in the query language from different techniques can be combined before retrieval [126] or late fusion, in which we combine ranked retrieval results. However, early fusion presents a challenge because both techniques must agree on the target vocabulary tokenization beforehand to generate a sparse representation in the same output space. Therefore, in this chapter, we experiment with a late fusion technique, Reciprocal Rank Fusion (RRF) [41], to combine retrieval results from SPLADE-X and BLADE with PSQ. We find that the sparse neural CLIR approaches yield results complementary to those obtained using PSQ.

5.3 Experiments

In this section, we describe our experimental setup.

5.3.1 Test Collections & Evaluation

In our experiments, we utilize test collections from the CLEF 2003 multilingual ad-hoc retrieval track [28] for documents in French (FR), Italian (IT), German (DE), and Spanish (ES), and from the TREC 2022 NeuCLIR track[4] for documents in Chinese (ZH)[5] and Russian (RU). In every case, we use the English title field as the query, which produces queries with lengths typical of a Web search. Table 5.1 provides collection statistics. To evaluate effectiveness, we focus on Mean Average Precision (MAP) and Recall@100 (R@100). For significance testing, we use a two-tailed paired t-test ($p < 0.05$) with Holm-Bonferroni multiple test correction for the difference in means.

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

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

Table 5.1: Test collection statistics. Queries are in English with at least one relevant doc, Passages are as split for BLADE.

	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

5.3.2 Parallel and Comparable Corpora

For BLADE intermediate pretraining, we explore parallel and comparable texts from publicly available sources. For sentence-aligned parallel text, we use a diverse range of OPUS [176] corpora, including from EuroParl [92], GlobalVoices,[6] MultiUN [53], News-Commentary,[7] QED [2], TED [153], UNPC [211], and WMT-News [13].

Prior work has primarily used parallel corpora of aligned sentences to train MT systems. However, using only the limited context present in aligned sentences to fine-tune a PLM may be suboptimal for CLIR. To test this hypothesis, we also created a new passage-aligned parallel corpus. For each source of bitext above, we obtain the original monolingual corpora in the query and document languages, along with the sentence-level 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

[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

(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 [173], 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 another 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. [197] for C3, for each language pair *en-xx*, we identify the highest ranked non-English passage 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 5.2 shows corpus statistics.

5.3.3 Implementation Details

We implement SPLADE-X and BLADE using the Tevatron toolkit [64], which is built on top of the HuggingFace Transformers [187] library. To initialize SPLADE-X, we employ an mBERT model and select the target vocabulary by tokenizing the MS MARCO corpus in English using the mBERT tokenizer.[9] We then selected only those subwords containing lowercase alphanumeric characters, resulting in a list of 33k unique subwords for SPLADE-X modeling. To initialize BLADE, we use a smaller bilingual language model, released

[9]If we had wanted to experiment with using non-English queries, we would have instead used the translated MS MARCO corpora.

Table 5.2: 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

by Geotrend,[10] 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 [22].[11] We perform 100,000 steps of fine-tuning with an effective batch size of 256 using 8 V100 GPUs and a learning rate of 1×10^{-5} with the Adam [90] optimizer. Our maximum query length is 32 tokens, and passage lengths are limited to 256 tokens. Our SPLADE-X and BLADE implementation differs from that described in the SPLADEv2 [55] in that we utilize in-batch negative samples for training rather than the noise contrastive estimation process for mining hard negative training examples. We observe the effect of this change is small (on the order of 2%) when comparing a monolingual SPLADE model trained without hard negatives with an off-the-shelf SPLADEv2 model.

For intermediate pretraining with BLADE, we use either parallel or comparable passages or parallel sentences, with the pretraining objective in Eq. (5.6). We pretrain the model for 200,000 steps with an effective batch size of 192 on 8 V100 GPUs and a learning rate of 1×10^{-5} using Adam. When pretraining, the English passage is encoded as the query segment, 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 l to 1% of the total vocabulary size of the corresponding

[10] An example EN-DE model: `https://huggingface.co/Geotrend/bert-base-en-de-cased`
[11] `https://huggingface.co/datasets/unicamp-dl/mmarco/tree/main/data/google`

Geotrend bilingual model. Thus, the number of dimensions in the output vectors ranges from 330 to 380 for the six evaluation languages. 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-l passage term weights generated by the BLADE model. We then perform retrieval using the indexed passages and queries generated by a SPLADE-X or a BLADE model to generate a ranked list of 10,000 passages. The final step uses MaxP [18, 44] score aggregation to generate the top-1000 ranked documents from the ranked list of passages.

5.3.4 PSQ baseline

We compare SPLADE-X and different variants of BLADE with a PSQ-HMM baseline, which is described in Section 2.3.1. To obtain the translation probabilities, we combine results from three alignment tools: GIZA++ [137], BerkeleyAligner [102], and Eflomal [138]. For each language pair, we train each aligner using parallel sentences from all sources listed in Section 5.3.2 except UNPC, and we also add bilingual Panlex dictionaries [86] to the training set.[12] We use the same preprocessing for the bilingual corpora as for the queries

[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.

and the document collections: lowercasing tokens, removing punctuation and normalizing diacritics. We exclude translation probabilities of less than 1×10^{-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. This is an indexing-time implementation of the query-time implementation proposed in the original PSQ paper [47].

Table 5.3: MAP and R@100 for different sparse CLIR models 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
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
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**
PSQ-HMM + SPLADE-X	0.481	0.806	0.381	**0.727**	0.426	0.694	0.413	0.672	0.303	**0.540**	0.316	**0.520**	0.387	0.660
PSQ-HMM + BLADE-C	**0.492**	**0.826**	**0.397**	**0.727**	**0.446**	**0.713**	**0.440**	**0.698**	**0.306**	0.539	**0.328**	0.510	**0.402**	**0.669**

5.4 Retrieval Effectiveness of Sparse Bi-Encoders

We evaluate the retrieval effectiveness of sparse bi-encoder using results from Table 5.3. We start by first comparing the sparse neural models SPLADE-X and vanilla BLADE. Vanilla BLADE, which lacks intermediate pretraining, is fine-tuned only on the task-specific loss, but it differs from SPLADE-X in that it uses bilingual Geotrend embeddings rather than only query-language (English) embeddings in the output space. Overall, SPLADE-X has a higher MAP and R@100 compared to vanilla BLADE, on average, across all the 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 over all six languages) 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. 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 compared to the vanilla BLADE model. Compared to pretraining with comparable text pairs, MAP degrades without pretraining by 25% on average across the six languages. 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 than the other four. 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 [63], 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 [172] or using a regularization step [149]. 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.

We now compare the PSQ-HMM baselines with the sparse CLIR bi-encoders SPLADE-X and BLADE-C. First, we observe that PSQ performs better than SPLADE-X on average across the six collections. In contrast, 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. We further establish the complementary nature of the systems by focusing on the retrieval effectiveness of the combined systems. The ensemble of PSQ-HMM with SPLADE-X numerically improves the effectiveness over the individual base systems in all the languages for both measures. The differences are statistically significant except in French (R@100), Russian (MAP), and Italian (MAP and R@100). BLADE-C and PSQ-HMM are also clearly complementary, with statistically 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. Overall, combining BLADE-C with PSQ-HMM leads to numerically better retrieval effectiveness than combining SPLADE-X

with PSQ-HMM. The differences are statistically significant only in German for MAP and Spanish for R@100.

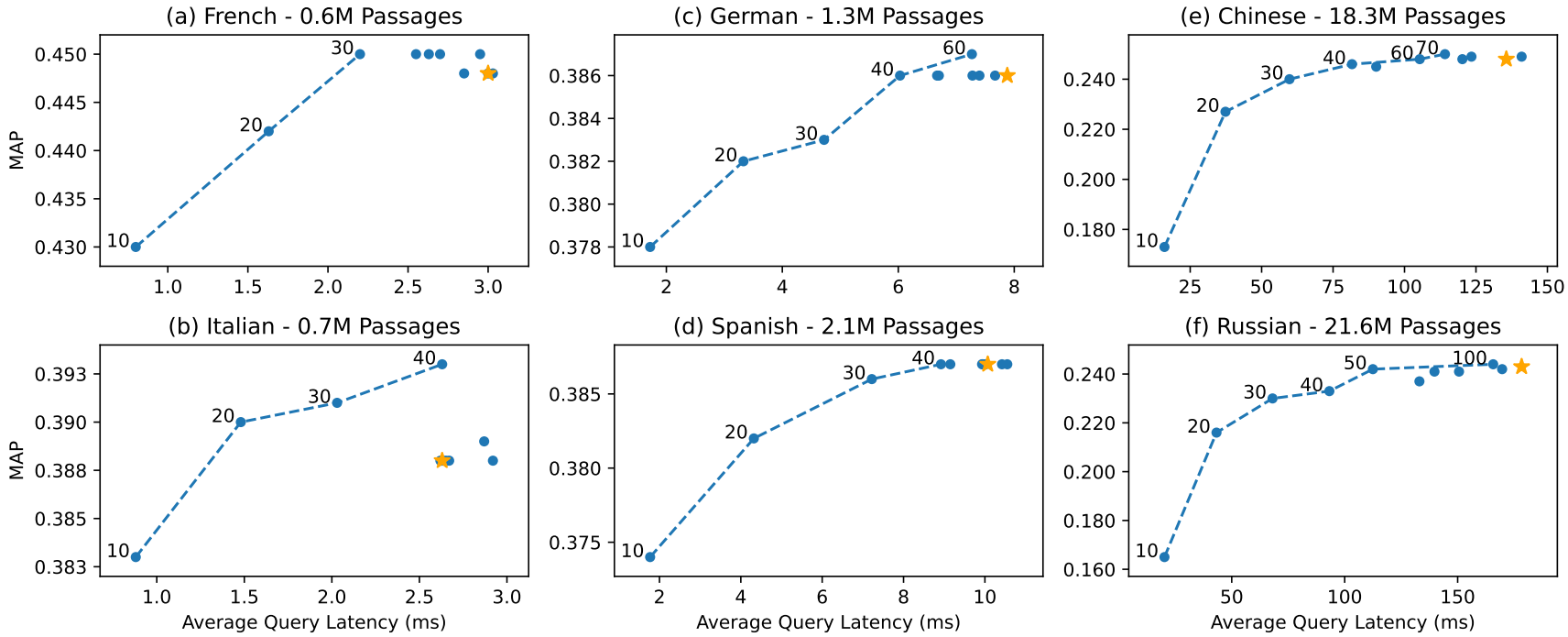


Figure 5.1: Average Query Latency vs MAP for BLADE-C model on the CLEF-03 and NeuCLIR collections using English queries with l ranging from 10..100 in intervals of 10. $\star$ denotes the BLADE-C model run with default l (1% of vocabulary size)

5.5 BLADE-C: Query Latency vs. Retrieval Effectiveness

In a lexical expansion framework such as BLADE, query latency is affected by the number of terms in the expanded query. The experiments above set l to 1% of the total vocabulary size, which ranges from 330 to 380. Such l values pose nearly no constraints on the number of non-zero tokens output by BLADE, which usually outputs less than 100 tokens. However, the number of resulting tokens directly affects the query latency. Enforcing a tighter constraint on the number of output tokens is a trade-off between the query latency and the retrieval effectiveness. We vary the value l from 10 to 100 and plot the Pareto frontier of average query latency (in milliseconds) and MAP in Figure 5.1.

