END-TO-END MULTILINGUAL INFORMATION RETRIEVAL WITH MASSIVELY LARGE SYNTHETIC DATASETS

by

Shuo Sun

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland

June, 2022

Abstract

End-to-end neural networks have revolutionized various fields of artificial intelligence. However, advancements in the field of Cross-Lingual Information Retrieval (CLIR) have been stalled due to the lack of large-scale labeled data. CLIR is a retrieval task in which search queries and candidate documents are in different languages. CLIR can be very useful in some scenarios: for example, a reporter may want to search foreign-language news to obtain different perspectives for her story; an inventor may explore the patents in another country to understand prior art.

This dissertation addresses the bottleneck in end-to-end neural CLIR research by synthesizing large-scale CLIR training data and examining techniques that can exploit this in various CLIR tasks. We publicly release the Large-Scale CLIR dataset and CLIRMatrix, two synthetic CLIR datasets covering a large variety of language directions. We explore and evaluate several neural architectures for end-to-end CLIR modeling. Results show that multilingual information retrieval systems trained on these synthetic CLIR datasets

are helpful for many language pairs, especially those in low-resource settings. We further show how these systems can be adapted to real-world scenarios.

Primary Reader and Advisor: Kevin Duh

Secondary Readers: Philipp Koehn & Paul McNamee

Acknowledgments

There are no words I can use to describe how grateful I am to my advisor Prof. Kevin Duh, for his help, advice, and support during my academic years at JHU. He has taught me how to conduct original research, formulate and execute experiments, and present coherent results in research papers and talks. I also want to thank Kevin for correcting my bad habits of focusing too much on trying to beat state-of-the-art systems and spending too much research time on parameter tuning and feature engineering. Kevin has also trained me to approach research problems from broader perspectives. I am indebted to Kevin for his unwavering guidance over the past five years. His dedication and passion for work make him a role model to me in pursuing my career after graduation.

I want to thank Prof. David Yarowsky, Prof. Carey Priebe, Prof. Sanjeev Khudanpur, and Prof. Matt Post for being part of my Graduate Board Oral (GBO) exam committee and providing valuable comments on my proposal, I would also like to thank my dissertation committee members, Prof. Philipp

Koehn and Dr. Paul McNamee, for carefully reading my dissertation and providing insightful suggestions that make this work better. I am also grateful to Prof. João Sedoc for working with me on the Haodf and COVID-19 chatbot projects and Prof. Mark Dreze for giving me the opportunity to work as a teaching assistant for his machine learning class and teaching me how to be a better teacher.

I want to thank all professors and my friends in the CLSP community who made my doctoral study at the JHU a happy memory. A special thanks to Hongyuan Mei for helping me settle down in Baltimore, for driving me to the car dealership, and for waiting for me while I was negotiating the car price. I also want to thank my dinner buddies: Dongji Gao, Hang Lv, and Chunxi Liu, and collaborators: Shota Sasaki, Susanna Sia, Adam Poliak, Max Fleming, Cash Costello, Kenton W Murray, Mahsa Yarmohammadi, Shivani Pandya, Darius Irani, Milind Agarwal, Udit Sharma, Nicola Ivanov, Lingxi Shang, Kaushik Srinivasan, Seolhwa Lee, Xu Han, and Smisha Agarwal.

Off-campus, I am lucky to have the opportunity to work with Dr. Jian Su, Dr. Bin Chen, and Dr. Wei Zhang. They introduced me to Natural Language Processing and gave me the opportunities to work on relationship extraction, entity linking, and sentiment analysis. Thanks to Dr. Jie Cao, Dr. Zuohui Fu, Dr. Wilson Tam, and Dr. Cheng Niu for being the best teammates while building our systems for DSTC7. Thanks to Dr. Paco Guzman and Prof. Lucia

ACKNOWLEDGMENTS

Specia for mentoring me on the quality estimation projects and to other collaborators for providing useful feedback and technical assistance while I was interning at Facebook: Ahmed El-Kishky, Vishrav Chaudhary, James Cross, Hongyu Gong, Holger Schwenk, Adithya Renduchintala, Marina Fomicheva, Lisa Yankovskaya, Frédéric Blain, Mark Fishel, Nikolaos Aletras.

I am grateful to A*STAR Graduate Academy for providing the funding to support my studies throughout my undergraduate and doctorate years. I would also like to thank Ruth Scally, Zack Burwell, and Kim Franklin for providing help on many occasions involving administrative matters.

I want to thank my parents and parents-in-law for their love, support, and unwavering belief in me. Without you, I would not be the person I am today.

Finally, I want to express my deepest gratitude to my wife, Cheng Xu, for her love and constant support, for keeping our apartment and belongings COVID-free, and for keeping me motivated over the past few months. Thank you for being my best friend. I owe you everything.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cross-Lingual Information Retrieval (CLIR) or Cross-Language Information Retrieval is a sub-field of information retrieval that deals with search queries and documents written in different languages. As the internet becomes ubiquitous worldwide, the amount of easily-accessible documents written in different languages has drastically increased over time. The need to search through this sea of information gives rise to the field of Information Retrieval (IR). Normally, users would be interested in the relevant information in the same native language as their search query. However, there are several scenarios where useful information is only available in other languages:

- a reporter may want to search foreign language news to obtain different perspectives for her story
- an inventor may explore the patents in another country to understand prior art
- an investor who wishes to monitor consumer sentiment of an international brand in Twitter conversations around the world

A CLIR system, as shown in Figure 1.1 is a specialized information retrieval system that is capable of ingesting a search query in one language and returning relevant documents in another language.

There are two main approaches to building CLIR systems: *the modular approach* and the *direct modeling approach*. The modular approach involves

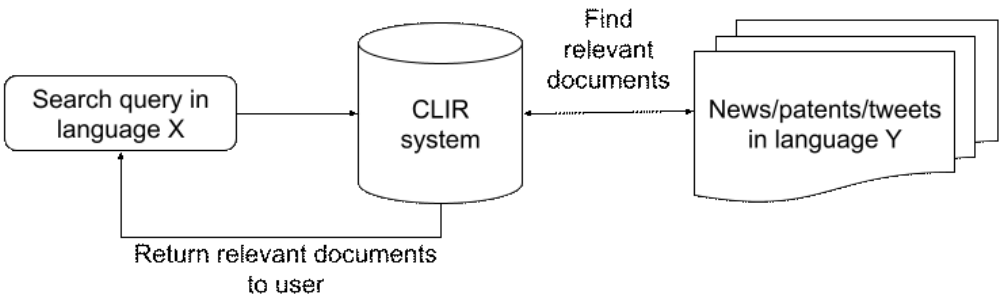


Figure 1.1: System pipeline of a CLIR system.

two consecutive steps: The first step is translation, where we translate queries or documents with machine translation systems. The second step is retrieval, where we retrieve relevant documents using monolingual information retrieval systems. The direct modeling approach focuses on end-to-end models capable of directly handling the raw input queries and documents without an intermediate machine translation step.

The performance of the modular approach relies heavily on the availability of good machine translation and monolingual information retrieval systems. However, machine translation is not yet effective in many language directions, hindering the performance of the downstream retrieval step. In addition, the lack of monolingual IR data in most languages increases the challenges of CLIR in low-resource language pairs such as Tagalog-Swahili. Therefore, recent work has focused on end-to-end direct modeling approaches that avoid the challenges of building translation and monolingual information retrieval systems separately.

1.1 Motivation

Despite growing interest in end-to-end CLIR, the lack of a large-scale, easily-accessible CLIR dataset covering many language directions in high-, mid-, and low-resource settings has detrimentally affected the CLIR community's capability to replicate and compare with previously published work. Advancements in the field of Cross-Lingual Information Retrieval (CLIR) have been stalled due to the lack of large-scale labeled data.

This dissertation addresses the research limitations of CLIR by contributing two large-scale synthetic CLIR datasets to the community: 1) The Large-Scale CLIR Dataset[1] and 2) CLIRMatrix[2]. Both datasets are publicly available and contain many training examples across various language pairs, making them suitable for training end-to-end multilingual information retrieval systems. Equipped with these datasets, we can explore and thoroughly evaluate several suitable neural architectures for end-to-end CLIR modeling. We further show that multilingual information retrieval systems trained on these large-scale CLIR datasets benefit many language pairs, especially those in low-resource settings.

[1] https://www.cs.jhu.edu/~kevinduh/a/wikiclir2018/
[2] https://github.com/ssun32/CLIRMatrix

1.2 Overview of Contributions

1.2.1 Datasets

Dataset	#Language #query	Manual? #document	Multilingual? #triplets
(CLEF 2000-2003, 2003)	10 2.2K	yes 1.1M	yes 33K
(MATERIAL, 2017)	7 11.5K	yes 90K	no ∼20K
(Schamoni et al., 2014b)	2 245K	no 1.2M	no 3.2M
Large-Scale CLIR Dataset	25 10.9M	no 23.9M	no 40.1M
CLIRMatrix BI-139 raw	139 49.3M	no 50.5M	no 34.1B
CLIRMatrix BI-139 base	139 27.5M	no 50.1M	no 22.3B
CLIRMatrix MULTI-8	8 10.4K	no 13.4M	yes 72.8M

Table 1.1: Comparison of CLIR datasets by number of languages (**#Lang**), whether it is manually constructed or supports multilingual retrieval, and data statistics. Large **#query** and **#triplets** are needed for neural training.

Although CLIR datasets are available in shared tasks such as CLEF and NTCIR (Galuščáková et al., 2021), most of these existing datasets contain only around 20 to 200 queries for each language pair, making them more suitable for evaluation purposes rather than training direct modeling models. The more recent and more extensive IARPA MATERIAL/OpenCLIR collection (Zavorin

et al., 2020) is not yet publicly accessible. Therefore, no existing large-scale CLIR dataset can support direct modeling approaches in various languages.

To obtain relevance judgments, one typically needs a bilingual speaker who can read a foreign-language document and assess whether it is relevant to a given English query. This can be an expensive process that is not scalable to most language directions. This dissertation describes the building procedures of two synthetic datasets: Large-Scale CLIR Dataset and CLIRMatrix. A comparison of various existing CLIR datasets is presented in Table 1.1.

The Large-Scale CLIR Dataset

In chapter 5, we introduce the Large-Scale CLIR Dataset. This dataset is derived from Wikipedia and contains more than 2.8 million English single-sentence queries with relevant documents from 25 other selected languages.

All queries and documents in this dataset are extracted from the August 23, 2017, version of the Wikipedia dump. For practical purposes, each document is limited to the first 200 words of the article. Empty documents and category pages are also filtered. Relevance judgments are constructed from the inter-language links between English Wikipedia articles and Foreign Language Wikipedia articles. A relevance level 2 is assigned to the (English) cross-lingual mate and level 1 to all other articles linked to the mate and linked by the mate. Statistics of this dataset are shown in Table 1.2. This

Language	#Doc	#Query	#SR
Arabic	535	324	194
Catalan	548	339	625
Chinese	951	463	462
Czech	386	233	720
Dutch	1908	687	1646
Finnish	418	273	665
French	1894	1089	4048
German	2091	938	4612
Italian	1347	808	2635
Japanese	1071	426	2912
Korean	394	224	343
Norwegian-Nynorsk	133	99	150
Norwegian-Bokmål	471	299	663
Polish	1234	693	1777
Portuguese	973	611	1130
Romanian	376	199	251
Russian	1413	664	1656
Simple English	127	114	135
Spanish	1302	781	2113
Swahili	37	22	35
Swedish	3785	639	1430
Tagalog	79	48	23
Turkish	295	185	195
Ukrainian	704	348	565
Vietnamese	1392	354	257
(All numbers are in units of one thousand)			

Table 1.2: Large-Scale CLIR dataset statistics. For each language X, we show the total number of documents in language X and the number of English queries. The number of "most relevant" documents is by definition equal to #Query. The number of "slightly relevant" documents is shown in the column #SR.

dataset is publicly available at https://www.cs.jhu.edu/~kevinduh/a/wikiclir2018/.

THE CLIRMATRIX COLLECTION

In chapter 7, we introduce the CLIRMatrix Collection, which contains bilingual CLIR datasets for 19,182 language pairs and multilingual IR datasets jointly aligned in 8 languages. This dataset is constructed from Wikipedia in an automated manner, exploiting its large variety of languages and massive number of documents. The core idea is to synthesize relevance labels via an existing monolingual IR system, then propagate the labels via Wikidata links that connect documents in different languages. We were able to mine 49 million unique queries in 139 languages and 34 billion (query, document, label) triplets. This dataset is publicly available at `https://github.com/ssun32/CLIRMatrix`. It is also available in **ir_datasets**, a package for acquiring, managing, and performing typical operations over datasets used in IR (MacAvaney et al., 2021).

1.2.2 Models and Findings

Based on our newly created massively large synthetic datasets, we propose and explore several neural architectures useful for cross-lingual information retrieval.

In chapter 4, we propose the regularized self-attention ranking network (RSARN), which is a listwise neural approach to the learning to rank problem. We propose novel attention regularizers designed to control the weights of self-

attention layers over the vector representations of query-document pairs. We show that RSARN can significantly outperform state-of-the-art ensemble tree-based methods on publicly available monolingual datasets and the CLEF 2003 CLIR dataset.

In chapter 5, we explore a CLIR model that uses convolutional neural networks to encode and predict the relevance of a document to a query. We propose a method to bootstrap bilingual IR models for languages with less training data by using parameter sharing among different language pairs. For example, using the training data for Japanese-English CLIR, we can improve the Mean Average Precision (MAP) results of a Swahili-English CLIR system by 5-7 points.

In chapter 6, we empirically explore the cross-encoder version of the Multilingual BERT Ranker Model (MBRM) and show it outperforms state-of-the-art systems with minimal supervision. We further show that MBRM is robust and does not suffer from the partial-input baseline problems observed in other tasks (Poliak et al., 2018; Gururangan et al., 2018).

In chapter 7, we conduct more experiments on the CLIRMatrix datasets. Our experiment results show that a single MBRM trained on data from multiple language pairs significantly outperforms an ensemble of bilingual ranker models.

In chapter 8, we explore a bi-encoder variant of the MBRM and show it per-

forms better than modular CLIR systems on real-world CLIR datasets from the CLEF 2003 evaluation campaign. We explore several strategies when dealing with scenarios with few or no training examples in the domain of interest. Our experiment results show that modular CLIR systems only work when sufficient parallel sentences exist, while direct model CLIR systems outperform modular CLIR systems in low-resource settings. We further show it is beneficial to first train CLIR systems on synthetic CLIR datasets such as CLIRMatrix and fine-tune those models on in-domain data.