We use PISA [116] on an AMD EPYC 7713 64-core processor with 256 GB of CPU RAM to measure the time to retrieve passages given the query set using the BLADE-C model.[13] We use PISA's multi-threaded processing with 32 threads to retrieve the top 10,000 passages for each query concurrently. Given our choice of large number of passages to retrieve, we use the MaxScore [179] dynamic pruning algorithm, as it has been shown to work well in such settings [117].

For French and Italian, with fewer than 1M passages each, we see query latency between 1 and 3 ms, while stronger sparsity constraints (i.e., smaller l) provide the best effectiveness and efficiency trade-off. Larger l values (the points without a number) are sometimes far from the Pareto frontier. For German and Spanish, with between 1M and 2M passages each, we have longer query latency, between 2 and 12 ms. Again, we can achieve almost the same MAP with lower values of l compared to the unconstrained case,

[13] 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.

i.e., l being 1% of the vocabulary. For the two large NeuCLIR test collections, Chinese and Russian, with between 18M and 21M passages each, we see considerably higher query latency, between 50 and 200 ms. For large collections, allowing more tokens to be output by BLADE (larger l) contributes more to effectiveness than for smaller collections. Larger collections can benefit from more distinguishing power between documents, so allowing more tokens benefits retrieval more than smaller collections. Furthermore, we can better tune l for a given collection size with a validation set. From these results, we can conclude that query latencies for BLADE can be tuned to be well within the range needed for interactive applications, literally faster than the blink of an eye, without adversely affecting the MAP values for BLADE-C reported above.

5.6 Chapter Summary

In this chapter, we introduce two sparse CLIR bi-encoder models, SPLADE-X and BLADE, that generate sparse vectors for retrieval. SPLADE-X is initialized with mBERT and employs a vocabulary reduction technique that restricts the multilingual vocabulary space to the terms corresponding to the query language only. In contrast, BLADE is initialized with a pruned bilingual PLM and allows the model to learn bilingual term expansions for both queries and documents. We additionally introduce an intermediate pretraining step using aligned text pairs to reduce the impact of translationese present in translated collections. Our experiments show that our model performs on par with a strong PSQ baseline on several CLIR test collections and, when combined, performs significantly better than the individual systems.

Chapter 6: Balancing Effectiveness and Efficiency for Scalable CLIR[1]

The ever-increasing volume of digital content has created a need for search systems that can scale efficiently while maintaining the quality of the retrieval outputs. To achieve this goal, such systems must focus on two key objectives, i) to be able to efficiently index large amounts of text that may be arriving in a streaming manner, and (ii) to retrieve relevant content from a large collection in a timely manner in response to a query. In this chapter, we operationalize these objectives as indexing latency and query latency, respectively. However, these objectives are not sufficient in isolation, as the ability of the system to produce high-quality retrieval outputs, as measured by the retrieval effectiveness, is also essential. As a result, each system must balance indexing and query latency with retrieval effectiveness. Depending on the specific application requirements, different systems may be more suitable. In this chapter, we identify the set of Pareto-optimal systems that offer the best balance between these contrasting objectives using the traditional and neural CLIR systems previously introduced in Chapters 4 and 5.

Throughout this dissertation, one of the recurring themes is the relationship between MT and the different stages of the retrieval process. The CLIR systems discussed so far can be categorized into three groups based on their use of MT: during the training, indexing, and

[1]This chapter contains content from: **Suraj Nair**, Eugene Yang, Dawn Lawrie, James Mayfield, and Douglas W Oard. "BLADE: Combining Vocabulary Pruning and Intermediate Pretraining for Scaleable Neural CLIR." In Preparation. [130]

querying phases. For instance, PSQ relies on intermediate MT outputs during its training phase to learn translation probabilities, while ColBERT-X, SPLADE-X, and BLADE use MT to generate training data in the translate-train setting. In contrast, some systems use MT during the indexing phase to generate translated documents, whereas others rely on MT during the querying phase to translate queries for retrieval purposes.

The evaluation of CLIR conducted so far involves retrieving content in a non-English language using a query expressed in English. As such, it is important to consider the direction of translation when using MT for different retrieval stages. Translating from non-English to English offers advantages for both MT and CLIR. Firstly, translating from a morphologically-rich language to a language with simpler morphology, like English, is less challenging than the reverse [15]. Additionally, the availability of large-scale training resources and several PLMs in English makes it easier to train neural IR systems that access English content. It is also worth noting that the effect of translation direction on CLIR is independent of whether we are translating queries or documents, as studies have shown that translation direction matters more than the unit of translation [119]. Given our focus on retrieving content in non-English, we additionally use traditional and neural CLIR systems with the same architecture as previously introduced in Chapters 4 and 5 but working with translated documents.

The remainder of this chapter is structured as follows. In Section 6.1, we list the test collections and the evaluation measures for performing the Pareto-optimality experiments. In Section 6.2, we list the description of each CLIR system that we use in our experiments. We show the results of retrieval effectiveness and indexing latency in Section 6.3, Section 6.4, and Section 6.5. Section 6.6 shows the Pareto-frontier of retrieval effectiveness and

Table 6.1: Test collection statistics. Queries are in 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	FA	ZH	RU
Queries	52	51	56	57	45	47	44
Documents	130K	158K	295K	454K	2,232K	3,179K	4,628K
Passages	0.5M	0.6M	1.3M	2.1M	12.6M	18.3M	21.6M
MT Passages	0.5M	0.6M	1.1M	1.7M	9.6M	13.7M	16.8M

query latency. Finally, we conclude the Chapter in Section 6.7.

6.1 Experimental Setup

In this section, we describe the test collections and the evaluation methods to analyze the retrieval effectiveness and the efficiency of the indexing and querying phases.

6.1.1 Test Collections

We not only reuse the test collections from Section 5.3 in the previous Chapter 5 but also evaluate our CLIR systems on Persian documents from TREC NeuCLIR 2022. As with the previous chapters, we use the English title field as the query. Table 6.1 provides the updated collection statistics.

6.1.2 Evaluation

Following a similar approach in the previous chapters, we evaluate retrieval effectiveness by focusing on the Mean Average Precision (MAP) and Recall@100 (R@100) of the top-k documents returned by the CLIR model in response to a query. As before, we employ a paired two-tailed t-test ($p < 0.05$) with Holm-Bonferroni correction for multiple tests

when testing for significance in the difference in means. To measure efficiency during the indexing phase, we compute the indexing latency per document in milliseconds (ms), which we characterize as the combination of the time it requires to translate the documents using MT (if any), the time it takes to run the CLIR model to generate outputs, and the time it takes to index the generated output from CLIR models. To measure efficiency during the querying phase, we compute the query latency in milliseconds, which we characterize as the time it takes to return the top-k passages (or documents) in response to a query. It is worth noting that we exclude the time it takes to generate document rankings from passage rankings, as this process is executed in memory and is thus fast relative to retrieval. In the next section, we outline how we compute the indexing and query latencies for each system evaluated in this chapter.

6.2 System Description

In this dissertation, the CLIR systems that have been introduced can be broadly categorized into two groups, i) those that do not utilize MT during indexing, which we refer to as MT-free indexing, and ii) those that require MT during indexing, which we refer to as MT-based indexing. The MT model we use in this chapter is similar to the one used in Chapter 4, consisting of a 6-layer encoder/decoder transformer stack implemented using Sockeye 2 [49, 73] trained on publicly available sentence-aligned parallel text data. The model has a decoding speed of roughly 50 sentences/sec on a single V100 GPU with 32 GB VRAM, which amounts to 280 ms per document averaged across the seven document collections. For CPU-based indexing, we use 24 AMD EPYC 7713 2.0 GHz CPU cores (2

threads per core for a total of 48 threads) with 256 GB of RAM. In this section, we provide a description of each system, with a specific focus on the category it belongs to and how it impacts the indexing latency.

6.2.1 PSQ

PSQ is an MT-free indexing system that operates on English queries and documents in their native language. To implement this system, we use a vector of English term counts generated during the indexing phase from a vector of term counts in the document language, similar to the approach in Chapter 5, and build an index based on those English counts. To enable this indexing-time-based implementation, we use a custom Python implementation of sparse matrices from the SciPy [181] toolkit. For retrieval, we employ the HMM implementation of the PSQ framework, as described in Section 2.3.1. The process of estimating translation probabilities is the same as in the previous Chapter 5. We measure the indexing latency as the overall time it takes to process and store the sparse indexes, run on 32 threads in parallel.

6.2.2 BLADE

Similar to PSQ, BLADE is also an MT-free indexing system that processes English queries and documents in their native language. We choose the BLADE-C model introduced in Chapter 5 due to its intermediate pre-training on the comparable corpora CLIRMatrix, which resulted in higher retrieval effectiveness compared to other models within the same family. In this chapter, we refer to the BLADE-C variant as BLADE. We keep intact

the design choice of BLADE from the previous chapter, where we split the native language documents into overlapping passages of 256 BPE tokens produced by the bilingual Geotrend tokenizer with a stride of 128 tokens. BLADE's indexing latency is computed by summing the inference time per document to run the BLADE model on a single V100 GPU, with the indexing time taken by Anserini using 48 threads. We then perform retrieval using the indexed passages and queries generated by the BLADE model to generate a ranked list of 10,000 passages. The final step uses MaxP [18, 44] score aggregation to generate the top-1000 ranked documents from the ranked list of passages.

6.2.3 PLAID-X

Two subsequent works, ColBERTv2 [162] and PLAID [163], addresses different issues following ColBERT. ColBERTv2 aims to tackle the indexing space footprint problem by using a residual compression approach [12], which has been previously applied to the ANN search techniques [35, 79]. The authors observe that the embedding space of ColBERT functions as a semantic space at the term level, allowing the term embeddings to be clustered and represented using their residual vectors with fewer bits compared to storing the entire vector. These cluster centroids can be viewed as an index with multiple terms or their corresponding passages. During the querying phase, ColBERTv2 generates an ordered set of passages by summing the dot product between the cluster centroids and the query terms. The late interaction step involves reconstructing the original term embeddings by finding the nearest centroid and its residual vector, also called "decompression," and using the MaxSim heuristic to rerank the documents. However, this leads to a bottleneck where

most of the query latency is spent on locating the nearest centroid and decompressing the embeddings. PLAID [163] focuses on centroid interaction and pruning steps to address this issue. It builds upon the ColBERTv2 approach by first filtering the centroids with low scores. As part of centroid interaction, PLAID introduces an approximate MaxSim operation that is computed using the passage centroid embeddings and the query term vectors. The final late-interaction step is the same as in ColBERTv2, which now scores fewer passages than before.

In this chapter, we introduce PLAID-X, which is a generalization of the PLAID framework designed to handle English queries with native language documents. Unlike in the monolingual setting, we do not train a v2 checkpoint and instead use the ColBERT-X model checkpoint directly with PLAID. Given the similarity of the architecture and training process of ColBERT and ColBERT-X, we hypothesize that the embedding space of ColBERT-X should also exhibit semantic properties. We confirm this hypothesis by comparing the retrieval effectiveness of ColBERT-X with PLAID-X in the seven languages and observe numerical improvements in MAP (from 0.370 to 0.378) and in R@10 (from 0.624 to 0.634). As in Chapter 4, we use a ColBERT-X model initialized with an XLM-R Large [40] multilingual encoder. We adopt a translate-train approach to fine-tune the model, using mMARCO passage translations as generated by Bonifacio et al. [22] using a Marian MT model (referred to as Helsinki) for CLEF languages. For NeuCLIR languages, we use MS MARCO passage translations generated by a Sockeye2 MT model as detailed in Chapter 4. In both cases, the translated passages are paired with an untranslated English MS MARCO query. Following the ColBERT-X model, we split the documents in their native language into overlapping passages of 180 XLM-R Sentencepiece tokens with a

stride of 90 tokens. PLAID-X indexing is performed using 8 V100 GPUs, and we record the final per-GPU indexing latency by multiplying the total indexing latency by the number of GPUs (8). We generate a ranked list of top-10,000 passages and use MaxP [18, 44] score aggregation to generate the top-1000 ranked documents from the ranked list of passages, as done in BLADE.

6.2.4 DT-BM25

DT-BM25 is a non-neural CLIR system run in the query language, where queries are in English and documents are machine translations of the text, thus belonging to the category of MT-based indexing systems. We perform retrieval with the BM25 implementation [155] from the Anserini toolkit [199] with the default hyperparameters ($k_1 = 0.9$, $b = 0.4$). The indexing latency is computed by adding two factors: the time it takes to translate the documents and the indexing time using Anserini run on 48 threads.

6.2.5 DT-SPLADE

DT-SPLADE is a CLIR system built using the SPLADEv2 architecture which uses English queries to retrieve translated documents, thereby falling into the category of MT-based indexing systems. 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 with English queries and English passages. For a fair comparison with SPLADE-X and BLADE training, no hard negative mining step described in the original SPLADEv2 [55] 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.[2] Following the BLADE model, we split the translated documents into overlapping passages of 256 BPE tokens produced by the BERT tokenizer with a stride of 128 tokens. Indexing time for DT-SPLADE 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. We generate a ranked list of top-10,000 passages and use MaxP [18, 44] score aggregation to generate the top-1000 ranked documents from the ranked list of passages as done in BLADE.

6.2.6 DT-PLAID

We use PLAID from Section 6.2.3, which uses English queries to retrieve translated documents. We use a ColBERT model initialized with a BERT-base encoder[3] and train the model on English MS MARCO triples following the same hyperparameters as the original ColBERT model. PLAID indexing is performed using 8 V100 GPUs, and we record the final per-GPU indexing latency by multiplying the total indexing latency by the number of GPUs (8). We generate a ranked list of top-10,000 passages and use MaxP [18, 44] score aggregation to generate the top-1000 ranked documents from the ranked list of passages.

We additionally create model ensembles of the systems belonging to each of the two categories. As in previous chapters, we use a late-fusion technique, Reciprocal Rank Fusion (RRF) [41], for combining the retrieval outputs from the different systems. Because system combination has implications for both effectiveness and efficiency, this allows us to explore

[2] https://github.com/naver/splade/tree/main/weights/distilsplade_max
[3] https://huggingface.co/bert-base-uncased

a broader range of options in that trade space. For model ensembles, we report the overall indexing latency by summing the per-document indexing latencies of individual systems.

Table 6.2: MAP and R@100 for retrieving content in 7 languages using English title queries.

	CLEF 03								NeuCLIR 2022						Average		Indexing Latency
	French		Italian		German		Spanish		Persian		Chinese		Russian				
Systems	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	MAP	R@100	per doc (ms)
Human Translation Monolingual Baseline																	
BM25	0.406	0.737	0.387	0.720	0.296	0.485	0.431	0.695	0.193	0.421	0.183	0.431	0.281	0.428	0.311	0.560	0.300
MT-free CLIR Indexing																	
PSQ	0.419	0.772	0.325	0.632	0.379	0.624	0.374	0.606	0.213	0.500	0.236	0.465	0.253	0.447	0.314	0.578	0.410
BLADE	0.448	0.783	0.389	0.730	0.386	0.634	0.387	0.640	0.225	0.495	0.248	0.453	0.243	0.429	0.332	0.595	43.37
+ PSQ	0.492	0.826	0.397	0.727	0.446	0.713	0.440	0.698	0.292	0.574	0.306	0.539	0.328	0.510	0.386	0.655	43.78
PLAID-X	0.458	0.753	0.411	0.730	0.421	0.652	0.393	0.624	0.288	0.583	0.329	0.548	0.349	0.548	0.378	0.634	61.10
+ PSQ	0.490	0.815	0.417	0.737	0.465	0.721	0.441	0.705	0.305	0.634	0.364	0.603	0.372	0.578	0.408	0.685	61.51
+ BLADE + PSQ	**0.511**	**0.830**	**0.440**	**0.789**	**0.484**	**0.747**	**0.469**	**0.724**	**0.321**	**0.644**	**0.379**	**0.608**	**0.386**	**0.590**	**0.427**	**0.705**	104.88
MT-based CLIR Indexing																	
DT-BM25	0.446	0.772	0.421	0.725	0.465	0.702	0.425	0.650	0.220	0.493	0.266	0.469	0.269	0.442	0.359	0.608	280.90
DT-SPLADE	0.486	0.846	0.418	0.731	0.476	0.756	0.448	0.670	0.273	0.584	0.310	0.576	0.353	0.552	0.395	0.674	313.87
+ DT-BM25	0.501	0.878	0.426	0.780	0.490	0.791	0.475	0.723	0.289	0.583	0.322	0.566	0.349	0.569	0.407	0.699	314.11
DT-PLAID	0.512	0.852	0.429	0.755	0.480	0.741	0.439	0.688	0.269	0.587	0.328	0.553	0.332	0.564	0.398	0.677	313.73
+ DT-BM25	0.529	0.872	**0.451**	**0.805**	0.523	0.794	0.474	0.722	0.288	0.597	0.332	0.568	0.352	0.566	0.421	0.703	346.94
+ DT-SPLADE + DT-BM25	**0.539**	**0.876**	0.438	0.803	**0.527**	**0.812**	**0.492**	**0.743**	**0.317**	**0.626**	**0.357**	**0.581**	**0.403**	**0.603**	**0.439**	**0.721**	347.18

6.3 Optimizing for Retrieval Effectiveness

Table 6.2 shows MAP and R@100 for different methods across the seven language pairs. We first focus on the MT-free indexing individual systems. Among the two non-neural CLIR systems, we see PSQ performing comparably to the human translation monolingual BM25 baseline on average, with the average MAP and the average R@100 slightly favoring PSQ-HMM across all seven languages. These broadly comparable results demonstrate that our PSQ 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. The next best individual system, BLADE, performs better than PSQ on average across both MAP and R@100, as already described in the previous chapter 5. PLAID-X achieves the highest effectiveness among the MT-free indexing group of individual systems. With just one exception (R@100 for French), PLAID-X consistently numerically outperforms the PSQ-HMM baseline by both MAP and R@100. These improvements are significant (by both measures) only in Italian among the CLEF languages and all NeuCLIR languages. Similarly, PLAID-X numerically outperforms BLADE in every NeuCLIR collection and performs slightly better in some CLEF languages with exceptions (R@100 for French and Spanish). Differences between PLAID-X and BLADE are significant (by each measure) only for 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 can be achieved by the dot product similarity between

a single query vector and a single document vector in the BLADE model.

Looking at the model ensembles of the MT-free indexing systems, it is clear that substantial improvements can be made over the effectiveness of any of the constituent systems. We already established the complementary nature of BLADE and PSQ in the previous Chapter 5, with significant improvements over the base systems (except in Italian) for both measures. The lower effectiveness of PSQ compared to BLADE in Italian contributes to lower gains after ensembling. Similarly, combining PLAID-X with PSQ leads to numerical improvements, although the differences are significant in German and Spanish for MAP and six of the seven languages (except Italian) for R@100. In all of the cases, a three-system ensemble of PLAID-X, BLADE, and PSQ has the highest effectiveness by both measures on average across all the collections. The base systems, while belonging to different modeling families, differ in terms of the training resources used (parallel sentences for PSQ, comparable passages for BLADE, and translated retrieval collections for BLADE and PLAID-X) and the language models employed (bilingual Geotrend for BLADE and XLM-R for PLAID-X). This diversity ultimately enhances the effectiveness of the ensemble results obtained from these systems.

We now focus on MT-based indexing systems that apply MT to every document. While this approach is computationally expensive, it has 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. First, when comparing the non-neural DT-BM25 with the non-neural PSQ system, we observe higher effectiveness using DT-BM25 for both measures on average across all languages. However, the differences are significant only for Italian and German for both measures and for Spanish (R@100 only). Notably, the numer-

ical differences are higher in CLEF collections than in NeuCLIR collections, signifying a degradation in the quality of MT between these collections. 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 is only significant for French, Chinese, and Russian. The clear advantage of DT-SPLADE results from the lexical expansion for 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 BLADE model, we see that DT-SPLADE achieves numerically higher MAP and R@100 in every language, although that improvement comes at a large indexing time cost. These differences are significant only in three languages (German, Spanish, and Russian) for MAP and in four languages (French, German, Chinese, and Russian) for R@100. 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 DT-PLAID, which has higher effectiveness on average than DT-BM25, with significant differences in four languages for both measures (except Italian, German, and Spanish). However, DT-PLAID is indistinguishable from DT-SPLADE, with no significant differences in any language.

Similar to the model ensembles in MT-free indexing, we observe similar gains in effectiveness with ensembles in MT-based indexing systems, although with fewer relative gains over the individual base system. An ensemble of DT-SPLADE with DT-BM25 has modest gains in MAP on average across the seven languages, but it sometimes helps (and,

on average, never hurts) R@100 for any language. Only the gains in MAP and R@100 in Spanish are significant with both base systems. We observe a similar case with the ensemble of DT-PLAID with DT-BM25, where only the improvements over the base systems in German and Spanish are significant for both MAP and R@100. The three-system ensemble has the highest effectiveness on average across the seven languages, with significant improvements over the base systems in five languages (except French and Italian) for MAP and three languages (German, Spanish, and Russian) for R@100. Comparing the three-system ensembles between the two indexing categories, MT-based and MT-free, we see higher effectiveness on average with the MT-based indexing ensemble than with the MT-free indexing ensemble. However, the differences are significant only in German for both MAP and R@100 and additionally in French for R@100. In short, the three-system MT-free indexing ensemble achieves roughly 97.3% MAP and 97.8% R@100 of the corresponding three-system MT-based indexing ensemble.

Karen Spärck Jones [170] proposed that differences in MAP scores of 0.05 could be considered *noticeable*, while differences of 0.1 could be regarded as *material*. Using this as an additional criterion for evaluating experimental results, we find noticeable differences between PSQ and BLADE only in Italian. In contrast, we identify more instances of noticeable differences between PSQ and PLAID-X in Italian, Persian, Chinese, and Russian. However, none of these instances exhibit material differences in MAP scores. When comparing the three-system MT-free indexing ensemble with the individual base systems, we observe material differences relative to PSQ in almost all languages, except for Italian, where the difference is noticeable. In comparison to BLADE, the differences are noticeable across all languages, with material differences only in Chinese and Russian. The

three-system ensemble displays fewer noticeable differences when compared to PLAID-X, limited to French, Spanish, Chinese, and German, and no material differences. On average, we observe only noticeable differences in MAP scores when comparing the three-system ensemble with PSQ and BLADE. Similarly, for the MT-based three-system ensemble, the average differences in MAP scores are noticeable solely in comparison to DT-BM25.