Chapter 2

Background

2.1 Cross-Lingual Information Retrieval

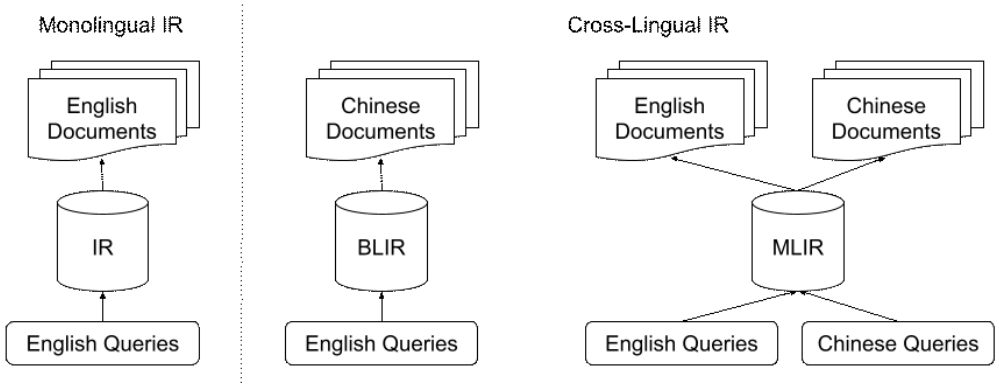


Figure 2.1: Examples of monolingual IR system, bilingual IR (BLIR) system and multilingual IR (MLIR) system

Monolingual IR systems are designed to handle queries and documents written in the same language, whereas cross-lingual IR systems specialize in handling queries and documents written in different languages. The differences between the IR systems, as mentioned earlier, are illustrated in Figure 2.1. CLIR systems can be further sub-categorized into bilingual IR (BLIR) systems and multilingual IR (MLIR) systems: BLIR systems handle only one language direction (LD), and a separate BLIR system has to be built for each language direction; MLIR systems, on the other hand, are capable of handling queries and documents in multiple languages, thereby reducing the need to build BLIR systems for every language direction.

We now lay the foundation for the remaining of this dissertation by formally defining the CLIR task. The standard notation we used is shown in Table 2.1.

Notation	Meaning
q	query
q_i	term at position i of query q
d	document
d_i	term at position i of document d
f	ranking function
q^X	query written in language X
q_i^X	term at position i of query q written in language X
d^Y	document written in language Y
d_i^Y	term at position i of document d written in language Y
f_{BL}	bilingual ranking function
f_{ML}	multilingual ranking function
S	set of source languages for queries
T	set of target languages for documents
Q	collection of queries
D	collection of candidate documents

Table 2.1: Notation used in this dissertation

2.1.1 Bilingual Information Retrieval

Bilingual information retrieval (BLIR), as its name suggests, handles only two languages. Given some query in language X, q^X and a collection of candidate documents in language Y, $D = \{d_1^Y, d_2^Y \dots d_N^Y\}$ under the condition $X \neq Y$, the goal of a BLIR system is to learn a ranking function f_{BL}, such that $f_{BL}(q^X, d^Y)$ returns a value that represents the relevancy of d^Y given q^X. Ideally, for any pair of documents (d_i^Y, d_j^Y), we want $f_{BL}(q^X, d_i^Y) > f_{BL}(q^X, d_j^Y)$ if d_i^Y is a more relevant document compared to d_j^Y. A BLIR system would return the most relevant documents to users based on the outputs of the ranking function f_{BL}.

2.1.2 Multilingual Information Retrieval

An MLIR system can handle queries written in any source language from S and retrieve documents written in any target language from T. S and T are sets of supported source and target languages. For any document pair $(d_i^Y, d_j^{Y'})$ where $Y, Y' \in T$ and Y and Y' can be different languages, a MLIR system should ideally learn a ranking function f_{ML} such that $f_{ML}(q^X, d_i^Y) > f_{ML}(q^X, d_j^{Y'})$ if d_i^Y is considered more relevant than $d_j^{Y'}$ given an input query q^X, where $X \in S$ [1].

2.2 Monolingual Approaches

As discussed in Section 2.1, the research challenge of information retrieval is how to build an accurate ranking function f that computes the degree of relevancy of a query-candidate pair. Many different approaches have been proposed in the past decades, especially for monolingual IR systems. We now provide a high-level overview of some of the most well-known approaches to monolingual IR and discuss various common methods to adapt the monolingual systems to cross-lingual settings.

[1] Technically, MLIR systems can also be trained to handle monolingual information retrieval, but we would only focus on cross-lingual cases where $X \neq Y$ in this dissertation.

2.2.1 Vector Space Model

One of the earliest approaches to modeling the ranking function f is the vector space model (Salton et al., 1975). This model assumes that the relevance of a candidate document d to a query q is approximately the similarity between their vector representations. The ranking function is defined as:

$$f(q, d) = sim(E_1(q), E_2(d)) \tag{2.1}$$

where E_1 and E_2 are encoder functions used to convert queries and documents into vector representations, and sim is a function that measures the degree of similarity between a query and a document. Traditionally, the encoder functions transform input strings of arbitrary lengths into vector representations of dimension V, where V is the size of a predetermined vocabulary. Each element in the vector representations is the weight of the corresponding term in the vocabulary. Various weighting schemes have been developed over the past decades. A naive method is to weigh the terms in a binary manner by assigning a weight of 1 to the i-th element if the input string contains the i-th term in V. Another commonly used weighting scheme is the term frequency-inverse document frequency (tf-idf), which increases the weight of a query term proportionate to the number of times it appears in a candidate document and is offset by the number of candidate documents that contain the term. The most

commonly used similarity function is the cosine similarity, defined as:

$$sim(\vec{V}, \vec{V'}) = \frac{\sum_{i=1}^{N} \vec{V_i} \cdot \vec{V_i'}}{\sqrt{\sum_{i=1}^{N} \vec{V_i}} \cdot \sqrt{\sum_{i=1}^{N} \vec{V_i'}}} \tag{2.2}$$

where $\vec{V}$ and $\vec{V'}$ are vectors of dimension N.

2.2.2 BM25

BM25 is a bag-of-word retrieval function that computes the similarity between a query and a document based on the occurrence of query terms in the document (term frequency) and the inverse document frequency of the query terms.

Given a query string q containing the terms $\{q_1, q_2, \ldots q_n\}$ and a document d containing the terms $\{d_1, d_2, \ldots d_m\}$, the relevance score of d given q is:

$$f(q, d) = \sum_{i=1}^{n} IDF(q_i).\frac{TF(q_i, d).(k_1 + 1)}{TF(q_i, d) + k_1.(1 - b + b.\frac{|d|}{avgdl})} \tag{2.3}$$

where $TF(q_i, d)$ is the term frequency of q_i in document d, $IDF(q_i)$ is the inverse document frequency of q_i, $|d|$ is the length of document d in terms of number of word tokens and $avgdl$ is the average length of all candidate documents. b and k_1 are hyper-parameters that can be optimized[2]. $TF(q_i, d)$ is commonly defined

[2]Practitioners usually use the default values $b = 0.75$ and $k_1 = 1.2$

as the number of times the term q_i appears in document d, while $IDF(q_i)$ is a weighting mechanism to scale down the importance of the term q_i if it appears too often in candidate documents. A commonly used method to calculate the IDF is:

$$IDF(q_i) = \ln(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1) \tag{2.4}$$

where N is the total number of candidate documents and $n(q_i)$ is the number of documents that contain the term q_i.

2.2.3 Language Modeling (LM)

Language modeling is a task that deals with assigning probabilities to sentences. It was commonly used in statistical machine translation (Brown et al., 1993) and speech recognition (Jelinek, 1997) but has also been adapted to the task of information retrieval (Ponte and Croft, 1998; Zhai and Lafferty, 2001). The main idea of this approach is that the relevance of a document d, given a query q, can be modeled by the posterior probability:

$$p(d|q) = \frac{p(q|d)p(d)}{p(q)} \propto p(q|d)p(d) \tag{2.5}$$

We can then rank candidate documents based on their posterior probabilities given the input query, q. The calculation of $p(q|d)$ differs from model to model but it is commonly assumed that the query terms are independent (Song and

Croft, 1999; Ponte and Croft, 1998) and $p(d)$ is uniform. This simplifies equation 2.5 to:

$$p(d|q) = p(q|d) = \prod_{q_i \in q} p(q_i|d) \tag{2.6}$$

and $p(q_i|d)$ can be estimated by a statistical language model of the document.

2.2.4 Learning to Rank

With recent advances in learning techniques and increased availability of monolingual information retrieval data, the research community has been focusing primarily on *learning to rank (LTR)*, which is the application of machine learning to build ranking functions for IR systems.

The ranking function can be learned using the learning to rank paradigm with supervised, unsupervised, or other machine learning training methods. The most successful learning to rank methods is typically trained in a supervised manner, which fits a global ranking function $f(\cdot)$ on a training set that consists of queries, collections of documents, and their desired rankings. The ranking function is trained with either of the three common approaches:

1. **Pointwise approaches** optimize the relevance score of a single document without considering other documents in the same set.

2. **Pairwise approaches** optimize the ranking between pairs of documents, such that $f(q, d_i) > f(q, d_j)$ if d_i ranks better than d_j.

3. **Listwise approaches** directly attempts to optimize the target IR metric (such as MAP or NDCG), which is based on the entire set of document scores.

Pointwise approaches use popular regression algorithms such as Breiman (2001); Friedman (2002) to estimate relevance judgments of documents directly. Pairwise approaches such as Adomavicius and Tuzhilin (2005); Joachims (2002); Burges et al. (2007) formulate the learning to rank task as a binary classification problem and use machine learning algorithms such as ensemble trees (Burges et al., 2011) and neural networks (Burges et al., 2005; Cao et al., 2007). Listwise approaches are more challenging because popular IR metrics such as MAP, ERR, and NDCG are not differentiable. Since gradient descent-based methods cannot be directly used for optimization, various surrogate loss functions have been proposed over the years. For example, Cao et al. (2007) proposed ListNet, which uses the cross-entropy between the permutation probability distributions of the predicted ranking and the ground truth as a loss function. Taylor et al. (2008) proposed SoftRank that uses smoothed approximations to ranking metrics. Burges (2010) proposed LambdaRank and LambdaMART, which approximate gradients by the directions of swapping two documents, scaled by the change in ranking metrics.

We can further categorize learning to rank methods into *feature-based learning to rank* and *neural learning to rank* as shown in Figure 2.2.

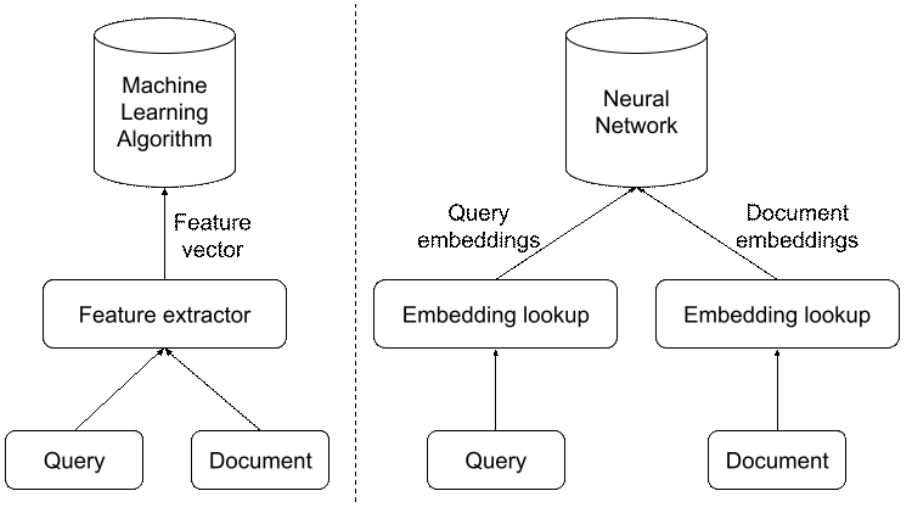


Figure 2.2: System pipeline of feature-based learning to rank (left) and neural learning to rank (right)

FEATURE-BASED LEARNING TO RANK

Feature-based learning to rank methods converts each query and document pair into a feature vector using manually constructed feature extractor functions and then applies machine learning methods to rank documents based on the feature vectors. Some commonly used features are the sum of tf-idf weights of query terms and the number of terms in query and document. Existing work also finds it helpful to include the cosine similarities from the vector space model, BM25 scores, and posterior probabilities from the language model as described in section 2.2. Readers interested in a more thorough explanation of the history of learning to rank and the technical details of various methods are encouraged to read the book by Liu (2011).

NEURAL LEARNING TO RANK

Neural learning to rank is a rapidly advancing field of research fueled by the recent advances in deep neural network techniques and cheaper compute resources. While earlier work trained neural network models on manually constructed features similar to the ones described in the previous section (Cao et al., 2007), recent research focuses on *end-to-end* models that directly ingest raw query and document texts. These models employ embedding lookup functions to convert queries and documents into vector representations and compute similarity scores between the query and document vectors with deep neural networks.

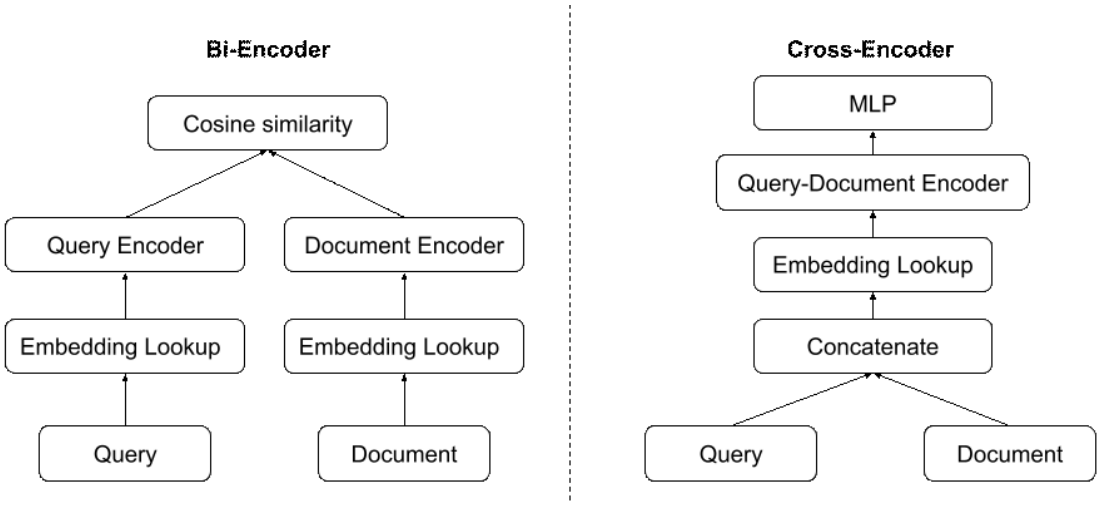


Figure 2.3: Bi-encoder (left) and cross-encoder (right) architectures for neural learning to rank

Many different neural architectures have been proposed for neural learning to rank over the years, of which two of the most successful neural architectures

are the *bi-encoder* architecture and the *cross-encoder* architecture.

The initial pre-processing steps of both architectures involve converting queries and documents with a variable number of tokens into fixed-sized query and document vectors using embedding lookup functions. More formally, an embedding lookup function E can be defined as:

$$E(t) = \vec{T} \tag{2.7}$$

where t is a token and $\vec{T}$ is the vector representation of that token. A token can be either word, character, or subword, and tokenizers are used to split queries or documents into lists of tokens. Readers who want a better understanding of the mechanisms of recent tokenizers can refer to Sennrich et al. (2016b); Kudo (2018) for more details. The dimension of $\vec{T}$ is fixed and pre-determined at training time, and some of the commonly used dimensions are 200, 300, 768, and 1024. E is designed only to accept tokens from a pre-defined vocabulary. In contrast, tokens not in that list are treated as out-of-vocabulary (OOV) and default to a shared vector representation used by all OOV tokens.

Earlier work randomly initializes token vectors' values and hopes to learn meaningful values for the vectors when training the models on IR datasets. A recent and more successful approach is to initialize them with word embeddings that were pre-trained on large-scale text data (Mikolov et al., 2013c;

Pennington et al., 2014b). These pre-trained word embeddings were trained such that words with similar meanings would be placed closer together in the embedding space.

The key differences between bi-encoder and cross-encoder architectures are summarized as follows:

- Bi-encoder architecture encodes queries and documents separately with independent query and document encoders and then computes the similarity between the query and document vector representations using a similarity function

- Cross-encoder architecture concatenates queries and documents before feeding their joint embedding into a query-document encoder. It then compresses the output encodings into similarity scores with neural networks such as multilayer perception (MLP layer).

In either architecture, the objective of the encoders is to "summarize" a sequence of query token vectors and document token vectors into one or two contextualized vectors. Some standard practices are encoding token vectors using recurrent neural network (RNN) such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997b) and then aggregating the outputs using pool methods such as sum pooling or mean pooling.

In recent research, the encoders are typically replaced by contextualized

language models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). These are huge language models with millions of parameters pre-trained on many text data using unsupervised learning. These models have drastically improved the performance of many tasks, including information retrieval (Akkalyoncu Yilmaz et al., 2019) and other natural language processing tasks such as question answering (Rajpurkar et al., 2018).

As neural learning to rank models contains many parameters, these models are typically trained on annotated IR datasets, using specialized hardware such as GPU and TPU to accelerate the training process.

2.3 Cross-Lingual Approaches

There are two main approaches to building CLIR systems. The *modular approach* involves a pipeline of two components: translation (machine translation or bilingual dictionary look-up) and monolingual information retrieval (IR).

A distinctly different way to build CLIR systems is what may be called the *direct modeling approach* (Bai et al., 2010; Sokolov et al., 2013). This approach assumes the availability of CLIR training examples of the form (q, d, r), where q is an English query, d is a foreign-language document, a r is the corresponding relevance judgment for d for q. One directly builds a retrieval model $S(q, d)$ that scores the query-document pair. While q and d are in different languages,

the model directly learns the CLIR training data's translation and retrieval relevance. Direct modeling is advantageous to the modular approach because it focuses on learning beneficial translations for retrieval rather than translations that preserve sentence meaning/structure in bitext.

2.3.1 Modular Approach

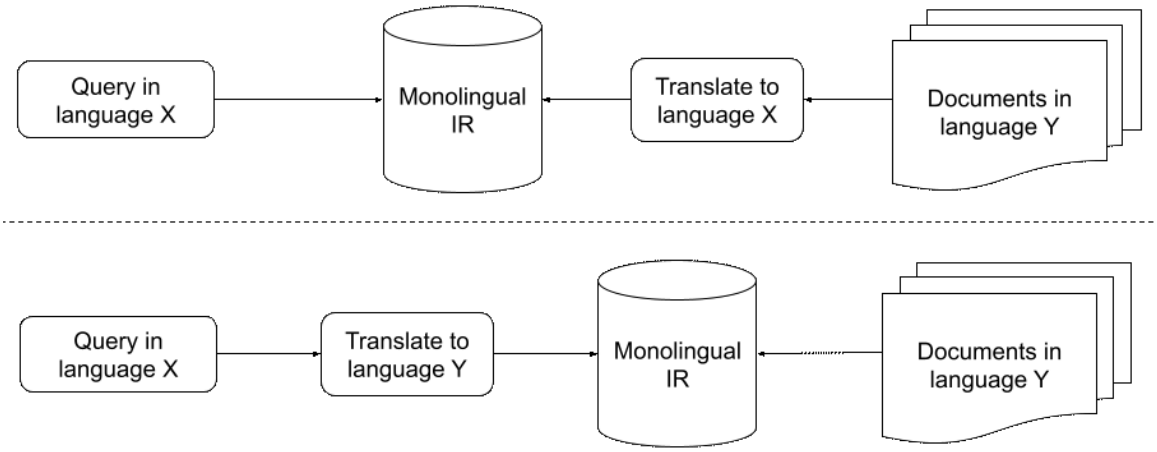


Figure 2.4: CLIR systems with document translation approach (top) and query translation approach (bottom).

The modular approach (or translation approach) may be further divided into the *document translation* and *query translation* approaches (Nie, 2010). In the former, one translates all foreign-language documents to the language of the user query before IR indexing; in the latter, one indexes foreign-language documents and translates the query. Another less commonly used approach is translating the query and document into a third language. The idea is to solve the translation problem separately so that CLIR becomes document retrieval

in the monolingual setting, and any of the techniques we describe in section 2.2 can be used to handle the actual retrieval of documents.

The performance of the downstream monolingual IR system relies heavily on the quality of the upstream machine translation system. Work on the modular approach generally follows the research progress of machine translation:

DICTIONARY-BASED MACHINE TRANSLATION

Earlier work such as Pirkola et al. (2001); Levow et al. (2005) translates queries and documents with bilingual dictionaries. However, there are several limitations to this approach. First, it assumes the existence of a comprehensive dictionary in the source-target language pair we are interested in or the existence of dictionaries that can map source and target languages to another language. However, the coverage of dictionaries for some language pairs might be limited, and the dictionary approach might not fully translate all search queries. Second, this approach requires language-specific pre-processing such as handling inflected words and proper nouns. Further, there are lexical ambiguities in source and target languages. Language-specific pre-processing requires linguistic knowledge of the languages we are interested in; therefore, dictionary-based modular systems might not scale to many language directions, especially those for languages without existing linguistic tools.

STATISTICAL MACHINE TRANSLATION

The second wave of CLIR systems is based on statistical machine translation that generates translations based on statistical machine translation (SMT) models trained on bilingual text corpora (Nikoulina et al., 2012; Katris et al., 2016). SMT models are usually based on the noisy channel model, where the most likely target sentence $\hat{t}$ and a source sentence s is given by:

$$\begin{aligned} \hat{t} &= \arg\max_{t} p(t|s) \\ &= \arg\max_{t} \frac{p(s|t)p(t)}{p(s)} \\ &= \arg\max_{t} p(s|t)p(t) \end{aligned} \tag{2.8}$$

where $p(t)$ is a language model of the target language learned from monolingual text data in the target language. $p(e|t)$ is a translation model trained on parallel sentences, where alignments between words or phrases are learned from source, and target sentences Koehn et al. (2003).

NEURAL MACHINE TRANSLATION

The current wave of translation-based CLIR systems use neural machine translation (NMT) that learns translation models with artificial neural networks (Bi et al., 2020; Saleh and Pecina, 2020; Yao et al., 2020). Unlike SMT

models that consist of different components such as language models and alignment models, NMT models are trained in an end-to-end manner to maximize training accuracy (Bahdanau et al., 2014; Luong et al., 2015).

NMT usually uses the sequence-to-sequence (Seq2Seq) modeling paradigm, where it uses neural networks such as LSTM (Hochreiter and Schmidhuber, 1997a) or transformer (Vaswani et al., 2017) to first encode a source sentence into a vector. Then based on that vector as an input, NMT uses another neural network to generate the translated sentence token by token.

Various improvements to the NMT have been proposed over the years. Some examples are using attention-mechanism to attend to all vector representations of tokens in source sentence (Luong et al., 2015), non-autoregressive machine translation that generates all translated tokens in parallel (Gu et al., 2017), data augmentation using the back translation technique (Sennrich et al., 2016a).

QUERY OR DOCUMENT TRANSLATION?

As queries are typically shorter than documents, the query translation approach might be more efficient and is the go-to choice when designing a CLIR system. However, short queries without context words are ambiguous, and machine translation might not preserve the original meaning of the queries. The document translation approach, on the other hand, has the added advan-

tage that words in a document occur in sequence, and machine translation models can produce more accurate translations based on the context of words (Galuščáková et al., 2021). Therefore, the general intuition is that document translation is generally more accurate and leads to better downstream IR performance. However, earlier studies (Franz et al., 1999; McCarley, 1999) have not observed any significant differences between these approaches, while later study (Saleh and Pecina, 2020) finds that query translation outperforms document translation in the medical domain.

This dissertation will revisit this question by examining query translation and document translation approach with the latest state-of-the-art machine translation techniques in chapter 8.

2.3.2 Direct Modeling Approach

The prerequisite for the modular approach is the availability of a machine translation system that can translate queries or documents without losing the semantics of the original texts. However, machine translation is not yet effective for many language directions, especially those without high-quality parallel corpora for training. Consequently, the lack of adequately translated queries and documents would severely affect the performances of the downstream monolingual IR systems.

An alternative solution is the direct modeling approach that handles the

retrieval of cross-lingual query-document pairs end-to-end. Earlier work designs end-to-end neural CLIR models that vaguely follow the neural learning to rank architectures as shown Figure 2.3. CLIR practitioners usually use recurrent neural networks or convolutional neural networks to extract features from both queries and documents and then optimize relevance scores $f(q, d)$ via some ranking loss (Huang et al., 2013a; Shen et al., 2014; Xiong et al., 2017; Mitra et al., 2017). To get better performance, these models usually initialize the word embedding layers with language-specific parameters trained on existing text documents (Mikolov et al., 2013a; Pennington et al., 2014a).

The advent of large pre-trained contextualized language models (Devlin et al., 2019; Conneau et al., 2020) has led to new state-of-the-art results on many NLP tasks. In chapter 6, 7 and 8, we examine the effectiveness of building CLIR models on these large pre-trained multilingual language models.

2.4 Existing Datasets

2.4.1 Cross-Lingual Mate-Finding

Before the availability of annotated CLIR datasets, researchers commonly evaluated their systems on the cross-lingual mate-finding task. In this task, original documents are treated as queries, and each has exactly one relevant

document in the translated versions of those documents. For example, Dumais and Letsche (1997) work with a test collection of 1500 English documents and 1500 corresponding French documents, all of which are sampled from the Hansard collection (the Canadian Parliament proceedings) (Roukos and Melamed, 1995). Each English document is considered a query, and the goal is to retrieve its translated "mate" from the French documents.

2.4.2 CLIR Datasets

Like monolingual information retrieval, existing CLIR datasets are usually released in shared task evaluations. Most of these existing datasets contain less than 100 queries, making them more suitable for evaluation than training end-to-end neural CLIR models. We summarize some of the shared task evaluation campaigns and their test collections:

TREC

The Text REtrieval Conference (TREC), co-sponsored by the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense, is designed to support research within the information retrieval community. TREC was the first venue that started evaluating CLIR systems on bilingual datasets involving languages such as Arabic, Spanish and Chinese. Most of the TREC datasets contain 25 to 50 queries.

CLEF 2000-2003 TEST COLLECTION

The Cross-Language Evaluation Forum (CLEF) organized cross-lingual information retrieval shared tasks focused on bilingual and multilingual document retrieval in European languages such as English, German, French and Italian from 2000 to 2003. Each language pair contains 40 to 60 queries and 40 to 500 thousand documents.

NTCIR 3-6 TEST COLLECTION

NII Testbeds and Community for Information Access research (NTCIR) is a series of workshops organized by Japan's National Institute for Informatics (NII). From 2002 to 2007, NTCIR organized a series of CLIR and MLIR shared tasks that mainly focused on Asian languages such as Japanese, Chinese, and Korean. Each language pair contains around 50 to 80 queries and 10 to 900 thousand documents.

FIRE 2008-2012 TEST COLLECTION

The Forum for Information Retrieval Evaluation (FIRE) is the south Asian counterpart to TREC, CLEF, and NTCIR. From 2008 to 2012, it organized various CLIR evaluation campaigns targeted at south Asian languages such as Hindi, Bangla, Marathi, Tamil, Telugu, Punjabi, and Malayalam. Each language pair contains around 50 queries and 95 to 500 thousand documents.

MATERIAL/OPENCLIR TEST COLLECTION

The Material/OpenCLIR is a series of evaluation campaigns organized by NIST that focus on low-resource languages such as Swahili and Tagalog. The whole test collection contains around 11 thousand queries and 90 thousand documents.

THE TREC 2022 NEUCLIR TRACK

Recently, the TREC 2022 NeuCLIR track[3] presents a new cross-language information retrieval challenge. The queries are written in English and have three target language collections: Chinese, Persian, and Russian. Each language pair contains around 50 to 60 queries and 3 to 5 million documents collected from the Common Crawl News Collection.

2.5 Evaluation Metrics

Given that human evaluation of retrieved results can be time-consuming and subjective, we typically evaluate the performance of an IR system by computing an automatic evaluation metric against a test dataset.

This subsection summarizes the most commonly used evaluation metrics: mean average precision (MAP) and normalized discounted cumulative gain

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

(NDCG).

2.5.1 Mean Average Precision (MAP)

Let Q be a set of unseen queries, D be a set of candidate documents, we define a *test dataset* as a set of triples $\{(q, d, r)\}$ for $q \in Q$, $d \in D$ and r is a value that represents the degree of relevancy of a document d given query q.

The mean average precision (MAP) score can be defined as the mean of the average precision scores over all queries in Q:

$$MAP = \frac{1}{|Q|} \cdot \sum_{q \in Q} AP(q) \tag{2.9}$$

where $|Q|$ is the number of queries in Q and $AP(q)$ is the average precision score of query q defined as:

$$AP(q) = \frac{1}{\sigma} \cdot \sum_{i=1}^{N} p@i(q) \cdot rel@k \tag{2.10}$$

where σ is the total number of relevant documents for query q, N is the total number of documents retrieved for q by an IR system, and $rel@k$ is an indicator function which is 1 if the retrieved document at position i is relevant and 0 otherwise. Lastly, $p@i(q)$ is the precision score of query q when only considering

the top i retrieved documents in $\{d'_1, d'_2, \ldots, d'_N\}$:

$$p@i(q) = \frac{1}{i} \cdot \sum_{i=1}^{i} g(q, d'_i) \tag{2.11}$$

where $g(q, d'_i)$ is an indicator function that is 1 if d'_i is a relevant document for query q and 0 otherwise. The range of MAP score is $[0, 1]$, where an ineffective IR system that fails to retrieve any relevant document for all queries will get a MAP score of zero, while a sound IR system that seldom makes mistakes will obtain a MAP score close to 1. The limitation of this metric is that it assumes the degree of relevancy is binary, where a document is considered either relevant or not relevant. Therefore, it is not suitable for IR datasets with multiple levels of relevancy.