6.4 Optimizing for Indexing Latency

In this section, we quantify the differences in the efficiency of systems during the indexing phase. We operationalize efficiency during the indexing phase as indexing latency, the time (in milliseconds) to perform any necessary translation, run any needed model inference, and index the documents. PSQ has the lowest indexing latency 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). BLADE is the next fastest method, averaging faster than PLAID-X by a factor of 1.4. PLAID-X uses a larger multilingual XLM-R encoder than a bilingual model employed by BLADE. In addition, PLAID-X has a clustering step, further contributing to its higher indexing time.

BLADE and PLAID-X both have considerably lower indexing latencies than MT-based DT-SPLADE and MT-based DT-PLAID by a factor of 7.2 on average for BLADE and 5.1 on average for PLAID-X. This is primarily because the DT-SPLADE's and DT-PLAID's indexing latency includes three costs: a) translating the documents to the query language, b) running the monolingual model on the translated texts, and c) indexing the

Systems	Document Translation Time	Model Inference Time	Index Generation Time	Total Indexing Latency
PSQ	-	-	0.41	0.41
BLADE	-	40.76	2.70	43.37
PLAID-X	-	43.76	17.34	61.10
DT-BM25	280.66	-	0.24	280.90
DT-SPLADE	280.66	31.55	1.66	313.87
DT-PLAID	280.66	22.12	10.96	313.73

Table 6.3: Breakdown of the indexing latency for different CLIR systems averaged across the seven collections.

generated vectors. The average indexing latency for the DT-SPLADE is slightly higher than that of DT-PLAID, even when both models use the same monolingual BERT encoder. We attribute that to the difference in maximum input sequence length for the two models. The maximum sequence length for DT-PLAID is 180, as opposed to 256 for DT-SPLADE. 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 6.1 shows, the NeuCLIR collections are an order of magnitude larger than a typical CLEF collection.

Regarding the model ensembles, the indexing times are additive since each model has a different index. However, the translation time is added only once, as it is a one-time operation. In the case of MT-free indexing systems, model ensembles with PSQ result in virtually no gain in indexing latency due to the already low indexing latency of PSQ. This is also true for MT-based indexing systems that use DT-BM25. Comparing the MT-free and MT-based indexing model ensembles, our ensemble of three systems (PLAID-X, BLADE, and PSQ) for MT-free indexing is, on average, 3.3 times faster than the corresponding ensemble (DT-PLAID, DT-SPLADE, DT-BM25) for MT-based indexing.

Table 6.3 breaks down the total indexing latency for different CLIR systems based on Translation Time, Model Inference Time, and Index Generation Time. For MT-free indexing systems like BLADE and PLAID-X, most of the time is spent generating outputs,

as recorded in the inference time, as compared to indexing the generated outputs. The same applies to MT-based indexing systems like DT-SPLADE and DT-PLAID. Moreover, the model inference time for MT-based indexing systems is generally lower than their MT-free counterparts because they use a monolingual BERT model with fewer parameters compared to bilingual and multilingual models. However, for MT-based indexing systems, document translation time is a significant component of the total indexing latency compared to model inference and index generation time.

6.5 Balancing Retrieval Effectiveness and Indexing Latency

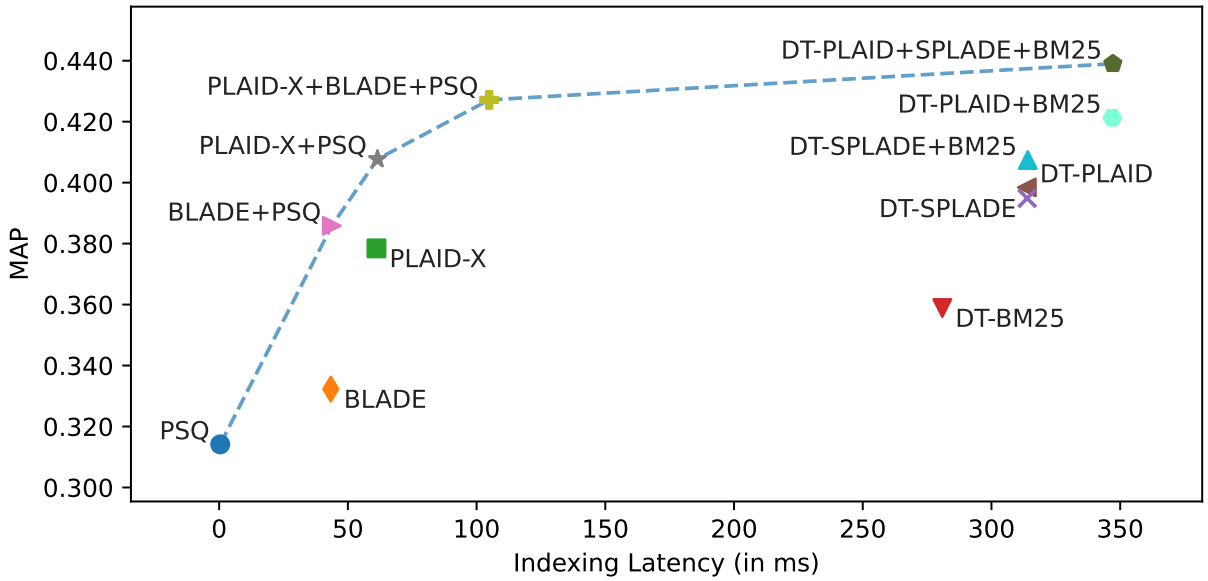


Figure 6.1: Illustrating the tradeoff between MAP and indexing times, averaged over six CLIR collections, using English title queries.

In this section, we illustrate the tradeoff between the efficiency during the indexing phase measured as the retrieval effectiveness and indexing latency, measured using MAP.[4] Figure 6.2 shows this tradeoff for each language, and Figure 6.1 summarizes those plots

[4]The use of R@100 yields similar results, and hence we do not include those plots.

using averages across all seven languages. The best outcome would be in the upper left corner of those figures, where the system achieves both low indexing latency and high effectiveness. Note that, in this analysis, we set a threshold of 2 ms for indexing latency and a threshold of 0.006 for MAP, in order to be considered as an improvement for Pareto-optimality analysis. This is done to prevent the emergence of multiple Pareto-optimal systems with only minor differences between the measures.

As shown in Figure 6.2, PSQ, the two-system ensemble of BLADE and PSQ, and the three-system ensemble of PLAID-X, BLADE, and PSQ are all on the Pareto frontier for each of the seven languages. The two-system ensemble of PLAID-X with PSQ is on the frontier for five out of seven languages, except French and Spanish. It is to be noted that neither BLADE nor PLAID-X alone lies on the Pareto frontier due to the low indexing latencies of PSQ, which leads to improved effectiveness with the PSQ ensemble.

The most striking point is that none of the single systems alone is anywhere near the Pareto frontier in any language. Said another way, when indexing latency matters, an MT-based indexing system alone is never the best choice. The two-system ensemble of DT-PLAID and DT-BM25 is on the Pareto frontier for Italian only. A three-system ensemble of DT-PLAID, DT-SPLADE, and DT-BM25 fares relatively better as it is on the Pareto frontier for four languages. Most notably, with the exception of the three-system ensemble in Russian, no other MT-based indexing systems are near the Pareto frontier in the NeuCLIR collection. This highlights that as the collection sizes scale up, MT-based indexing systems tend not to be the Pareto-optimal choice for building retrieval systems.

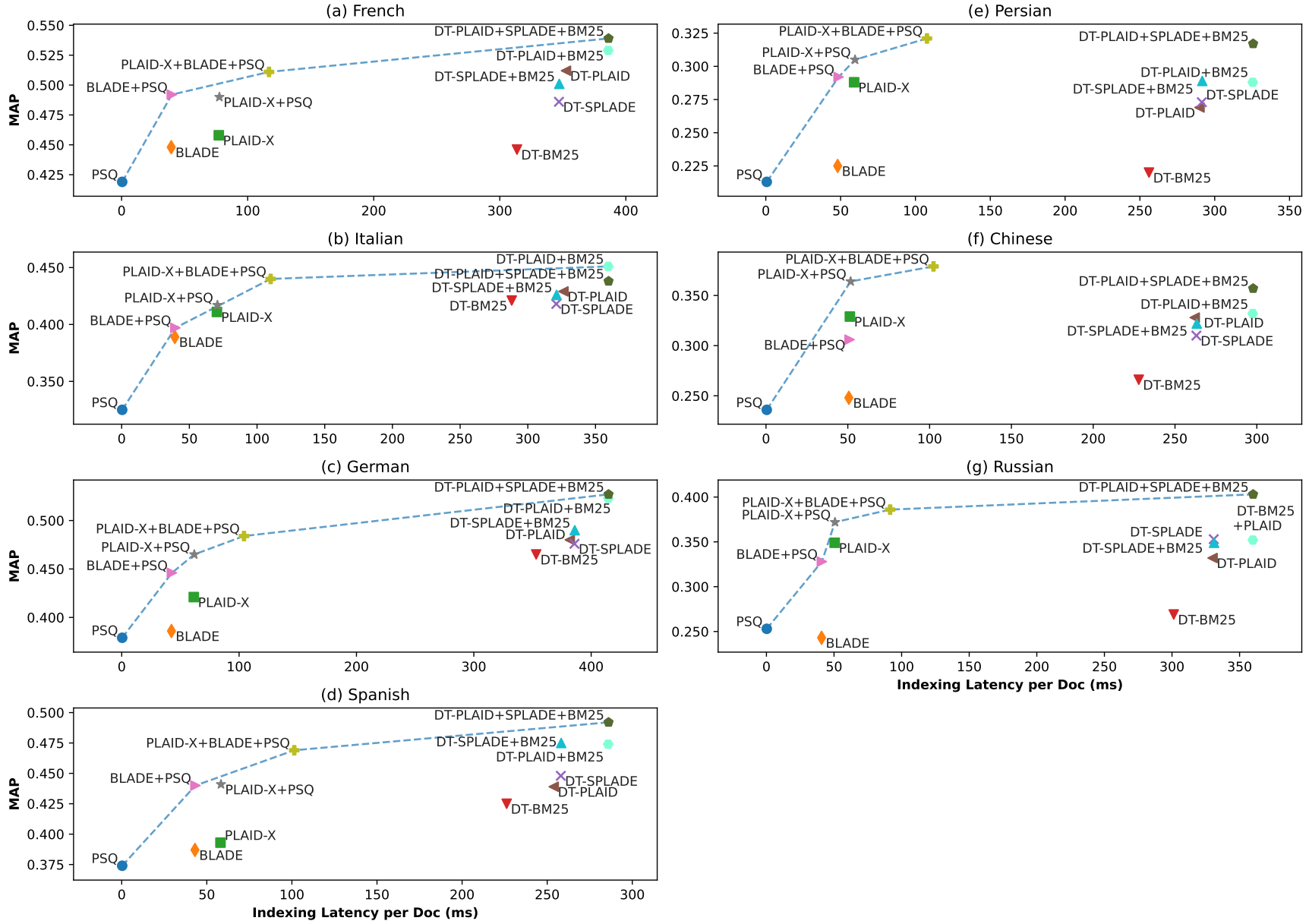


Figure 6.2: Indexing Latency vs. MAP for six collections using English queries.