2.5.2 Normalized Discounted Cumulative Gain (NDCG)

Normalized discounted cumulative gain (NDCG) is an IR metric that measures the usefulness of documents based on their ranks in the search results (Järvelin and Kekäläinen, 2002). It is a popular metric to measure the effectiveness of IR systems, and supports datasets with multiple levels of relevancy. NDCG is defined as the discounted cumulative gain (DCG) of a retrieved list of

documents normalized by the DCG of the ideal ranking of the documents. For a given query, let r'_i be the relevance judgment label of the i-th document in the predicted document ranking and r_i be the relevance judgment label of the i-th document in the optimal document ranking. The DCG of a predicted document ranking is defined as:

$$\text{DCG} = \sum_{i=1}^{N} \frac{2^{r'_i} - 1}{log_2(i+1)} \tag{2.12}$$

where N is the total number of retrieved documents. The main idea of DCG is that highly relevant documents that appear lower in the predicted document ranking will be penalized because each relevance judgment label is reduced logarithmically proportional to its position in the document list.

Following a common practice from the IR community, we calculate NDCG@k, which only evaluates the top k returned documents. We define DCG@k and ideal DCG@k as:

$$\begin{aligned} \text{DCG@k} &= \sum_{i=1}^{k} \frac{2^{r'_i} - 1}{log_2(i+1)} \\ \text{IDCG@k} &= \sum_{i=1}^{k} \frac{2^{r_i} - 1}{log_2(i+1)} \end{aligned} \tag{2.13}$$

We can calculate NDCG@k for that query as:

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}} \tag{2.14}$$

The NDCG@k of a test set is the arithmetic mean of NDCG@k values for all

queries. The range of the metric is [0, 1] and a higher NDCG@k score means predicted rankings are closer to the ideal rankings.

2.6 Recent Work

Recent advances in deep learning methods and releases of new CLIR datasets have attracted the attention of researchers and led to a new wave of work in CLIR, including improvements in modular approaches and novel direct modeling methods.

The recent MATERIAL/OpenCLIR challenge ignited research in applying multilingual embeddings in CLIR and the development of zero-shot or few-shot modeling approaches, while our release of the Large-Scale CLIR Dataset (Chapter 5) and CLIRMatrix (Chapter 7) further empowered novel research in areas such as language model pre-training and extensive CLIR evaluations in a large number of language directions. We summarize some of the recent work based on CLEF 200-2003, MATERIAL/OpenCLIR, and our datasets:

2.6.1 CLEF 2000-2003

Bonab et al. (2019) study the impact of translation resource scarcity on the performance of CLIR systems by developing a contrastive analysis framework that uses high-resource languages to simulate low-resource languages.

Yu and Allan (2020) empirically evaluate the effectiveness of combining interaction-based neural matching model such DRMM (Guo et al., 2016b) and KNRM (Xiong et al., 2017) with cross-lingual word embeddings (Litschko et al., 2018). The authors show that combining DRMM with cross-lingual word embeddings achieves the best results in four language directions from CLEF.

Nair et al. (2022) propose the ColBERT-X model for CLIR, which is based on the XLM-R (Conneau et al., 2019) and explore two ways to train the model. In the zero-shot setting, the authors trained ColBERT-X only on English MS MARCO collection and relied on XLM-R for cross-lingual mappings. In the translate-train setting, the authors trained ColBERT-x on English queries and translations of the MS MARCO passages.

2.6.2 MATERIAL/OpenCLIR

Zhao et al. (2019) address low-resource CLIR task by designing a model that is not trained on annotated CLIR datasets, instead but is weakly supervised by samples extracted from translation corpora.

Yarmohammadi et al. (2019) address the weaknesses of the document translation approach in low-resource CLIR settings by proposing a document representation that combines N-best translations and a novel bag-of-phrases output from MT systems. The authors show that richer document representations with multiple translation hypotheses consistently improve the performance of

low-resource CLIR systems.

Zbib et al. (2019) propose a neural network model to estimate word translation probabilities for CLIR. They improve the neural machine translation of a modular CLIR system by incorporating source word context and by encoding the character sequences of input source words to generate translations of out-of-vocabulary words. Their approach uses an unsupervised model to compute CLIR relevance scores.

Zhang et al. (2020) address the challenge of low-resource CLIR task by combining four different term interaction-based neural networks with cross-lingual word embeddings as inputs. Their model outperforms translation baselines and can be used in zero-shot and few-shot transfer learning settings.

Barry et al. (2020) project queries and documents in different languages into a shared embedding space with various contextualized language models and learn a matcher function that outputs the degree of relevance between query and document. The authors show that their approaches are competitive with strong translation baselines.

2.6.3 Our datasets

LARGE-SCALE CLIR DATASET

Lignos et al. (2019) show that it is challenging to optimize the upstream

translation models in a modular cross-lingual information retrieval system as the choice of IR dataset can substantially affect the predictive of MT tuning decisions and evaluations, which can potentially introduce dissociation between translation systems and the overall retrieval systems.

Liu et al. (2020) propose the Smooth Cosine Similarity, a novel measure of relevance between queries and documents, and the Smooth Ordinal Search Loss, a novel objective function for model training. The authors further provide a theoretical guarantee on the generalization error bound for the proposed framework.

Kuwa et al. (2020) introduce a method that learns embeddings for meta-textual categories and further improves retrieval performances based on combined embeddings of textual and meta-textual information.

Zhang and Tan (2021) explore different levels of text representations such as sub-word and character-level representations and show that building retrieval systems with various text representations lead to search improvements on our dataset.

Xu et al. (2021) introduce a semi-interactive mechanism that builds their model upon non-interactive architecture but encodes each document together with its associated multilingual queries, which significantly boosts the retrieval accuracy while maintaining the computational efficiency on our dataset.

Novak et al. (2022) propose the LM-EMD model, which uses Multilingual

BERT and Earth Mover's Distance (EMD). The authors evaluate their models on our dataset and provide interpretable insights on why a document is relevant to the query.

CLIRMatrix

Wang et al. (2021) achieve state-of-the-art results on CLIRMatrix by applying a triple loss to multilingual BERT and further aligning the token embeddings of different languages with adversarial networks.

Zhang et al. (2021) propose a model named CLIR with hierarchical knowledge enhancement (HIKE). The model encodes queries and documents with multilingual BERT, incorporates knowledge graph information with a hierarchical information fusion mechanism, and demonstrates substantial improvements over state-of-the-art neural CLIR models on CLIRMatrix.

Yang et al. (2022) yield improvements over state-of-the-art neural retrieval models by leveraging CLIRMatrix to continue pre-training off-the-shelf multilingual contextualized language models before fine-tuning on retrieval tasks.

Chapter 3

Overview of Models

This chapter provides an overview of the models used in this dissertation.

3.1 Regularized Self-Attention Ranking Network

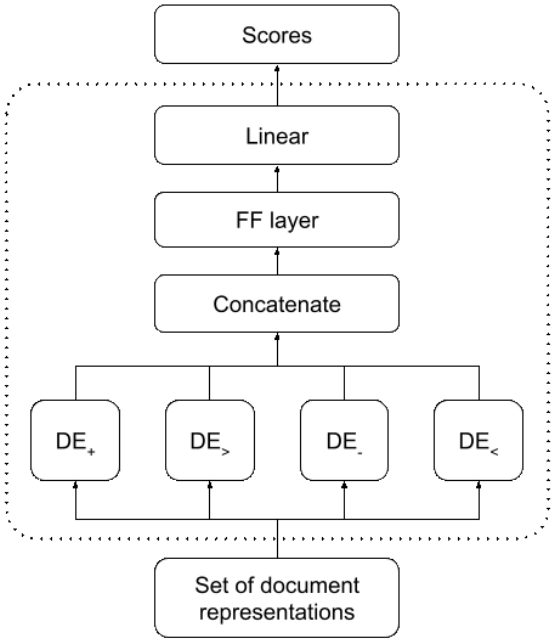


Figure 3.1: Architecture of the regularized self-attention ranking network

In chapter 4, we propose the regularized self-attention ranking network (RSARN), which models documents interactions with self-attention based neural networks. As seen in Figure 3.1, RSARN accepts a set of document representations and encodes them separately using different document encoders. Each document encoder has a self-attention layer where the attention weights

are regularized by a regularizer term. It then concatenates the outputs from the document encoders and compresses them into scores with feedforward layers. Every document is assigned a relevance score that represents its degree of relevance to a given query, and the final ordering of the candidate documents is obtained by sorting the predicted scores.

RSARN is a listwise learning to rank model that uses the ListNet loss function for optimization (Cao et al., 2007).

3.2 CLIR with Convolutional Neural Network

In chapter 5, we propose a cross-lingual information retrieval model based on convolutional neural network (CNN). Given a query q^X in language X and a document d^Y in language Y, we first use an embedding layer to convert each word into an n-dimensional vector, so q^X and d^Y are represented as matrices $\mathbf{Q^X} \in \mathbb{R}^{n \times |q^X|}$ and $\mathbf{D^Y} \in \mathbb{R}^{n \times |d^Y|}$, where $|q^X|$ and $|d^Y|$ are the numbers of tokens in q^X and d^Y. We then apply a convolutional layer on the query matrix and another convolutional layer with the same configurations on the document matrix. Each convolutional layer is configured to use filter size of n$\times$4, a stride of 1, and an output channel of 100. We then apply tanh activation and average-pooling to the outputs of the convolutional layers to obtain the vector represen-

tations $\hat{q^X}$ and $\hat{d^Y}$.

We experiment with two methods in computing the degree of relevancy between a query and a document $f(q^X, d^Y)$. The first is a *cosine model* which computes cosine similarity between $\hat{q^X}$ and $\hat{d^Y}$. The second is a *deep model* with a fully connected layer on top of the concatenation of $\hat{q^X}$ and $\hat{d^Y}$. This network is optimized using pairwise ranking loss.

3.3 Multilingual-BERT Ranker Model

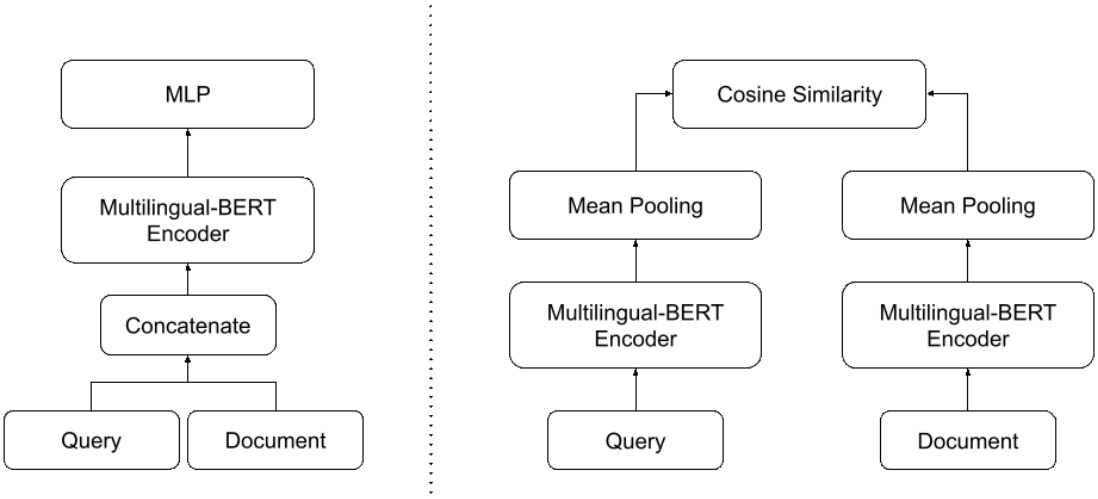


Figure 3.2: Multilingual-BERT Ranker Models: (Left) cross-encoder architecture (right) and bi-encoder architecture

3.3.1 Cross-Encoder

In chapter 6 and 7, we experiment with the multilingual BERT ranker model, which is inspired by (MacAvaney et al., 2019). This model follows the cross-encoder neural architecture that encodes a query-document pair with multilingual BERT (Devlin et al., 2019) and stacks a linear combination layer on top of the [CLS] token. At training time, we sample document pairs in which positive documents have higher relevance judgment labels than negative ones and optimize the model with pairwise hinge loss. At inference time, we rerank documents based on the output scores from the BERT ranker model. We further show that a multilingual information retrieval model trained on data concatenated from 56 language directions outperforms 56 individual bilingual information retrieval models, where each model is trained on data from one language direction.

To the best of our knowledge, we are the first among the CLIR community to show the effectiveness of CLIR models that encode query and document pairs with multilingual BERT. The only comparable work that has incorporated multilingual BERT in CLIR previously is Jiang et al. (2020). However, due to the lack of (query, document, label) training triplets, the authors utilize a data augmentation technique that breaks each query into word-level and each doc-

ument into sentence-level and train a Noisy-OR model defined as:

$$\begin{aligned} P(d^Y \text{is R} | q^X) &= P(q^X \text{occurs at least in one sentence in } d^Y) \\ &= 1 - \prod_{s \in d^Y} (1 - P(q^X | s)) \\ &= 1 - \prod_{s \in d^Y} (1 - \prod_{t \in q^X} p(t|s)) \end{aligned}$$

where s is a sentence from d^Y and t is a term in q^X and $p(t|s)$ is modeled by multilingual BERT model. In contrast to a model that works directly on full query and document text, this model has several weaknesses. First, it makes a strong assumption that the individual terms in a query and the sentences in a document are independent and therefore ignores the contexts of t and s. Second, it has to run the CLIR model N $\times$ M times to get the relevance score for each query and document pair, where N is the number of terms in the query and M is the number of sentences in the document. This is computationally more expensive than our models, which run only once per query-document pair. Last but not least, it has to run language-specific word tokenizers on queries and language-specific sentence segmenters on documents, which might not be available for many language pairs.

3.3.2 Bi-Encoder

One issue with the Multilingual-BERT ranker model is that it requires a time complexity of $\mathcal{O}(MN)$ to find relevant documents for M queries from N candidate documents. In chapter 8, we experiment with a faster variant of the Multilingual-BERT ranker model, which is inspired by the Sentence-BERT method (Reimers et al., 2019). This model follows the bi-encoder architecture in Figure 3.2 which encodes queries and documents independently with the same multilingual BERT encoder and then uses a mean pooling layer to compress the list of outputs into vectors that represent the query or document. It then uses the cosine similarity between the vector representations of the query and document to estimate their degree of relevancy.

The time complexity of this method is improved to $\mathcal{O}(M + N)$ since we only have to encode the M queries and N documents. However, to rank and retrieve relevant documents for a given query, we still have to compute the cosine similarities of the query embedding with the document embeddings of all candidates. Fortunately, the computation overhead of calculating those cosine similarities is significantly lower than encoding query and document pairs with BERT, which contains millions of parameters.

Chapter 4

Modeling Document Interactions for Learning to Rank with Regularized Self-Attention

4.1 Introduction

In chapter 2, we introduce three common approaches to learning to rank: pointwise, pairwise, and listwise approaches. Importantly, these approaches only focus on the loss objective during the *training phase*. Whether the objective is pairwise or listwise, the function $f(q, d_i)$ computes relevance scores for each document d_i independently at *test inference time*. This chapter propose a new formulation for the relevance function based on the set of documents to be ranked: $f(q, d_i, \{d_1, d_2, \ldots, d_n\})$.[1] This is similar to how humans rank documents at test time: multiple competing documents are reviewed before assigning the relevance score to d_i.

Recently, self-attention has been successfully applied to many tasks such as machine translation (Vaswani et al., 2017) and natural language inference (Devlin et al., 2018; Liu et al., 2019; Lan et al., 2019). As self-attention can directly model the connections among elements within a set, it is a suitable mechanism for modeling interactions among documents. Using self-attention on a set of documents allows the model to adjust scores based on other competing documents.

However, experiment results on benchmark datasets show that ListNet with self-attention only performs marginally better. A deeper analysis of at-

[1]The notation will be described more precisely later. For now, the point is to illustrate the difference between modeling $f(q, d_i)$ independently for each d_i, versus adding the full document set in $f(q, d_i, \{d_1, d_2, \ldots, d_n\})$. Suppose some competing documents are dropped, $f(q, d_i)$ will output the same relevance score, whereas $f(q, d_i, \{d_1, d_2, \ldots, d_n\})$ will automatically adapt.

tention weights reveals that self-attention alone is ineffective in modeling document interactions. In section 4.2, we propose regularization terms that can push the model towards learning meaning weights that can better model interactions between documents.

We evaluate our model on both monolingual datasets (Yahoo, MSLR-WEB, and Istella LETOR datasets) and multilingual datasets (Chinese-English, German-English, and Russian-English from CLIRMatrix MULTI-8), and show that neural networks with properly regularized self-attention weights could significantly outperform existing strong ensemble trees and neural network baselines.

4.2 Model Description

Given a query q, a set of documents $\{d_1, d_2, \ldots, d_n\}$ and a feature extraction function ϕ, the input to learning to rank model is a set of feature vectors:

$$D = \{\phi(q, d_1), \phi(q, d_2), \ldots, \phi(q, d_n)\} \tag{4.1}$$

We want to model a ranking function f such that:

$$f(D) \rightarrow [s_1, s_2, \ldots, s_n] \tag{4.2}$$

where s_i is the predicted relevance score for document d_i. In this notation, $f(D)$ now has a vector output of dimension n, where each element represents the relevance score.

Ideally, we want s_i to be sorted in the same order as the desired ranking. We compute all relevance scores at test inference time and then sort the documents according to these scores.

4.2.1 ListNet

Our starting point for modeling f is the ListNet (Cao et al., 2007) algorithm. ListNet is a strong neural learning to rank algorithm which optimizes a listwise objective function. Due to the combinatorial nature of the ranking tasks, popular metrics such as NDCG (Järvelin and Kekäläinen, 2002) and ERR (Chapelle et al., 2009) are not differentiable with respect to model parameters. Consequently, we cannot directly use gradient descent-based learning algorithms for optimization. Therefore, ListNet optimizes a surrogate loss function which is defined below:

Given predicted relevance judgments $f(D) = \{s_1, s_2, \ldots, s_n\}$ and ground truth $R(D) = \{r_1, r_2, \ldots, r_n\}$. The top one probability of document d_j based on f is:

$$P_f(d_j) = \frac{e^{s_j}}{\sum_k e^{s_k}} \tag{4.3}$$

and the top one probability of document d_j based on R is:

$$P_R(d_j) = \frac{e^{r_j}}{\sum_k e^{r_k}} \tag{4.4}$$

Loss is defined as the cross entropy between the top one probability distribution of predicted scores and the top one probability distribution of ground truth:

$$L = -\sum_{i=1}^{n} P_R(d_i) log(P_f(d_i)) \tag{4.5}$$

4.2.2 Self-Attention (SA)

Self-attention is an attention mechanism that learns to represent every element in a set by a weighted sum of every other element within the same set. Self-attention-based neural networks have found success in many NLP tasks such as machine reading, machine translation, and sentence representation learning (Lin et al., 2017; Vaswani et al., 2017; Cheng et al., 2016; Devlin et al., 2018).

The input to the self-attention layer is a set of vector representations:

$$V = [v_1, v_2, \ldots, v_n] \tag{4.6}$$

here v_i is a d-dimensional vector representation of the i-th document, $v_i \in \mathbb{R}^d$. V is a $n \times d$ matrix, which concatenates the vector representation of the n documents. The output of the self-attention layer is:

$$V' = \sigma\left((VW_q)(VW_k)^\intercal\right)(VW_v) \tag{4.7}$$

where $D'_i \in \mathbb{R}^{n \times h}$, $D \in \mathbb{R}^{n \times d}$, $W_q^i, W_k^i, W_v^i \in \mathbb{R}^{d \times h}$ and σ is the sigmoid function. W_q^i, W_k^i and W_v^i are trainable weight matrices.

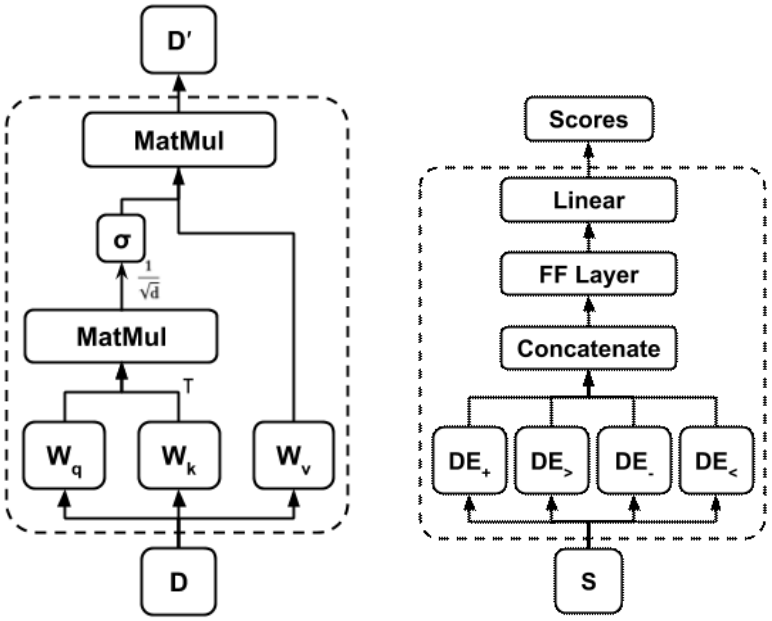
4.2.3 ListNet + Self-Attention (SA)

Figure 4.1: A document encoder consisting of two feed forward layers and a self-attention layer. G_1, G_2, G_3 are highway connections (Srivastava et al., 2015).

ListNet uses a single-layer feed-forward neural network without bias term and nonlinear activation function. We improve the original architecture with recent techniques such as layer normalization (Ba et al., 2016), highway connections (Srivastava et al., 2015) and exponential linear units (Clevert et al., 2015). Inspired by the transformer (Vaswani et al., 2017) architecture, we insert a self-attention layer in the middle of two feed-forward layers. We will refer to this architecture as a document encoder (DE). Figure 4.1 shows the

architecture of document encoder.

4.2.4 ListNet + Regularized Self-Attention (RSA)



(a) Self-attention **(b)** ListNet + RSA

Figure 4.2: Self-attention layer and ListNet + Regularized Self-Attention (RSA).

We observe that specific document interactions are embedded in the datasets: 1) relative orderings between documents and 2) arithmetic differences in relevance judgments between documents. We hypothesize that this information can provide effective supervision for learning the self-attention weights. We explore four different document encoders, each of which is supervised by another regularization term:

- DE_{+} is a document encoder that enhances vector representations of doc-

uments by paying attention to other documents that are more relevant. i.e, for a given document d_i, the attention weight W_{ij}^{+} for d_j is:

$$W_{ij}^{+} = \begin{cases} 1 & \text{if } r_j > r_i \\ 0 & \text{if } r_j <= r_i \end{cases} \tag{4.8}$$

- $\text{DE}_{>}$ is similar to DE_{+} except that it assigns exponentially higher attention weights to documents with higher relevance judgments:

$$W_{ij}^{>} = \begin{cases} \frac{e^{(r_j - r_i)}}{\sum_{i=0}^{k} e^{i}} & \text{if } r_j > r_i \\ 0 & \text{if } r_j <= r_i \end{cases} \tag{4.9}$$

- DE_{-} does the opposite of DE_{+}. It assigns positive attention weights to documents that are less relevant.

$$W_{ij}^{-} = \begin{cases} 1 & \text{if } r_j < r_i \\ 0 & \text{if } r_j >= r_i \end{cases} \tag{4.10}$$

- $\text{DE}_{<}$ is similar to DE_{-}, except that it assigns exponentially higher attention weights to documents with lower relevance judgments:

$$W_{ij}^{<} = \begin{cases} \frac{e^{(r_i - r_j)}}{\sum_{i=0}^{k} e^{i}} & \text{if } r_j < r_i \\ 0 & \text{if } r_j >= r_i \end{cases} \tag{4.11}$$

In equations (4.9) and (4.11), k refers to the maximum relevance judgment. In this paper, k=4 for the monolingual datasets and k=6 for the cross-lingual datasets.

The outputs from the four document encoders are concatenated and then converted to scores via another feed-forward layer. We use the final scores to rank the documents.

4.2.5 Regularization Terms

We introduce regularization terms which encourage the document encoders to learn attention weights close to the values mentioned in equations (4.8), (4.9), (4.10) and (4.11):

Rewrite equation (4.7) as:

$$V' = \Sigma(VW_v) \tag{4.12}$$

where:

$$\Sigma = \sigma\left((VW_q)(VW_k)^\intercal\right) \tag{4.13}$$

Σ is the attention matrix of a document encoder, $\Sigma \in \mathbb{R}^{n \times n}$.

The regularization terms are defined as the average binary cross entropy between the attention weight matrices and the ideal attention weight matrices

defined in equations (4.8), (4.9), (4.10) and (4.11):

$$L_\gamma = -\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}[W_{ij}^{\gamma} log(\Sigma_{ij}^{\gamma}) + (1 - W_{ij}^{\gamma}) log(1 - \Sigma_{ij}^{\gamma})] \tag{4.14}$$

for $\gamma \in \{+, >, -, <\}$.

The final objective function is the summation of the ListNet loss function and the regularization terms:

$$L = L + L_+ + L_> + L_- + L_< \tag{4.15}$$

4.3 Experiment Setup

4.3.1 Datasets

MONOLINGUAL DATASETS

We conduct evaluations on the Yahoo LETOR (Chapelle and Chang, 2011), MSLR-WEB30K (Qin and Liu, 2013) and Istella LETOR (Dato et al., 2016) datasets shown in table 4.1. We also include results on the MSLR-WEB10K and Istella-S LETOR datasets, sampled from MSLR-WEB30K and Istella LETOR, respectively. Due to privacy regulations, all datasets only contain extracted feature vectors, and raw texts of queries and documents are not publicly avail-

Table 4.1: Characteristics of the datasets.

Dataset	Year	#Feats	Type	#Q	#D	Avg # D/Q
Yahoo LETOR	2010	700	Train	20K	473K	23.7
			Validation	3K	71K	23.7
			Test	6.9K	166K	23.7
MSLR-WEB10K	2010	136	Train	6K	723K	120.6
			Validation	2K	235K	117.6
			Test	2K	242K	120.8
MSLR-WEB30K	2010	136	Train	19K	2.3M	120.0
			Validation	6K	747K	118.5
			Test	6.3K	754K	119.5
Istella-S LETOR	2016	220	Train	19K	2.0M	106.17
			Validation	7.2K	684K	118.5
			Test	6.6K	681K	103.8
Istella LETOR	2016	220	Train	17K	5.5K	315.0
			Validation	5.9K	1.9M	316.9
			Test	9.8K	3.1M	319.3
CLIRMatrix Multi-8	2020	20	Train	10K	1M	100
			Validation	1K	100K	100
			Test	1K	100K	100

able. Every monolingual dataset has five levels of relevance judgment, from 0 (not relevant) to 4 (highly relevant), and every cross-lingual dataset has seven levels of relevance judgment, from 0 (not relevant) to 6 (highly relevant).

MULTILINGUAL DATASETS

Table 4.2: Features extracted from CLIRMatrix MULTI-8 datasets

Feature name	Description
Covered query term number	Number of terms in query covered by the document.
Covered query term ratio	Covered query term number divide by number of query terms
Document length	Number of document terms
Inverse document frequency	Mean IDF of query terms
Sum/Min/Max/Max/Var of term frequencies	Aggregation features of query term frequencies
Sum/Min/Max/Max/Var of normalized term frequencies	Aggregation features of query term frequencies normalized by query lengths
Sum/Min/Max/Max/Var of tf-idf	Aggregation features of the product of term frequencies and inverse document frequencies
BM25	The BM25 score of query-document pair

We further extend our experiments to three language pairs from the CLIRMatrix MULTI-8 datasets (Chapter 7): Chinese→English, German→English and Russian→English. For every language pair, we first train a neural machine translation system based on the transformer architecture (Vaswani et al., 2017) on parallel sentences from OPUS (Tiedemann, 2012) and then translate

the queries into the same language as the documents.

For every translated query-document pair, we extract 20 features as shown in Table 4.2. These features are similar to those in MSLR-WEB10K and MSLR-WEB30K LETOR datasets.

4.3.2 Baseline Systems and Parameters Tuning

We implemented all neural models with PyTorch[2]. We also provide results of two strong learning to rank algorithms based on ensembles of regression trees: MART (Friedman, 2002) and LambdaMART (Burges, 2010). We use RankLib[3] to train and evaluate these models and did hyperparameter tuning on the number of trees and the number of leaves per tree.

Models with the highest NDCG@10 scores on validation sets were used to obtain final results on test sets, and significance tests were conducted using paired t-tests.