6.6 Balancing Retrieval Effectiveness and Query Latency

In this section, we demonstrate the tradeoff between the efficiency at the querying phase, measured as query latency, and retrieval effectiveness, measured using R@100. To conduct this analysis, we make several design modifications to the systems used. Specifically, we compute the time it takes to retrieve the top-100 documents in response to a query for PSQ and DT-BM25, given that we chose R@100 as the effectiveness measure. For systems that retrieve passages, such as BLADE, PLAID-X, DT-SPLADE, and DT-PLAID, we return the top-1000 passages and use MaxP score aggregation to compute the ranked list of the top-100 documents. We use the PISA framework to run BLADE and DT-SPLADE similarly to the process described in Section 5.5 of the previous Chapter 5. Based on the findings in Section 5.5, we restrict the query expansion factor of BLADE and DT-SPLADE to 50 and 80 terms for CLEF and NeuCLIR collections, respectively. All systems, except PSQ, are run in a multi-threaded setting with 32 threads. We report query latency numbers for PLAID-X and DT-PLAID using the hyperparameters from the original PLAID paper. We run PLAID-X and DT-PLAID in two settings: PLAID-X-C/DT-PLAID-C, which runs on 32 threads, and PLAID-X-G/DT-PLAID-G, which runs on a single V100 GPU with 32 threads.[5]

Figure 6.3 presents the tradeoff between average query latency, measured in milliseconds, and R@100 for each system in every language. As in the previous section, the optimal outcome would be in the upper left corner of the figure, where a system achieves both low average query latency and high effectiveness. All systems on the Pareto frontier are MT-

[5]Similar to PLAID, we restrict the number of threads using `torch.set_num_threads`

based indexing systems that work with translated documents. Notably, DT-SPLADE falls on the Pareto frontier in all languages. DT-BM25 falls on the Pareto frontier for three languages, while DT-PLAID-G, run on a single GPU, falls on the Pareto frontier for four languages. For the NeuCLIR collections, all monolingual systems, except in Chinese, fall on the Pareto frontier.

Among the systems that work with documents in their native language, BLADE has the lowest average query latency compared to PSQ, PLAID-X-C, and PLAID-X-G. PLAID-X-C has the highest query latency for the CLEF collection, but this can be improved by transferring the computation to a GPU, as done in PLAID-X-G. The average query latency of PSQ increases rapidly as the collection size scales, particularly in large NeuCLIR collections. Our current implementation of the PSQ framework based on HMM operates on a single thread, and query latencies can be reduced by a multi-threaded implementation. However, none of these systems fall on the Pareto frontier. The number of passages in the collection for documents in their native language is higher than for translated documents, as shown in Table 6.1, with the ratio increasing as the collection size scales. Bilingual or multilingual models typically generate more subwords than monolingual models, which is the case with the monolingual systems used for translated documents, ultimately impacting the query latency.

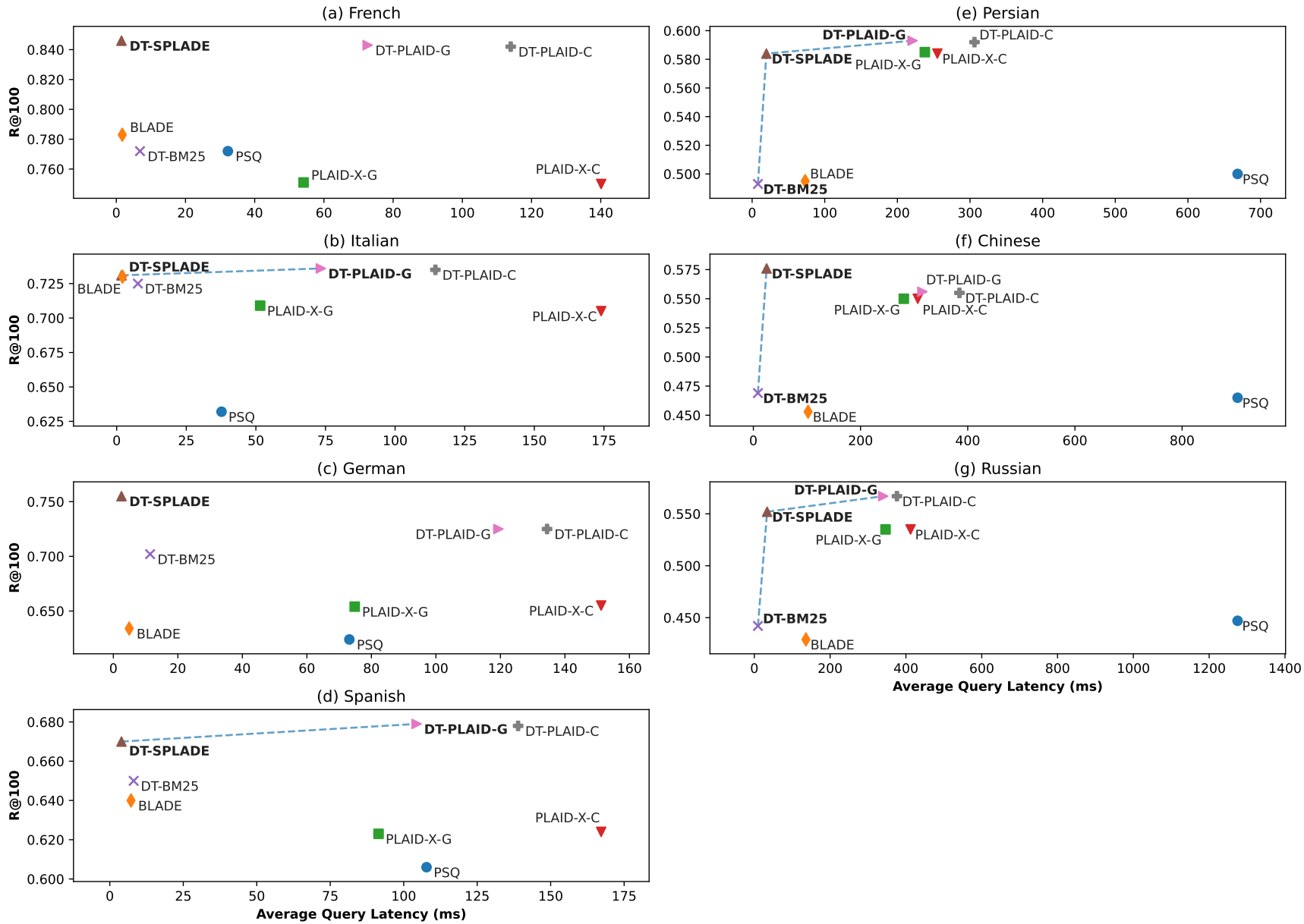


Figure 6.3: Average Query Latency vs. MAP for CLEF-03 and NeuCLIR collections using English queries. Systems in bold lie on the Pareto frontier.

6.7 Chapter Summary

This chapter focuses on identifying the set of Pareto-optimal CLIR systems that provide the best balance between the contrastive objectives of retrieval effectiveness and indexing latency. We categorize the CLIR systems into two groups: MT-free indexing systems, which work with documents in their native language (PSQ, BLADE, and PLAID-X), and MT-based indexing systems, which use translated documents generated by an MT model (DT-BM25, DT-SPLADE, and DT-PLAID). Our results show that the ensemble of MT-free indexing systems resides on the Pareto frontier of retrieval effectiveness and indexing latency as compared to the MT-based indexing systems. Additionally, the three-system ensemble of MT-free indexing systems achieves up to 97% effectiveness of the corresponding three-system ensemble of MT-based indexing systems while achieving lower indexing latencies by a factor of 3.3. We also identify the Pareto-optimal CLIR systems that balance retrieval effectiveness and query latency, with MT-based indexing systems falling on the Pareto frontier compared to MT-free indexing systems.

Chapter 7: Conclusion

CLIR systems aim to retrieve relevant content in a language that is distinct from the language of the query. This task requires the CLIR systems to match the meanings of similar terms that are expressed in two different languages. In this dissertation, we introduce neural CLIR systems capable of matching texts in two languages using the contextual representations resulting from the improvements in language modeling techniques. The long-standing goal of the IR community is to build retrieval systems that balance the tradeoff between the effectiveness and efficiency. This balance becomes particularly important when designing neural retrieval systems because the underlying contextual language models are often compute-intensive and resource-hungry. This dissertation focuses specifically on designing neural CLIR systems that find a Pareto-optimal balance between the complementary objectives of effectiveness (i.e., quality of retrieval output) and efficiency (e.g., indexing and query latency).

In Chapter 3, we focus on retrieve-and-rerank pipelines that incorporate an efficient first-stage CLIR system followed by a cross-encoder that employs an MPLM to process cross-language texts. To enable the cross-encoder to handle long documents during the querying phase, we introduce CREPE, a passage selection strategy that selects the best passage(s) from a document to score with a cross-encoder to address **RQ1**. CREPE accom-

plishes this by utilizing an efficient first-stage CLIR system to rank passages and selecting the top-k highest-scoring passage(s) as the effective representation for the document. We demonstrate that scoring the single-best CREPE with cross-encoder results in better retrieval effectiveness across all test collections compared to a systematic strategy like FirstP, which selects the first passage from the document to score. Additionally, we design hybrid strategies that combine CREPEs with FirstP, which achieve up to 99% of the effectiveness of the MaxP strategy that offers the best retrieval effectiveness but also the highest query latency among all the strategies. It is important to note that both the single-best CREPE and the hybrid strategies lie on the Pareto frontier of query latency and retrieval effectiveness (Contribution S1), highlighting the significance of our proposed strategy. Moreover, we create a passage-level training dataset using CREPEs to fine-tune cross-encoders as part of the training phase. We observe consistent improvements in retrieval effectiveness with the cross-encoder trained on the dataset created using CREPEs compared to the first-stage CLIR system and a cross-encoder trained with a systematic approach (Contribution S2).

In Chapter 4, we introduce ColBERT-X, a multi-representation bi-encoder that leverages an XLM-R encoder to process cross-language text and perform CLIR. We create two variants of ColBERT-X, zero-shot, and translate-train, trained on the original MS MARCO corpus and translations of MS MARCO passages in document languages paired with English queries, respectively (Contribution S3). During the querying phase, the zero-shot variant takes a translated query as input, while the translate-train variant processes the original query expressed in its natural form. Both variants process documents in their original language. Therefore, the primary difference between the two lies in the application of MT; in zero-shot, MT is applied during the querying phase, while in the translate-train

setting, it is applied during the training phase. We demonstrate that using either variant of ColBERT-X to perform CLIR leads to improvements in retrieval effectiveness over traditional CLIR baselines. However, on average, the translate-train variant outperforms the zero-shot variant, thus addressing **RQ2**. We further enhance the retrieval effectiveness of ColBERT-X by applying pseudo-relevance feedback to achieve cross-language term expansion in the embedding space (Contribution S4). We analyze the effect of various MT models on both ColBERT-X variants and observe that improvements in the quality of the MT model, measured using BLEU, generally translate into better downstream CLIR task performance, measured using MAP.

In Chapter 5, we introduce SPLADE-X, our first single-representation bi-encoder that projects queries and documents into the sparse vocabulary space of MPLM. However, designing sparse CLIR models faces two issues that differ from monolingual IR models. These include multilingual models having larger vocabulary sizes than monolingual models, and the lack of supervision during multilingual pretraining to learn cross-language term associations. SPLADE-X addresses these challenges by restricting the vocabulary space to include terms from the query language only and leveraging a translate-train approach with MS MARCO to learn cross-language term associations. Our experiments reveal that SPLADE-X achieves comparable performance to a strong traditional CLIR baseline, PSQ. Building on the potential to enhance the design choices made by SPLADE-X, we introduce BLADE, our next sparse CLIR model (Contribution S5). BLADE brings two significant changes, first by switching to a pruned bilingual model to improve modeling efficiency and second, by introducing an intermediate pre-training step that utilizes aligned cross-language texts expressed in their natural forms as the training dataset to learn cross-

language term associations, which enhances retrieval effectiveness. We explore two primary sources of aligned text, parallel sentences, and comparable passages for intermediate pre-training. We further create a new dataset comprising parallel passages created from parallel sentences that we release for future research purposes (Contribution D1). Our experiments indicate that BLADE with intermediate pre-training consistently outperforms SPLADE-X and PSQ in terms of retrieval effectiveness on average across multiple test collections. We also demonstrate that the query expansion factor of BLADE can be optimized to achieve a Pareto-optimal balance between query latency and retrieval effectiveness.

As part of our objective to create highly scalable systems, we focus on CLIR systems that lie on the Pareto frontier of retrieval effectiveness and indexing latency in Chapter 6 to address **RQ3**. We focus on six CLIR systems, divided into two categories, i) MT-free indexing, working with documents in their native language (PLAID-X, BLADE, PSQ), and ii) MT-based indexing, which uses machine-translated documents (DT-PLAID, DT-SPLADE, and DT-BM25). We further create ensembles by combining the retrieval outputs of these individual systems using Reciprocal Rank Fusion. We demonstrate the complementary nature of BLADE and PLAID-X with the traditional PSQ baseline by showing effectiveness gains in ensembling them individually and together. Both BLADE and PLAID-X have lower indexing latencies than the MT-based indexing systems, which is primarily affected by the time required to translate documents to query language. Furthermore, we demonstrate that an ensemble of PLAID-X, BLADE, and PSQ lies on the Pareto frontier of retrieval effectiveness and indexing latency (Contribution S6). Additionally, we focus on identifying the CLIR systems that lie on the Pareto frontier of retrieval effectiveness and query latency in Chapter 6 to address **RQ4**. We find that MT-based indexing systems have

low query latencies compared to MT-free indexing systems, with DT-SPLADE offering the best tradeoff in every language. We release our implementations of PSQ, ColBERT-X, and BLADE, along with the PSQ translation tables in all the languages we worked with in this dissertation (Contribution C1, C2, C3, D2).

In conclusion, this dissertation highlights the interplay between MT and the various phases of the CLIR process. Although the connections between MT and CLIR indexing, as well as retrieval, have been extensively studied, this work highlights the effect of MT during the training of neural CLIR systems. We demonstrate that MT can be employed to create translated collections from large-scale training collections with relevance judgments in English, resulting in highly effective neural CLIR systems when compared to their non-neural counterparts. Furthermore, neural CLIR systems exhibit complementary behavior to non-neural CLIR systems in terms of training resources used (parallel/comparable/translated collection), underlying tokenization employed (whitespace-separated tokens/subwords), and the intermediate representation space (dense/sparse). This complementarity allows for the creation of highly effective MT-free indexing system ensembles that incorporate both non-neural and neural CLIR systems, exhibiting retrieval effectiveness comparable to MT-based indexing systems. For practitioners primarily concerned with indexing efficiency, particularly in applications that access streaming content, MT-free indexing systems that process documents in their native language present an appealing alternative. Meanwhile, those focused on retrieval effectiveness will find working with English translations advantageous for both MT and CLIR. This is attributed to the abundance of parallel texts for numerous language pairs, especially those involving English, which facilitates the creation of MT systems. In addition, translating from a morphologically-rich language to one with simpler

morphology, such as English, is less challenging than the reverse. Combining the availability of MT with large-scale training collections with relevance judgments and multiple PLMs in English facilitates the creation of neural CLIR systems that access English content. These crucial findings underscore the synergy between MT and CLIR across different phases, including training, indexing, and retrieval, and demonstrate the potential to create CLIR systems that offer a Pareto-optimal balance between retrieval effectiveness and indexing/querying efficiency.

7.1 Limitations

In this section, we list the limitations of our work

1. Our experiments focused on retrieving content in non-English languages using queries in English. To strengthen the claim of our proposed approaches, we would need to consider the general setting where the queries and documents could be in any two distinct languages.

2. All of our experiments included CLEF collections from early 2000-2003 which lack neural systems among the runs that contributed towards pooling. As a result, it is uncertain how reliably older CLEF collections can differentiate newer-generation neural retrieval systems.

3. Throughout our experiments, we used MT systems trained on parallel sentences to translate title queries, which are often expressed in short keywords. This mismatch in the domain of MT training data and the test queries on which MT is applied can be problematic. Additionally, MT systems segment long documents into individual

sentences and translate them independently. Such MT systems fail to handle the document context to produce better contextual translations.

4. Except in CREPE, we did not explore strategies to mine for harder negatives for ColBERT-X and BLADE. As such, we expect improvements to retrieval effectiveness in a similar vein as observed in monolingual applications.

5. For SPLADE-X and BLADE, we employ the subword tokens provided by the underlying multilingual or bilingual PLM. As a result, we inherit the challenges related to the underlying tokenization and the specific design choices made throughout the PLM development process. Notably, the ratio of subwords generated by a multilingual model is higher in comparison to that of a monolingual model.

6. The experiments evaluating efficiency in both indexing and querying phases largely rely on the specific implementation employed for producing translation and retrieval outputs. It is crucial to recognize that variations in translation and retrieval times may arise from employing different hardware configurations or toolkits, which could potentially influence the efficiency of the system being examined.

7.2 Future Work

In this section, we present an overview of the directions that can be explored as a result of our research in this dissertation.

7.2.1 Cross-language Query Expansion ⟷ Query Translation

Section 4.4 introduced a form of cross-language query expansion using the shared vector space of the ColBERT-X model to retrieve cross-language nearest neighbors for query terms. This approach has similarities with PSQ, which generates cross-language translation alternatives for query terms using translation probabilities learned from parallel sentences. Previous studies [32, 133] have also identified the connection between cross-language query expansion and query translation. While our current approach for finding cross-language expansion terms using ColBERT-X is restricted to the target retrieval collection, we can leverage large external corpora, particularly aligned texts such as parallel sentences or parallel/comparable passages, as explored in Chapter 5. This idea of using external corpora for expansion is not new, but it becomes particularly interesting in the context of CLIR due to the diverse approaches and datasets that can be used to find query expansions.

One approach to finding cross-language expansions is to replicate the ColBERT-X process using a large monolingual corpus in the document language. Another approach is to use a monolingual ColBERT model in the query language and aligned text, such as parallel sentences. For a given query, we can retrieve the highest-scoring sentence(s) from the parallel texts in the query language using the ColBERT model. From the retrieved sentences, we can find their aligned counterparts in the document language to generate contextual query expansion terms (or translation alternatives). Furthermore, we can build a contextual PSQ model by restricting the translation alternatives to only include terms occurring in the aligned texts of the retrieved sentences.

7.2.2 Knowledge Distillation for CLIR

Knowledge distillation has been proven effective in monolingual retrieval applications [55, 162] by transferring knowledge from an expressive teacher model, such as cross-encoders, to a weaker student model, such as bi-encoders. In Chapter 5, both SPLADE-X and BLADE models used SPLADE as the teacher model for knowledge distillation. However, the design choices in the CLIR are different compared to the monolingual setting. Several possible choices for teacher language models exist, including a monolingual model trained in either the query or document language or a multilingual teacher model. The choice of the dataset also varies based on the language. Furthermore, we can obtain complementary sources of knowledge stored in the individual teacher models, depending on the type of language model and the training dataset used. Once multiple teacher models are available, we can evaluate their performance based on the effectiveness of their student model.

7.2.3 CLIR Training Data using Large Language Models

Training neural CLIR system is particularly challenging due to the limited availability of large-scale collections that contain queries and documents in their native languages. In this dissertation, we explored various approaches to address this challenge. Chapter 4 investigated the use of machine translation (MT) to translate monolingual collections to create CLIR training data. However, such collections may suffer from translationese, and so in Chapter 5, we proposed an intermediate pretraining step that used aligned texts with documents expressed in their natural form. This begs the question of whether there are

other ways to acquire CLIR training data.

A possible approach to creating data is to utilize a model that can generate queries that are relevant to a given document. Such query generation models are available in the monolingual setting, and they have mainly been used as document expansion systems. Building on the concept presented in Section 7.3.1 utilizing aligned texts, we can employ a monolingual model to generate queries from a text in the query language and then match the generated queries with the corresponding aligned text in the document language. However, finding aligned texts that match the domain of the target document collection, such as news, legal and biomedical, can be challenging. An interesting option, however, is the use of large language models such as GPT-3 [29] and FLAN-T5 [112] XXL to generate queries, which is being actively explored in monolingual applications [21, 23, 46, 80]. The main advantage, in this case, is that we can use texts that match the domain of the target document collection we want to retrieve for generating synthetic queries.

7.3 Implications

"The test of our progress is not whether we add more to the abundance of those who have much; it is whether we provide enough for those who have too little." - Franklin D. Roosevelt [156]. In his second inaugural speech in January 1937, FDR spoke these words in the hope of providing basic necessities to the citizens of the United States, who were still reeling from the effects of the Great Depression. Today, almost nine decades later, these words still hold true, albeit in a different context. We now strive to reduce the digital divide and enable access to resources to more users worldwide. However, even if we managed to

reduce this divide, the digital language divide would still present a significant challenge, preventing many users from accessing resources in their own native language.[1] CLIR offers the potential to reduce this language divide by enabling users to find content in a language different from the one they are searching in. This dissertation introduces a set of CLIR systems that can access content in one language using a query expressed in another. As we aim to break the language barriers and provide access to content in any language, it is crucial to build scalable systems that can handle potentially large web-scale collections. This dissertation takes initial steps towards this goal by identifying a set of systems that balance the contrasting objectives of better scalability while maintaining retrieval quality.

[1] https://www.wired.com/story/internet-digital-language-divide/

Bibliography

[1] Amine Abdaoui, Camille Pradel, and Grégoire Sigel. Load what you need: Smaller versions of multilingual BERT. In *SustaiNLP / EMNLP*, 2020.

[2] Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 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)*, pages 1856–1862, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/877_Paper.pdf`.

[3] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. Cross-Domain modeling of Sentence-Level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3490–3496, Hong Kong, China, November 2019. Association for Computational Linguistics.

[4] James Allan, Jamie Callan, W Bruce Croft, Lisa Ballesteros, Donald Byrd, Russell Swan, and Jinxi Xu. INQUERY does battle with TREC-6. *NIST Special Publication 500-240*, pages 169–206, 1998.

[5] Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*, 2016.

[6] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1073. URL `https://aclanthology.org/P18-1073`.

[7] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.618. URL `https://aclanthology.org/2020.emnlp-main.618`.

[8] Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. XOR QA: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.46. URL https://aclanthology.org/2021.naacl-main.46.

[9] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

[11] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268v3*, 2018.

[12] Christopher F Barnes, Syed A Rizvi, and Nasser M Nasrabadi. Advances in residual vector quantization: A review. *IEEE transactions on image processing*, 5(2):226–262, 1996.