4.3.3 Evaluation Metrics

We consider two popular ranking metrics which support multiple levels of relevance judgment:

[2]https://pytorch.org/

[3]https://sourceforge.net/p/lemur/wiki/RankLib/
We omit the results of other learning to rank algorithms in Ranklib as they perform significantly worse than MART and LambdaMART.

1. **Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen, 2002)** sums relevance judgments (gain) of ranked documents, which are discounted by their positions in ranking and normalized by the discounted cumulative gain of the ideal documents ordering.

2. **Expected Reciprocal Rank (ERR) (Chapelle et al., 2009)** measures the expected reciprocal rank at which a user will stop his search.

We report results at positions 1, 3, 5, and 10 for both metrics.

4.4 Results and Analysis

4.4.1 Results

Table 4.3: Evaluation results for Yahoo LETOR dataset. * and + indicate results which are statistically significant different from the results of ListNet + RSA at $p<0.01$ and $0.01 \leq p < 0.05$ respectively.

Algorithm	E@1	N@1	E@3	N@3	E@5	N@5	E@10	N@10
MART	.344	.684	.420	**.688**	.440	**.707**	**.455**	**.747**
LambdaMART	**.346**	**.687**	**.421**	.685	**.441**	.704	**.455**	.747*
ListNet	.339	.670	.414	.676	.435	.697	.450	.742*
ListNet + SA	.338*	.672*	.413*	.677*	.434*	.697*	.449*	.742*
ListNet + RSA	.342	.673	.417	.684	.438	**.707**	.453	.745

Tables 4.3, 4.4a, 4.4b, 4.5a, 4.5b, 4.6a, 4.6b and 4.6c present our main results on the datasets. We observe that ListNet with an additional self-attention

Table 4.4: Evaluation results on MSLR datasets.

(a) Results on MSLR-WEB10K

Algorithm	**E@1**	**N@1**	**E@3**	**N@3**	**E@5**	**N@5**	**E@10**	**N@10**
MART	.217*	.415*	.297*	.416*	.321*	.426*	.340*	.448*
LambdaMART	.226*	.428*	.306*	.424*	.329*	.431*	.348*	.451*
ListNet	.211*	.410*	.285*	.398*	.308*	.402*	.328*	.429*
ListNet + SA	.213*	.402*	.290*	.405*	.313*	.410*	.332*	.431*
ListNet + RSA	**.231**	**.439**	**.310**	**.435**	**.332**	**.438**	**.350**	**.457**

(b) MSLR-WEB30K

Algorithm	**E@1**	**N@1**	**E@3**	**N@3**	**E@5**	**N@5**	**E@10**	**N@10**
MART	.222*	.436+	.303*	.426	.328	.435	.347*	.457*
LambdaMART	.240*	.458+	.321*	.446	.344*	.451*	.363*	.471*
ListNet	.229*	.429*	.307*	.424*	.330*	.429*	.348*	.449*
ListNet + SA	.227*	.429*	.304*	.422*	.327*	.427*	.346*	.449*
ListNet + RSA	**.249**	**.464**	**.326**	**.452**	**.349**	**.457**	**.367**	**.478**

Table 4.5: Evaluation results for Istella datasets.

(a) Istella-S LETOR

Algorithm	**E@1**	**N@1**	**E@3**	**N@3**	**E@5**	**N@5**	**E@10**	**N@10**
MART	.556*	.625*	.674*	.605*	.692*	.633*	.700*	.704*
LambdaMART	.587*	.658*	.698*	.631*	.715*	.656*	.721*	.719*
ListNet	.586*	.657*	.698*	.632*	.714*	.658*	.720*	.719*
ListNet + SA	.592*	.663*	.703*	.635*	.719*	.662*	.725*	.723*
ListNet + RSA	**.599**	**.671**	**.711**	**.649**	**.726**	**.676**	**.732**	**.739**

(b) Istella LETOR

Algorithm	**E@1**	**N@1**	**E@3**	**N@3**	**E@5**	**N@5**	**E@10**	**N@10**
MART	.563*	.621*	.668*	.567*	.686*	.584*	.695*	.632*
LambdaMART	.594*	.654*	.696*	.596*	.712*	.610*	.719*	.657*
ListNet	.576*	.633*	.681*	.578*	.696*	.589*	.704*	.632*
ListNet + SA	.591*	.650*	.696*	.596*	.711*	.610*	.718*	.656*
ListNet + RSA	**.604**	**.665**	**.709**	**.615**	**.724**	**.629**	**.730**	**.678**

Table 4.6: Evaluation results on CLIRMatrix MULTI-8 datasets.

(a) Results for Chinese→English

Algorithm	**E@1**	**N@1**	**E@3**	**N@3**	**E@5**	**N@5**	**E@10**	**N@10**
MART	.319*	.324*	.415*	.375*	.441*	.400*	.459*	.432*
LambdaMART	.338*	.344*	.434*	.390*	.460*	.416*	.477*	.445*
ListNet	.332*	.337*	.427*	.375*	.443*	.403*	.468*	.437*
ListNet + SA	.336*	.341*	.431*	.382*	.454*	.413*	.473*	.445*
ListNet + RSA	**.342**	**.347**	**.441**	**.396**	**.467**	**.420**	**.480**	**.453**

(b) German→English

Algorithm	**E@1**	**N@1**	**E@3**	**N@3**	**E@5**	**N@5**	**E@10**	**N@10**
MART	.499*	.507*	.613*	.568*	.632*	.591*	.642*	.618*
LambdaMART	.509*	.517*	.621*	.574*	.641*	.600*	.651*	.626*
ListNet	.519*	.526*	.633*	.581*	.651*	.601*	.657*	.629*
ListNet + SA	.524*	.532*	.637*	.589*	.654*	.610*	.663*	.635*
ListNet + RSA	**.533**	**.542**	**.642**	**.593**	**.658**	**.614**	**.668**	**.641**

(c) Russian→English

Algorithm	**E@1**	**N@1**	**E@3**	**N@3**	**E@5**	**N@5**	**E@10**	**N@10**
MART	.400*	.406*	.504*	.462*	.526*	.486*	.540*	.519*
LambdaMART	.425*	.432*	.528*	.484*	.550*	.506*	.563*	.535*
ListNet	.422*	.418*	.530*	.477*	.545*	.501*	.555*	.526*
ListNet + SA	.430*	.437*	.532*	.486*	.553*	.509*	.567*	.538*
ListNet + RSA	**.443**	**.450**	**.540**	**.493**	**.561**	**.514**	**.574**	**.543**

layer is marginally better than ListNet without a self-attention layer.

However, ListNet with regularized self-attention consistently achieves solid ERR@i and NDCG@i scores at various positions i. In particular, ListNet with regularized self-attention is the single best system in all metrics measured in seven out of the eight datasets. For example, in MSLR-WEB10K, our model achieves 0.231 ERR@1, 0.439 NDCG@1, 0.350 ERR@10, and 0.457 NDCG@10, all outperforming the next-best model LambdaMART (which achieved 0.226 ERR@1, 0.428 NDCG@1, 0.348 ERR@10, 0.451 NDCG@10). This trend holds for the MSLR-Web30K, Istella-S, Istella, and the CLIRMatrix datasets. The only exception where ListNet + RSA does not win on all metrics is the Yahoo LETOR datasets: but even there, ListNet + RSA ranks second or third in most cases and still outperforms on NDCG@10.

These consistent improvements confirm that the proposed regularized self-attention mechanism effectively improves learning to rank results.

4.4.2 Impact of Regularization Terms

Figure 4.3 shows the plots of NDCG@10 scores against training epochs on all validation sets. As seen in the plots, the curves of models with regularization terms are almost always above those without regularization terms. Further, the former always converges to significantly higher NDCG@10 values. These phenomenons clearly show that our proposed regularization terms effec-

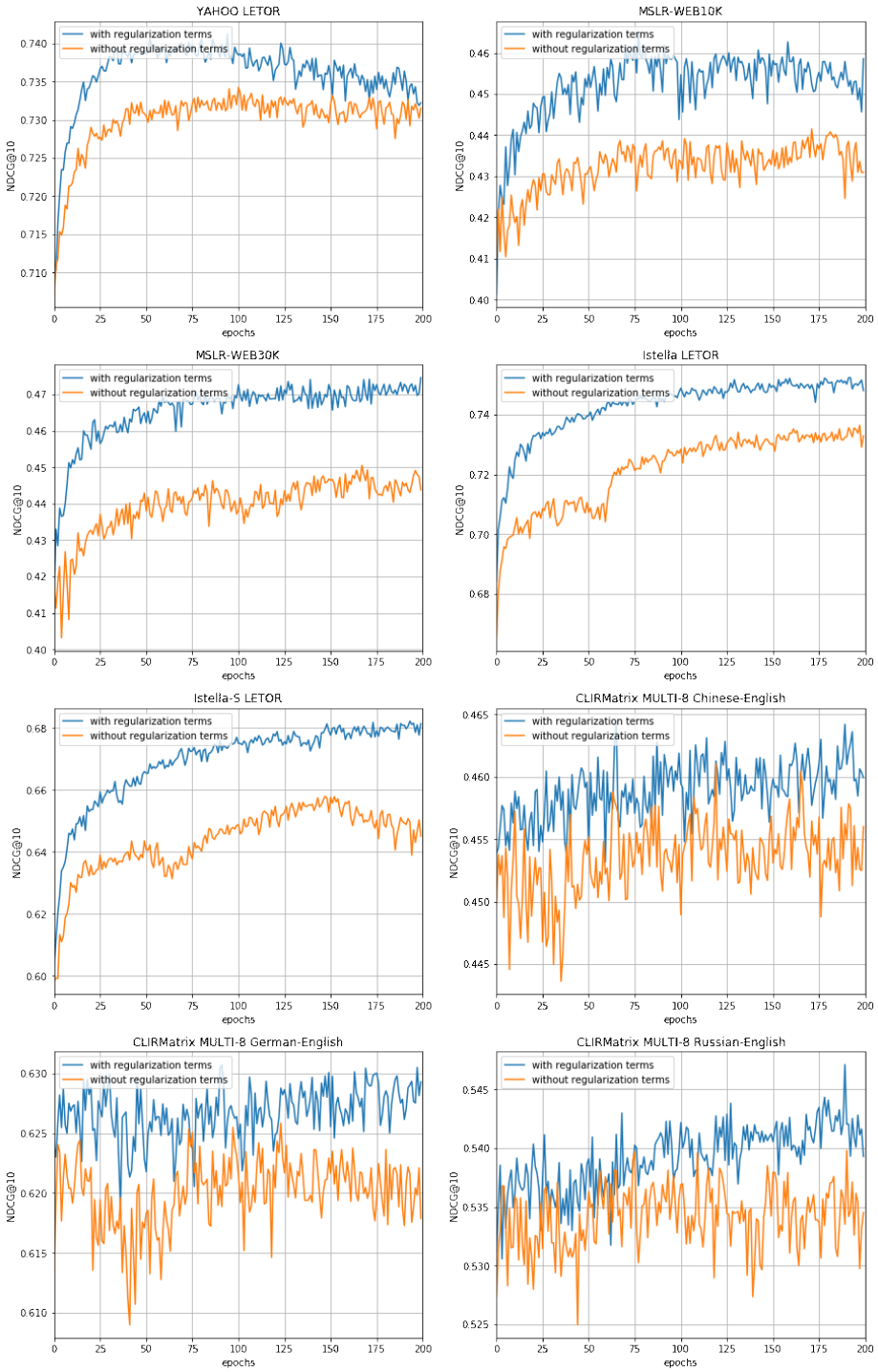


Figure 4.3: Plots of NDCG@10 scores against training epochs on all validation sets. Curves of models with regularization terms are almost always above the curves of models without regularization terms.

tively improve the performance of ListNet with the self-attention layer. Models without the regularization terms perform worse than MART and LambdaMART on all datasets.

4.4.3 Attention Visualization

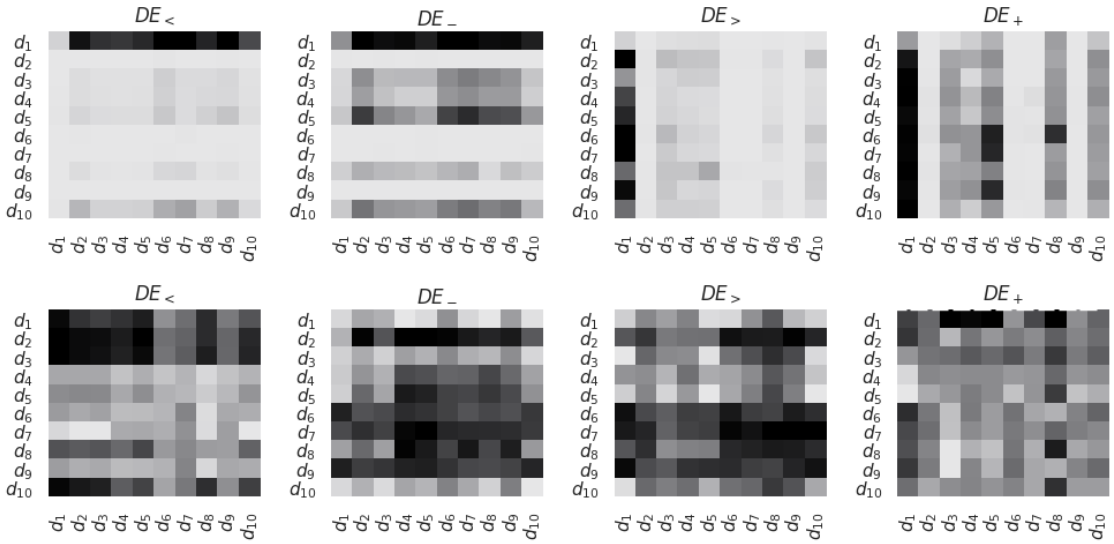


Figure 4.4: Top row: attention weights matrices. Bottom row: attention weights matrices without regularization terms. The relevance judgments of the documents for this sample query are $d_1 = 3$, $d_2 = 0$, $d_3 = 0$, $d_4 = 1$, $d_5 = 3$, $d_6 = 0$, $d_7 = 0$, $d_8 = 1$, $d_9 = 0$ and $d_{10} = 3$.

We sample query and document pairs from the Istella-S dataset and plot the heatmaps of the attention weights of the four different document encoders in Figure 4.4.

The bottom row of Figure 4.4 shows the attention heatmaps of a model trained without the regularization terms. We are unable to observe any ex-

plainable pattern in the attention matrices. From the results of our experiments, self-attention alone is not effective at figuring out attention weights that are useful for modeling document interactions.

In contrast, the top row of the visualization suggests that our model can learn better attention weights with the supervisions from the regularization terms: $DE_{<}$ and DE_{-} place more attention weights on *rows* with higher relevance judgments, while $DE_{>}$ and DE_{+} place more attention weights on *columns* with higher relevance judgments.

4.4.4 Impact of the Document Encoders

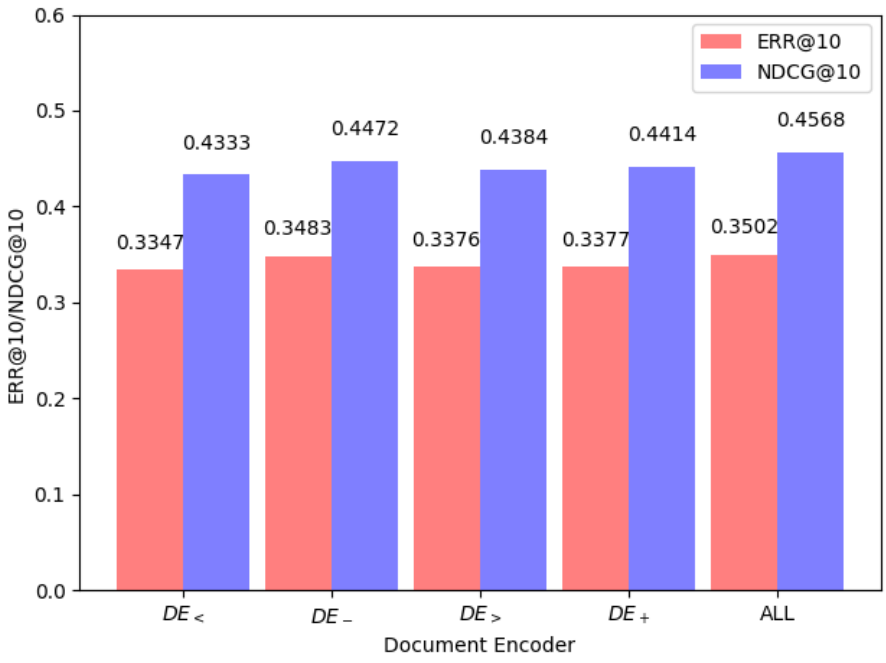


Figure 4.5: ERR@10 and NDCG@10 scores on the MSLR-WEB10K test set for different document encoders.

We train four separate models, each using only one of the four document encoders. Figure 4.5 presents our results.

We observe that DE_+ or DE_- perform better than $DE_>$ and $DE_<$, even though $DE_>$ and $DE_<$ are trained to put exponentially higher attention weights on documents with higher relevance judgments. We suspect the regularization terms for $DE_>$ and $DE_<$ are more challenging to optimize than the regularization terms for DE_+ and DE_-.

We also observe that the NDCG@10 score from a model with four document encoders is around 1 to 2.3 points higher than the models with individual document encoders. This shows that ensembling the document encoders effectively improves results in learning to rank tasks.

4.5 Related Work

There are two general research directions in learning to rank. In the traditional setting, machine learning algorithms are employed to re-rank documents based on preprocessed feature vectors. In the end-to-end setting, models are designed to extract features and rank documents simultaneously.

4.5.1 Traditional Learning to Rank

As there can be tens of millions of candidate documents for every query in real-world contexts, information retrieval systems usually employ a two-phase approach. In the first phase, a smaller set of candidate documents are shortlisted from the bigger pool of documents using simpler models such as vector space model (Salton et al., 1975) and BM25 (Robertson et al., 2009). In the second phase, shortlisted documents are converted into feature vectors, and more accurate learning to rank models are used to re-rank the feature vectors. Examples of commonly used features are term frequencies, BM25 scores, URL click rate, and length of documents.

Predicting the relevance scores of documents in pointwise approaches can be treated as a regression problem. Popular regression algorithms such as (Breiman, 2001; Friedman, 2002) are often directly used to estimate relevance judgments of documents. Pairwise approaches such as (Adomavicius and Tuzhilin, 2005; Joachims, 2002; Burges et al., 2007) treat learning to rank as a binary classification problem. Ensemble trees are generally recognized as the strongest systems, e.g. an ensemble of LambdaMART and other lambda-gradient models (Burges et al., 2011) won the Yahoo Learning to Rank challenge (Chapelle and Chang, 2011). Neural networks such as RankNet (Burges et al., 2005) and ListNet (Cao et al., 2007) are also effective. The common theme in these papers is to learn a classifier that can determine the correct ordering given a pair of

documents.

Optimizing listwise objectives can be difficult because popular IR metrics such as MAP, ERR, and NDCG are not differentiable, so we cannot directly use gradient descent-based methods for optimization. Various surrogate loss functions have been proposed over the years. For example, Cao et al. (2007) proposed ListNet, which uses the cross-entropy between the permutation probability distributions of the predicted ranking and the ground truth as a loss function. Taylor et al. (2008) proposed SoftRank, which uses smoothed approximations to ranking metrics. Burges (2010) proposed LambdaRank and LambdaMART, which approximate gradients by the directions of swapping two documents, scaled by the change in ranking metrics. Although these loss functions demonstrate various degrees of success in learning to rank tasks, most of the papers only use them for training global ranking models that independently predict the relevance scores of every document. In contrast, we design our model to explicitly model the interdependence between documents. We can replace our ListNet loss function with any existing loss objectives.

More recently, Ai et al. (2018) proposed the DLCM, which uses a recurrent neural network to sequentially encode documents in the order returned by strong baseline learning to rank algorithms such as LambdaMART. The authors find that incorporating local ranking context can further fine-tune the initial results of baseline systems. Unlike DLCM, which relies on the ranking

results from other learning to rank algorithms, our model is a self-contained learning to rank algorithm. Therefore, a direct comparison between DLCM and our model is not possible.

4.5.2 End-to-End Learning to Rank

As traditional learning to rank systems relies heavily on handcrafted feature engineering that can be tedious and often incomplete, there is growing interest in end-to-end learning to rank tasks among NLP and IR researchers. Systems in this category focus on generating feature vectors automatically using deep neural networks without extracting feature vectors.

End-to-end models can be further classified under two broad categories: 1) representation-based models and 2) interaction-based models. Representation-based models try to generate good representations of queries and documents independently before conducting relevance matching, e.g., Huang et al. (2013b); Hu et al. (2014). In contrast, interaction-based models focus on learning local interactions between query text and document text before aggregating the local matching signals, e.g., Guo et al. (2016b); Pang et al. (2017); McDonald et al. (2018).

Since the models above focus primarily on learning better vector representations of query-document pairs from raw texts, the output representations from those models can be directly fed as inputs to our model, which is designed

to learn the interactions among the documents. As end-to-end learning to rank is not the focus of this paper, we will explore end-to-end models in future work.

4.6 Conclusion

This chapter explores the possibility of modeling document interactions with the self-attention mechanism. Experiments on benchmark datasets show that neural learning to rank models only performs marginally better with additional self-attention layers. A deeper analysis of attention weights reveals that self-attention alone is ineffective in modeling document interactions. Therefore, this chapter proposes additional regularization terms that further supervise the learning of the attention weights. Our proposed regularized self-attention ranking network (RSARN) outperforms several strong baseline models on both monolingual datasets (Yahoo, MSLR-WEB, Istella LETOR) and multilingual datasets (CLIRMatrix MULTI-8). This shows that our method is effective and can potentially improve the performance of information retrieval systems in both monolingual and cross-lingual settings.

Chapter 5

Cross-Lingual Learning-to-Rank with Shared Representations

5.1 Introduction

As discussed in Chapter 2, there are two main approaches to building CLIR systems. The *modular approach* involves a pipeline of two components: translation (machine translation or bilingual dictionary look-up) and monolingual information retrieval (IR). A distinctly different way is the *direct modeling approach* Bai et al. (2010); Sokolov et al. (2013) that attempts to build neural learning to rank models in an end-to-end manner.

Direct modeling is advantageous because it focuses on learning beneficial translations for retrieval, rather than translations that preserve sentence meaning/structure in bitext. However, this approach generally comprises deep neural networks that require a large amount of data for supervised training. There is no existing large-scale CLIR dataset that can support direct modeling approaches in various languages.

In this chapter, we present a large-scale dataset constructed automatically from Wikipedia, which can support training and evaluation of CLIR systems between English queries and documents in 25 other languages (Section 5.2). The data is sufficient for direct modeling and can serve as wide-coverage evaluation data for the modular approaches.[1]

To demonstrate the utility of the data, we further present experiments for

[1]To facilitate CLIR research, the dataset is publicly available at http://www.cs.jhu.edu/~kevinduh/a/wikiclir2018/.

CLIR in low-resource languages. First, we introduce a neural CLIR model based on the direct modeling approach (Section 5.3.1). We then show how we can bootstrap CLIR models for languages with less training data by appropriate use of parameter sharing among different language pairs (Section 5.3.2). For example, using the training data for Japanese-English CLIR, we can improve the Mean Average Precision (MAP) results of a Swahili-English CLIR system by 5-7 points (Section 5.4).

5.2 Large-Scale CLIR dataset

Inspired by Schamoni et al. (2014a) who made an English-German CLIR dataset from Wikipedia, we want to extend the same idea to build a larger dataset that covers many more language directions. The general idea is to exploit *inter-language links*, defined as the links on one Wikipedia article in some language to equivalent Wikipedia articles in other languages. The raw texts and inter-language links can be extracted from the Wikipedia backup dumps. We created this dataset using Wikipedia dumps, released on August 23, 2017.

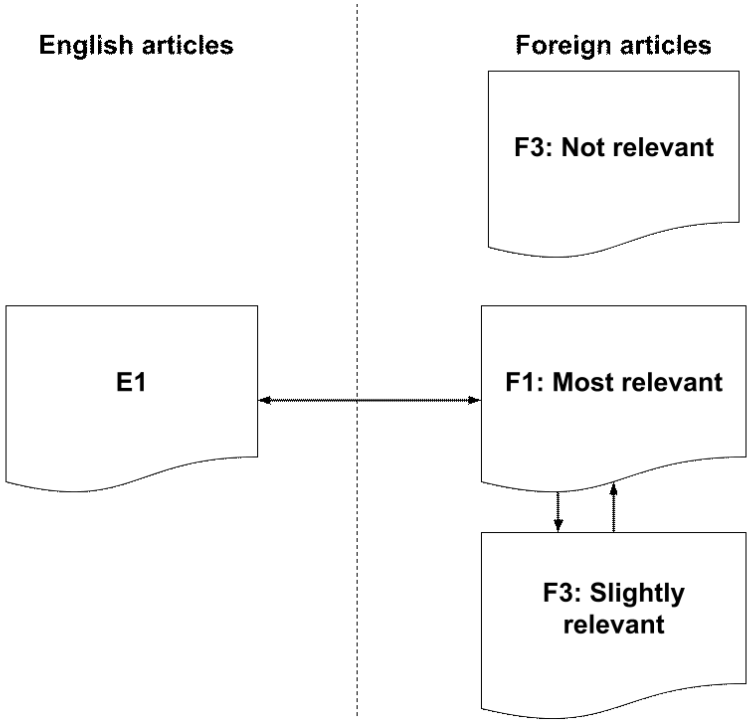


Figure 5.1: CLIR data construction process: From an English article (E1), we extract the English query. Using the inter-language link, we obtain the *most relevant* foreign-language document (F1). Any article that has mutual links to and from F1 are labeled as *slightly relevant* (F2). All other articles are *not relevant* (F3). The data is a set of tuples: (English query q, foreign document d, relevance judgment r), where $r \in \{0, 1, 2\}$ represents the three levels of relevance.

5.2.1 Construction Process

Figure 5.1 shows the construction process of this dataset. First, we obtain English queries by extracting the first sentence of every English Wikipedia article. The intuition is that the first sentence is usually a well-defined summary of its corresponding article and should be thematically related to articles linked to it from another language. Similar to Schamoni et al. (2014a), we removed the title words from the query sentences because they may be present across

different language editions. This deletion prevents the task from becoming an easy keyword matching task.

For practical purposes, we limit each document to the first 200 words of the article. Empty documents and category pages are filtered. This dataset consists of more than 2.8 million English queries and relevant documents from 25 other selected languages (see Table 1.2).

In summary, we have created a large-scale CLIR dataset in terms of both the number of examples and the number of languages. We can use this dataset in two scenarios: (1) one mixed-language collection where an English query may retrieve relevant documents in multiple languages. (2) 25 independent datasets for training and evaluating CLIR on English queries against one foreign language collection. In the experiments in Section 5.4, we will utilize the dataset in terms of scenario (2).[2]

5.3 Direct Modeling for CLIR

This section first describes a neural ranking model that adopts the direct modeling approach. Then, we extend that model by introducing a new method that shares representations across languages.

[2]For extensibility purposes, these experiments use only half of the data, randomly sampled by the query (the held-out data is reserved for other uses). Also it only considers binary relevance (*most relevant* vs *not relevant*) for simplicity. The exact data splits will be provided along with the data release.

5.3.1 Neural Ranking Model

Given an English query q^X and a foreign-language document d^Y, our models compute the relevance score $f(q^X, d^Y)$. First, we represent each word as a n-dimensional vector, so q^X and d^Y are represented as matrices $\mathbf{Q^X} \in \mathbb{R}^{n \times |q^X|}$ and $\mathbf{D^Y} \in \mathbb{R}^{n \times |d^Y|}$, where $|q^X|$ and $|d^Y|$ are the numbers of tokens in q^X and d^Y:

$$\begin{aligned} \mathbf{Q} &= [E_q(q_1^X); E_q(q_2^X); ...; E_q(q_{|q^X|}^X)] \\ \mathbf{D} &= [E_d(d_1^Y); E_d(d_2^Y); ...; E_d(d_{|d^Y|}^Y)] \end{aligned}$$

q_i^X and d_i^Y denote the i-th term in q^X and d^Y. E is an embedding function that transforms each term to a dense n-dimensional vector as its representation. ; is the concatenation operator. Then, we apply a convolutional feature map[3] to these matrices, followed by tanh activation and average-pooling to obtain each representation vector $\hat{q^X}$ and $\hat{d^Y}$.

$$\hat{q^X} = CNN_q(\mathbf{Q^X}); \quad \hat{d^Y} = CNN_d(\mathbf{D}^Y) \tag{5.1}$$

Next, we define two variations in calculating $f(q^X, d^Y)$. The first is a *cosine model* which computes cosine similarity between $\hat{q^X}$ and $\hat{d^Y}$:

$$S_{cos}(q^X, d^Y) = sim(\hat{q^X}, \hat{d^Y}) \tag{5.2}$$

[3]The $n \times 4$ convolution window has a filter size of 100 and a stride of 1.