[13] 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. 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)*, pages 1–61, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL https://aclanthology.org/W19-5301.

[14] Joel Barry, Elizabeth Boschee, Marjorie Freedman, and Scott Miller. SEARCHER: Shared embedding architecture for effective retrieval. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 22–25, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-55-9. URL https://aclanthology.org/2020.clssts-1.4.

[15] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 46(1):1–52, 2020.

[16] Emily Bender. The #benderrule: On naming the languages we study and why it matters. *The Gradient*, 2019.

[17] Emily M. Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018. doi: 10.1162/tacl_a_00041. URL https://aclanthology.org/Q18-1041.

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

[19] Steven Bird. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1225403.1225421. URL `https://aclanthology.org/P06-4018`.

[20] Hamed Bonab, Sheikh Muhammad Sarwar, and James Allan. Training effective neural CLIR by bridging the translation gap. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 9–18. Association for Computing Machinery, New York, NY, USA, July 2020.

[21] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2387–2392, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531863. URL `https://doi.org/10.1145/3477495.3531863`.

[22] Luiz Henrique Bonifacio, Israel Campiotti, Roberto Lotufo, and Rodrigo Nogueira. mMARCO: A multilingual version of MS MARCO passage ranking dataset. *arXiv preprint arXiv:2108.13897*, 2021.

[23] Leonid Boytsov, Preksha Patel, Vivek Sourabh, Riddhi Nisar, Sayani Kundu, Ramya Ramanathan, and Eric Nyberg. InPars-Light: Cost-Effective Unsupervised Training of Efficient Rankers. *arXiv e-prints*, art. arXiv:2301.02998, January 2023. doi: 10.48550/arXiv.2301.02998.

[24] Martin Braschler. CLEF 2000 — overview of results. In *Cross-Language Information Retrieval and Evaluation*, pages 89–101. Springer Berlin Heidelberg, 2001.

[25] Martin Braschler. CLEF 2001 — overview of results. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems*, pages 9–26, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45691-9.

[26] Martin Braschler. CLEF 2002—overview of results. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 9–27. Springer, 2002.

[27] Martin Braschler. Combination approaches for multilingual text retrieval. *Information Retrieval*, 7:183–204, 01 2004.

[28] Martin Braschler and Carol Peters. CLEF 2003 methodology and metrics. In *Comparative Evaluation of Multilingual Information Access Systems*, pages 7–20. Springer Berlin Heidelberg, 2004.

[29] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[30] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 89–96, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102363. URL https://doi.org/10.1145/1102351.1102363.

[31] Christopher Burges, Robert Ragno, and Quoc Le. Learning to rank with nonsmooth cost functions. *Advances in Neural Information Processing Systems*, 19, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/af44c4c56f385c43f2529f9b1b018f6a-Paper.pdf.

[32] Guihong Cao, Jianfeng Gao, Jian-Yun Nie, and Jing Bai. Extending query translation to cross-language query expansion with markov chain models. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, page 351–360, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595938039. doi: 10.1145/1321440.1321491. URL https://doi.org/10.1145/1321440.1321491.

[33] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

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

[35] Yongjian Chen, Tao Guan, and Cheng Wang. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 10(12):11259–11273, 2010.

[36] Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. Cross-lingual natural language generation via pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7570–7577, Apr. 2020. doi: 10.1609/aaai.v34i05.6256. URL https://ojs.aaai.org/index.php/AAAI/article/view/6256.

[37] Eunseong Choi, Sunkyung Lee, Minijn Choi, Hyeseon Ko, Young-In Song, and Jongwuk Lee. 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*, pages 272–282, 2022.

[38] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. URL https://openreview.net/pdf?id=r1xMH1BtvB.

[39] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H196sainb.

[40] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics.

[41] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 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*, pages 758–759, 2009.

[42] Koby Crammer and Yoram Singer. Pranking with ranking. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/5531a5834816222280f20d1ef9e95f69-Paper.pdf.

[43] Giovanni Da San Martino, Salvatore Romeo, Alberto Barroón-Cedeño, Shafiq Joty, Lluís Maàrquez, Alessandro Moschitti, and Preslav Nakov. Cross-language question re-ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1145–1148, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350228.

[44] Zhuyun Dai and Jamie Callan. 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*, SIGIR'19, pages 985–988, New York, NY, USA, July 2019. Association for Computing Machinery.

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

[46] Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. Promptagator: Few-shot Dense Retrieval From 8 Examples. *arXiv e-prints*, art. arXiv:2209.11755, September 2022. doi: 10.48550/arXiv.2209.11755.

[47] Kareem Darwish and Douglas W Oard. Probabilistic structured query methods. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 338–344, 2003.

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[49] Tobias Domhan, Michael Denkowski, David Vilar, Xing Niu, Felix Hieber, and Kenneth Heafield. 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)*, pages 110–115, Virtual, October 2020. Association for Machine Translation in the Americas. URL https://aclanthology.org/2020.amta-research.10.

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

[51] Chris Dyer, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, 2013.

[52] David M. Eberhard, Gary F. Simons, and Charles D. Fennig. *Ethnologue: Languages of the World*. SIL International, Dallas, Texas, twenty-sixth edition, 2023. URL http://www.ethnologue.com. Online version.

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

[54] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-1049. URL https://aclanthology.org/E14-1049.

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

[56] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 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*, pages 2288–2292, 2021.

[57] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 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*, pages 2353–2359, 2022.

[58] Walter Fox. *Writing the news: A guide for print journalists*. Wiley-Blackwell, 2001.

[59] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4(Nov): 933–969, 2003.

[60] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987.

[61] George W. Furnas, Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, Richard A. Harshman, Lynn A. Streeter, and Karen E. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In Yves Chiaramella, editor, *SIGIR'88, Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Grenoble, France, June 13-15, 1988*, pages 465–480. ACM, 1988.

[62] Petra Galuščáková, Douglas W Oard, and Suraj Nair. Cross-language information retrieval. *arXiv preprint arXiv:2111.05988*, 2021.

[63] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 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, 2019. URL `https://openreview.net/forum?id=SkEYojRqtm`.

[64] Luyu Gao, Xueguang Ma, Jimmy J. Lin, and Jamie Callan. Tevatron: An efficient and flexible toolkit for dense retrieval. *arXiv preprint arxiv:2203.05765*, 2022.

[65] Julio Gonzalo and Douglas W. Oard. iCLEF 2004 track overview: Interactive cross-language question answering. In Francesca Borri, Carol Peters, and Nicola Ferro, editors, *Working Notes for CLEF 2004 Workshop co-located with the 8th European Conference on Digital Libraries (ECDL 2004), Bath, UK, September 15-17, 2004*, volume 1170 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[66] Stephan Gouws and Anders Søgaard. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390,

Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1157. URL https://aclanthology.org/N15-1157.

[67] Mauro F Guillén and Sandra L Suárez. Explaining the global digital divide: Economic, political and sociological drivers of cross-national internet use. *Social forces*, 84(2):681–708, 2005.

[68] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 55–64, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340731. doi: 10.1145/2983323.2983769. URL https://doi.org/10.1145/2983323.2983769.

[69] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1908.10396.

[70] Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. Better word alignments with supervised itg models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 923–931, 2009.

[71] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[72] Ralf Herbrich, Thore Graepel, and Klause Obermayer. Large Margin Rank Boundaries for Ordinal Regression. In *Advances in Large Margin Classifiers*, pages 115–132. The MIT Press, 1999. URL http://www.herbrich.me/papers/nips98_ordinal.pdf.

[73] Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. Sockeye 2: A toolkit for neural machine translation. In *EAMT 2020*, 2020. URL https://www.amazon.science/publications/sockeye-2-a-toolkit-for-neural-machine-translation.

[74] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666*, 2020.

[75] Yedid Hoshen and Lior Wolf. Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 469–478, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[76] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. Co-pacrr: A context-aware neural ir model for ad-hoc retrieval. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 279–287, New

York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355810. doi: 10.1145/3159652.3159689. URL https://doi.org/10.1145/3159652.3159689.

[77] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkxgnnNFvH.

[78] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 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)*, pages 1–10, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1001. URL https://aclanthology.org/P15-1001.

[79] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, January 2011. doi: 10.1109/TPAMI.2010.57. URL https://hal.inria.fr/inria-00514462.

[80] Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. *arXiv e-prints*, art. arXiv:2301.01820, January 2023. doi: 10.48550/arXiv.2301.01820.

[81] Zhuolin Jiang, Amro El-Jaroudi, William Hartmann, Damianos Karakos, and Lingjun Zhao. Cross-lingual information retrieval with BERT. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 26–31, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-55-9. URL https://aclanthology.org/2020.clssts-1.5.

[82] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 133–142, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775067. URL https://doi.org/10.1145/775047.775067.

[83] J. Johnson, M. Douze, and H. Jegou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(03):535–547, Jul 2021. ISSN 2332-7790. doi: 10.1109/TBDATA.2019.2921572.

[84] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[85] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji,

Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P18-4020.

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

[87] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://www.aclweb.org/anthology/2020.emnlp-main.550.

[88] K Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJeT3yrtDr.

[89] Omar Khattab and Matei Zaharia. 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*, pages 39–48. Association for Computing Machinery, New York, NY, USA, July 2020.

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

[91] Kazuaki Kishida and Noriko Kando. A hybrid approach to query and document translation using a pivot language for cross-language information retrieval. In Carol Peters, Fredric C. Gey, Julio Gonzalo, Henning Müller, Gareth J. F. Jones, Michael Kluck, Bernardo Magnini, and Maarten de Rijke, editors, *Accessing Multilingual Information Repositories*, pages 93–101, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-45700-8.

[92] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. *Machine Translation Summit, 2005*, pages 79–86, 2005.

[93] Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL https://www.aclweb.org/anthology/N03-1017.

[94] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens,

Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `https://aclanthology.org/P07-2045`.

[95] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1007. URL `https://aclanthology.org/P18-1007`.

[96] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[97] Thomas K Landauer and Michael L Littman. A statistical method for language-independent representation of the topical content of text segments. In *Proceedings of the Eleventh International Conference: Expert Systems and Their Applications*, volume 8, page 85. Citeseer, 1991.

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

[99] Dawn Lawrie, James Mayfield, Douglas W Oard, and Eugene Yang. Hc4: A new suite of test collections for ad hoc clir. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I*, pages 351–366. Springer, 2022.

[100] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. Parade: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093*, 2020.

[101] Ping Li, Christopher J. C. Burges, and Qiang Wu. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 897–904, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

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

[103] Jimmy Lin. The neural hype, justified! a recantation. *SIGIR Forum*, 53(2):88–93, 2019. ISSN 0163-5840.

[104] Jimmy Lin and Xueguang Ma. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv:2106.14807*, 2021.

[105] Robert Litschko, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. Unsupervised cross-lingual information retrieval using monolingual data only. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1253–1256, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210157. URL https://doi.org/10.1145/3209978.3210157.

[106] Robert Litschko, Goran Glavaš, Ivan Vulic, and Laura Dietz. Evaluating resource-lean cross-lingual embedding models in unsupervised retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1109–1112, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331324. URL https://doi.org/10.1145/3331184.3331324.

[107] Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. Evaluating multilingual text encoders for unsupervised cross-lingual retrieval. In Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani, editors, *Advances in Information Retrieval*, pages 342–358, Cham, 2021. Springer International Publishing. ISBN 978-3-030-72113-8.

[108] Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. On cross-lingual retrieval with multilingual text encoders. *Information Retrieval Journal*, 25 (2):149–183, 2022.

[109] Michael L Littman, Susan T Dumais, and Thomas K Landauer. Automatic cross-language information retrieval using latent semantic indexing. In *Cross-language information retrieval*, pages 51–62. Springer, 1998.

[110] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[111] Ekaterina Loginova, Stalin Varanasi, and Günter Neumann. Towards multilingual neural question answering. In András Benczúr, Bernhard Thalheim, Tomáš Horváth, Silvia Chiusano, Tania Cerquitelli, Csaba Sidló, and Peter Z. Revesz, editors, *New Trends in Databases and Information Systems*, pages 274–285, Cham, 2018. Springer International Publishing.

[112] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

[113] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 1101–1104, New York, NY, USA, July 2019. Association for Computing Machinery.

[114] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 49–58, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164.

[115] Yu A Malkov and DA Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020.

[116] Antonio Mallia, Michal Siedlaczek, Joel Mackenzie, and Torsten Suel. 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.*, pages 50–56, 2019. URL `http://ceur-ws.org/Vol-2409/docker08.pdf`.

[117] Antonio Mallia, Michał Siedlaczek, and Torsten Suel. 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*, pages 353–368. Springer, 2019.

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

[119] J. Scott McCarley. 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*, pages 208–214, College Park, Maryland, USA, June 1999. Association for Computational Linguistics. doi: 10.3115/1034678.1034716. URL `https://aclanthology.org/P99-1027`.

[120] Paul McNamee. *Textual representations for corpus-based bilingual retrieval*. PhD thesis, University of Maryland, Baltimore County, 2008.

[121] Paul McNamee and James Mayfield. Comparing cross-language query expansion techniques by degrading translation resources. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 159–166, 2002.

[122] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[123] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[124] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf`.

[125] David R. H. Miller, Tim Leek, and Richard M. Schwartz. A hidden markov model information retrieval system. In Fredric C. Gey, Marti A. Hearst, and Richard M. Tong, editors, *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 214–221. ACM, 1999. doi: 10.1145/312624.312680. URL `https://doi.org/10.1145/312624.312680`.

[126] Suraj Nair, Petra Galuščáková, and Douglas W Oard. 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*, SIGIR '20, pages 1581–1584, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401270. URL `https://doi.org/10.1145/3397271.3401270`.

[127] Suraj Nair, Eugene Yang, Dawn Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W Oard. 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*, pages 382–396, 2022.

[128] Suraj Nair, Eugene Yang, Dawn Lawrie, James Mayfield, and Douglas W Oard. 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*, 2022.

[129] Suraj Nair, Petra Galuščáková, Douglas Oard, Le Zhang, Damianos Karakos, and Bonan Min. Rationale training based neural re-ranking for ad-hoc clir. In Preparation.

[130] Suraj Nair, Eugene Yang, Dawn Lawrie, James Mayfield, and Douglas W Oard. Blade: Combining vocabulary pruning and intermediate pretraining for scaleable neural clir. In Preparation.

[131] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International*

Conference on Machine Learning, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

[132] Ramesh Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71, 2004.

[133] Jian-Yun Nie. Cross-language information retrieval. *Synthesis Lectures on Human Language Technologies*, 3(1):1–125, 2010.

[134] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

[135] Rodrigo Nogueira and Jimmy Lin. From doc2query to docTTTTTquery. Technical report, University of Waterloo, 2019. URL `https://cs.uwaterloo.ca/~jimmylin/publications/Nogueira_Lin_2019_docTTTTTquery-v2.pdf`.

[136] Douglas W Oard and Anne R Diekema. Cross-language information retrieval. *Annual Review of Information Science and Technology (ARIST)*, 33:223–56, 1998.

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

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

[139] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. Terrier information retrieval platform. In *European Conference on Information Retrieval*, pages 517–519. Springer, 2005.

[140] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://aclanthology.org/P02-1040`.

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

[142] Biswajit Paria, Chih-Kuan Yeh, Ian E.H. Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. Minimizing flops to learn efficient sparse representations. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SygpC6Ntvr`.

[143] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith

Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[144] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[145] Carol Peters and Martin Braschler. European research letter: Cross-language system evaluation: The clef campaigns. *Journal of the American Society for Information Science and Technology*, 52(12):1067–1072, 2001.

[146] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. ACL.

[147] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 275–281, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 1581130155. doi: 10.1145/290941.291008. URL https://doi.org/10.1145/290941.291008.

[148] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319.

[149] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 813–823, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391320. doi: 10.1145/3488560.3498433. URL https://doi.org/10.1145/3488560.3498433.

[150] Khaled Radwan. *Vers l'acces multilingue en langage naturel aux bases de donnees textuelles*. PhD thesis, Paris 11, 1994.

[151] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

[152] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

[153] Nils Reimers and Iryna Gurevych. 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, 11 2020. URL https://arxiv.org/abs/2004.09813.

[154] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *TREC*, pages 109–123, 1994.

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

[156] Franklin D. Roosevelt. Second inaugural address of franklin d. roosevelt. Available online via Avalon Project http://avalon.law.yale.edu/20th_century/froos2.asp, January 1937.

[157] Andreas Rücklé, Krishnkant Swarnkar, and Iryna Gurevych. Improved cross-lingual question retrieval for community question answering. In *The World Wide Web Conference*, WWW '19, page 3179–3186, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366748.

[158] Sebastian Ruder. Why You Should Do NLP Beyond English. http://ruder.io/nlp-beyond-english, 2020.

[159] Koustav Rudra and Avishek Anand. Distant supervision in bert-based adhoc document retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2197–2200, 2020.

[160] G. Salton. Automatic processing of foreign language documents. In *International Conference on Computational Linguistics COLING 1969: Preprint No. 4*, Sånga Säby, Sweden, September 1969. URL https://aclanthology.org/C69-0401.

[161] Gerard Salton and Michael E Lesk. The smart automatic document retrieval systems—an illustration. *Communications of the ACM*, 8(6):391–398, 1965.

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

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

[164] Erwin K Scheuch and Philip J Stone. The general inquirer approach to an international retrieval system for survey archives. *American Behavioral Scientist*, 7(10): 23–28, 1964.

[165] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

[166] Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. URL https://proceedings.neurips.cc/paper_files/paper/2002/file/51de85ddd068f0bc787691d356176df9-Paper.pdf.

[167] P Shi and J Lin. Cross-lingual relevance transfer for document retrieval. *arXiv preprint arXiv:1911.02989*, 2019.

[168] Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. Cross-lingual training of dense retrievers for document retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 251–253, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.mrl-1.24. URL https://aclanthology.org/2021.mrl-1.24.

[169] Xing Shi and Kevin Knight. 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)*, pages 574–579, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2091. URL https://aclanthology.org/P17-2091.

[170] Karen Sparck Jones. Automatic indexing. *Journal of documentation*, 30(4):393–432, 1974.

[171] Philip J Stone, Robert F Bales, J Zvi Namenwirth, and Daniel M Ogilvie. The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. *Behavioral Science*, 7(4):484, 1962.

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

[173] Shuo Sun and Kevin Duh. 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)*, pages 4160–4170, 2020.

[174] Weiwei Sun, Chuan Meng, Qi Meng, Zhaochun Ren, Pengjie Ren, Zhumin Chen, and Maarten de Rijke. Conversations powered by cross-lingual knowledge. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1442–1451, 2021.

[175] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 77–86, 2008.

[176] Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf`.

[177] Jörg Tiedemann and Lars Nygaard. The OPUS corpus - parallel and free: http://logos.uio.no/opus. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[178] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

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

[180] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[181] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

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

[183] Ivan Vulić and Marie-Francine Moens. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 363–372, New York, NY, USA, 2015. Association for

Computing Machinery. ISBN 9781450336215. doi: 10.1145/2766462.2767752. URL https://doi.org/10.1145/2766462.2767752.

[184] Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. Pseudo-relevance feedback for multiple representation dense retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 297–306, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386111. doi: 10.1145/3471158.3472250. URL https://doi.org/10.1145/3471158.3472250.

[185] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France, May 2020. European Language Resources Association.

[186] Rachel Wicks and Matt Post. A unified approach to sentence segmentation of punctuated text in many languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3995–4007, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.309. URL https://aclanthology.org/2021.acl-long.309.

[187] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[188] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[189] Shengli Wu. *Data Fusion in Information Retrieval*. Springer Publishing Company, Incorporated, 2012. ISBN 3642288650.

[190] Zhijing Wu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Investigating passage-level relevance and its role in document-level relevance judgment. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 605–614, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331233. URL https://doi.org/10.1145/3331184.3331233.

[191] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1192–1199, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390306. URL https://doi.org/10.1145/1390156.1390306.

[192] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, 2015.

[193] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 55–64, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350228. doi: 10.1145/3077136.3080809. URL https://doi.org/10.1145/3077136.3080809.

[194] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

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

[196] Jun Xu and Hang Li. Adarank: A boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 391–398, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595935977. doi: 10.1145/1277741.1277809. URL https://doi.org/10.1145/1277741.1277809.

[197] Eugene Yang, Suraj Nair, Ramraj Chandradevan, Rebecca Iglesias-Flores, and Douglas W. Oard. 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*, SIGIR '22, page 2507–2512, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531886. URL https://doi.org/10.1145/3477495.3531886.

[198] Jheng-Hong Yang, Xueguang Ma, and Jimmy Lin. Sparsifying sparse representations for passage retrieval by top-k masking. *arXiv preprint arXiv:2112.09628*, 2021.

[199] Peilin Yang, Hui Fang, and Jimmy Lin. 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*, SIGIR '17, pages 1253–1256, New York, NY, USA, August 2017. Association for Computing Machinery.

[200] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

[201] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3481–3487, 2019.

[202] Puxuan Yu and James Allan. A study of neural matching models for cross-lingual ir. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1637–1640, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401322. URL https://doi.org/10.1145/3397271.3401322.

[203] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 271–278, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595935977. doi: 10.1145/1277741.1277790. URL https://doi.org/10.1145/1277741.1277790.

[204] Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 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*, CIKM '18, page 497–506, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271800. URL https://doi.org/10.1145/3269206.3271800.

[205] Rabih Zbib, Lingjun Zhao, Damianos Karakos, William Hartmann, Jay DeYoung, Zhongqiang Huang, Zhuolin Jiang, Noah Rivkin, Le Zhang, Richard Schwartz, et al. 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*, pages 645–654, 2019.

[206] Rui Zhang, Caitlin Westerfield, Sungrok Shim, Garrett Bingham, Alexander Fabbri, William Hu, Neha Verma, and Dragomir Radev. Improving low-resource cross-lingual document retrieval by reranking with deep bilingual representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3173–3179, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1306. URL https://aclanthology.org/P19-1306.

[207] Lingjun Zhao, Rabih Zbib, Zhuolin Jiang, Damianos Karakos, and Zhongqiang Huang. Weakly supervised attentional model for low resource ad-hoc cross-lingual information retrieval. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 259–264, Hong Kong, China, November 2019. Association for Computational Linguistics.

[208] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, Los Alamitos, CA, USA, dec 2015. IEEE Computer Society. doi: 10.1109/ICCV.2015.11. URL `https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.11`.

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

[210] Shengyao Zhuang and Guido Zuccon. 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*, pages 1483–1492, 2021.

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