The second is a *deep model* with a fully connected layer on top of the concatenation of $q^{\hat{X}}$ and $d^{\hat{Y}}$ (a 200-dimensional vector):

$$\begin{aligned} S_{deep}(q^X, d^Y) &= tanh(O \cdot h_{vec}^{\mathrm{T}}) \\ &= tanh(O \cdot relu(W \cdot [q^{\hat{X}}; d^{\hat{Y}}]^{\mathrm{T}})) \end{aligned} \tag{5.3}$$

Here, $O \in \mathbb{R}^{1 \times h}$ and $W \in \mathbb{R}^{h \times 200}$ are the deep model parameters, and h is the number of dimensions of the hidden state, $h_{vec} \in \mathbb{R}^{1 \times h}$. For regularization, we set the dropout rate as 0.5 (Srivastava et al., 2014) at the hidden layer.

In the training phase, we minimize pairwise ranking loss, which is widely used for learning-to-rank (Pang et al., 2016; Guo et al., 2016a; Hui et al., 2017; Xiong et al., 2017; Dehghani et al., 2017a), defined as follows:

$$L = max\left\{0, 1 - (S(q^X, d^{Y+}) - S(q^X, d^{Y-}))\right\} \tag{5.4}$$

where d^{Y+} and d^{Y-} are relevant and non-relevant document respectively. We fix only the word embeddings and tune the other parameters.

We note that many other ranking models can be adapted to CLIR (Huang et al., 2013a; Shen et al., 2014; Xiong et al., 2017; Mitra et al., 2017); they have a common framework for extracting features from both query and document and optimizing scores $f(q^X, d^Y)$ via some ranking loss.

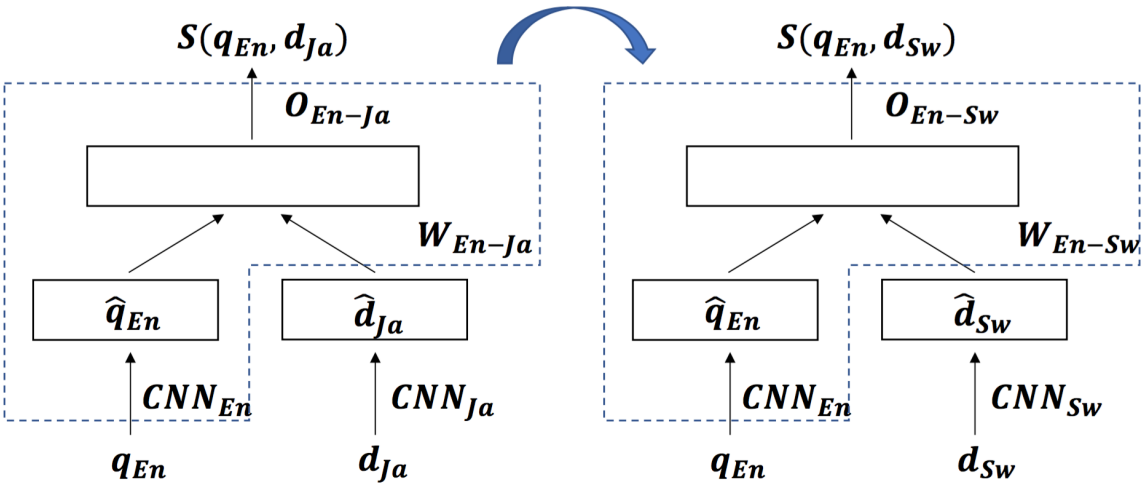


Figure 5.2: Illustration of the proposed method. On low resource dataset (e.g. Swahili-English), the parameters of the CNN for encoding query (CNN_{En}) and the parameters of the fully connected layer (O_{En-Sw}, W_{En-Sw}) are initialized by the ones pre-trained on high resource dataset (e.g. Japanese-English).

5.3.2 Sharing Representations

Training a network like the deep model generally requires a non-trivial amount of data. We propose a simple yet effective method that shares representations across CLIR models trained in different language pairs to address the data required for low-resource languages. Basically, we use the same architecture as the deep model ($f_{deep}(q^X, d^Y)$, Equation 5.3). However, we use the parameters trained on a high-resource dataset (e.g., Japanese-English) to initialize the parameters for a low-resource language pair (e.g., Swahili-English).

Figure 5.2 illustrates the idea: Concretely, we initialize the parameters of the CNN for encoding query (CNN_q) and the parameters of the fully connected layer (O, W) by using the pre-trained parameters. When training on

low-resource data, we fix only the word embedding and tune the parameters of CNNs and the fully connected layer.

The intuition behind this is that our direct modeling approach enforces $\hat{q^X}$ and $\hat{d^Y}$ to become language-independent representations of the query and document. The parameters O and W in the deep layer can be used for any language pair. Note that for the cosine model, we can also share parameters for CNN_q.

5.4 Experiment Results

Setup: We use datasets of 3 high-resource languages (Japanese [Ja], German [De], French [Fr]) and 2 low-resource languages (Tagalog [Tl], Swahili [Sw]). We also subsample German and French data to be equivalent to the size of Swahili to compare training size effects. Word embedding with dimension 100 for each language is trained on Wikipedia corpus, using word2vec SGNS (Mikolov et al., 2013b). The size of hidden states in the deep model is {100, 200, 300, 400, 500}. We adopt Adam (Kingma and Ba, 2014) for optimization, train for 20 epochs, and pick the best epoch based on development set loss. For the proposed method of parameter sharing, we use the weight parameters pre-trained on a Japanese-English dataset to initialize parameters.

High-resource results: Table 5.1 shows the P@1 (precision at top position) and MAP (mean average precision) for datasets consisting of on the order of

	Ja	De	Fr
$S_{cos}(q^X, d^Y)$: cos	.59/.74	.49/.66	.55/.70
$S_{deep}(q^X, d^Y)$: h=100	.61/.75	.64/.77	.69/.81
$S_{deep}(q^X, d^Y)$: h=200	.68/.80	.67/.79	.74/.84
$S_{deep}(q^X, d^Y)$: h=300	.70/.82	.70/.81	.74/.84
$S_{deep}(q^X, d^Y)$: h=400	**.73**/.83	**.71/.82**	.75/**.85**
$S_{deep}(q^X, d^Y)$: h=500	**.73/.84**	.70/.81	**.76/.85**

Table 5.1: P@1/MAP performance of the cosine model and the deep model with different hidden state size on **high resource datasets**. Best value in each column is highlighted in bold.

100k+ training queries. The deep models outperformed the cosine models under all conditions, suggesting that the fully connected layer can exploit the large training set in learning more expressive scoring functions.

Low-resource results: Table 5.2 shows the effects on the low resource datasets under two conditions: training on only the language pair of interest (in-language) or additionally sharing parameters using a pre-trained Japanese-English model. We observe that the cosine model outperforms the deep model for the in-language case. In contrast to the high-resource results, deep models with many parameters only become effective if provided with sufficient training data.

For the sharing case, the deep models with parameter sharing outperformed the basic deep models trained only on in-language data under almost all conditions. This indicates that our sharing method reduces the training data required. Importantly, the deep models can now outperform the cosine model and

		cos	h=100	h=200	h=300	h=400	h=500
Tl	In	.51/.68	.34/.50	.44/.58	.42/.57	.49/.63	.51/.64
	Sh	.50/.67	.48/.62	.55/.67	.49/.63	**.57/.69**	.54/.67
	Δ	-/-	+/+	+/+	+/+	+/+	+/+
Sw	In	.51/.67	.46/.62	.47/.63	.50/.65	.51/.66	.53/.68
	Sh	.49/.65	.46/.62	.52/.67	.58/.70	**.60/.73**	.56/.69
	Δ	-/-	=/=	+/+	+/+	+/+	+/+
De (subsample)	In	.40/.59	.39/.55	.41/.57	.44/.60	.45/.61	.44/.60
	Sh	.38/.56	.46/.62	.48/.63	.50/.65	**.51/.66**	.49/.65
	Δ	-/-	+/+	+/+	+/+	+/+	+/+
Fr (subsample)	In	.46/.63	.40/.57	.43/.60	.49/.65	.47/.64	.47/.63
	Sh	.43/.60	.46/.62	.51/.66	.51/.66	**.56/.70**	.51/.66
	Δ	-/-	+/+	+/+	+/+	+/+	+/+

Table 5.2: P@1/MAP performances on **low resource datasets**. Δ columns show the comparison between the basic deep models with in-language training (In) and the deep models with sharing parameters (Sh); + indicates Sh outperforms In, and - indicates the In outperforms Sh. Best value in each dataset is highlighted in bold.

achieve the best results on all datasets by sharing parameters.[4]

5.5 Conclusion

This chapter introduces the large-scale CLIR dataset containing English queries and foreign documents in 25 languages, enabling the training and evaluation of direct modeling approaches in CLIR. We further experiment with

[4]Sharing representations with the cosine models did not help; we hypothesize that cross-lingual sharing only works if given sufficient model expressiveness. We also tried the shared deep models on high resource datasets (e.g., using Japanese parameters on the full French dataset without subsampling). As expected, the results did not change significantly.

CHAPTER 5. CROSS-LINGUAL LEARNING-TO-RANK WITH SHARED REPRESENTATIONS

CLIR models with shared representation based on Convolutional Neural networks and demonstrate its effectiveness in bootstrapping CLIR in low-resource languages.

Chapter 6

An Empirical Study on the Feasibility of Multilingual BERT in Cross-Lingual Information Retrieval

6.1 Introduction

Recently, models based on contextual embeddings such as BERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2019) demonstrated significant improvements on a variety of NLP tasks. In particular, MacAvaney et al. (2019) show that fine-tuning *contextual embeddings* such as BERT achieves state-of-the-art results on several monolingual (English) IR tasks.

In this empirical study, we investigate whether *multilingual* BERT can be exploited similarly to achieve state-of-the-art results on CLIR tasks. We experiment with five language pairs from the Large-Scale Wikipedia CLIR dataset (Sasaki et al., 2018) and show that these embeddings are highly effective in both standard CLIR tasks and zero-shot cross-lingual transfer settings. Our results show that a simple CLIR ranker model based on multilingual BERT can outperform state-of-the-art systems with minimal supervision. Further analysis shows that these BERT ranker models are robust and do not suffer from the partial-input baseline problems observed in other tasks (Poliak et al., 2018; Gururangan et al., 2018).

6.2 Experiment Setup

6.2.1 Dataset

We conduct our experiments on the Large-Scale Wikipedia CLIR Dataset[1] introduced in the previous chapter. We use the same data splits as the experiments in section 5.4 and report results on three high-resource target languages (Japanese [Ja], German [De], French [Fr]) and two low-resource target languages (Tagalog [Tl] and Swahili [Sw]). The source language is English for all five datasets, and the statistics are shown in Table 6.1.

Split		Ja	De	Fr	Tl	Sw
train	#q	162K	343K	395K	17K	8.5K
dev	#d	510K	1M	935K	37K	18K
test	#q	21K	41K	51K	2.3K	1.1K
	#d	425K	835K	850K	34K	17K

Table 6.1: Number of queries (#q) and documents (#d) of selected languages from the Large-Scale Wikipedia CLIR Dataset.

6.2.2 Baseline CLIR Model

We include the results of the CLIR models in section 5.4 for comparison. The models are based on convolution neural networks and initialized with word embeddings pre-trained on Wikipedia documents. The previous results

[1] http://www.cs.jhu.edu/~kevinduh/a/wikiclir2018/

demonstrate that basic CLIR models trained on only in-language (In) data can perform significantly better if they initialize these models with shared parameters (Sh) from models pre-trained on high-resource languages.

6.2.3 BERT Ranker Model

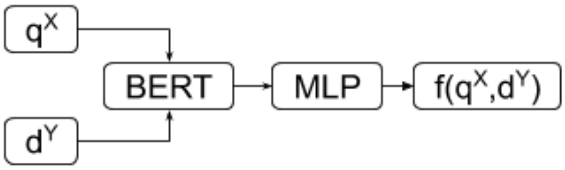


Figure 6.1: Multilingual BERT ranker model.

Inspired by the findings in section 5.4, we investigate whether CLIR can also benefit from sharing the parameters of multilingual BERT, which was pre-trained on significantly more language resources. We follow the implementation of the vanilla BERT ranker model (MacAvaney et al., 2019) as seen in Figure 6.1, which encodes a query–document pair with multilingual BERT (Devlin et al., 2019) and then converts the encoding into a similarity score by stacking a linear combination layer on top of the [CLS] token. We then fine-tune this model on supervised data containing queries, documents, and relevance judgments. Other than extending the ranker model to use the pre-trained *multilingual* cased version of BERT, we make no other changes to the hyperparameters and training strategy in the original implementation.[2] This

[2] https://github.com/Georgetown-IR-Lab/cedr

model is different from the one in (Jiang et al., 2020) which utilizes a data augmentation technique that breaks each query into word-level and each document into sentence-level and trains a Noisy-OR model.

6.3 Results

6.3.1 Main Results: Standard CLIR Setup

	Ja	**De**	**Fr**	**Tl**	**Sw**
Sasaki (In)	.73/.84	.71/.82	.76/.85	.51/.64	.53/.68
Sasaki (Sh)	-	-	-	.57/.69	.60/.73
BERT	**.94/.96**	**.96/.98**	**.97/.98**	**.84/.90**	**.88/.93**

Table 6.2: P@1/MAP performances on 5 languages. The BERT ranker models significantly outperform the baseline models, e.g. in Japanese achieving 94% P@1 (left) and 96% MAP (right).

Table 6.2 shows the P@1 (precision at top position) and MAP (mean average precision) of the CLIR datasets. The BERT ranker models significantly outperform the baseline CLIR models by large margins for all five languages. They achieve near-perfect performances for all three high-resource languages and outperform the best baseline models by 47.4% and 46.7% (in terms of P@1) for Tagalog and Swahili, respectively. The results are encouraging and demonstrate that the strategy of fine-tuning CLIR datasets on a pre-trained language model such as BERT is effective.

6.3.2 (Zero-Shot) Cross-Lingual Transfer

Train	Test				
	Ja	**De**	**Fr**	**Tl**	**Sw**
Ja	**.94/.96**	.96/.98	.96/.98	.80/.87	.88/.92
De	.88/.91	.96/.98	.97/.98	.82/.88	.85/.90
Fr	.88/.91	.96/.98	.97/.98	.78/.85	.85/.90
Tl	.89/.92	.97/.98	.97/.98	**.84/.90**	.88/.92
Sw	.90/.93	.96/.98	.97/.98	.80/.87	.88/.93

Table 6.3: P@/MAP of BERT ranker model in various zero-shot cross-lingual transfer settings. The diagonal repeats the results from Table 6.2. Results in bold are significantly better than the rest within the same columns.

Since multilingual BERT naturally supports more than 100 languages with a 110K shared wordpiece (Schuster and Nakajima, 2012) vocabulary, we can easily adapt the BERT ranker model to the zero-shot cross-lingual transfer setting, where we train the model on one language pair and evaluate the model on another language pair.

As seen in table 6.3, the zero-shot models perform well across all train-test combinations. For German and French, all models perform consistently at the P@1/MAP levels of around 96/98% and 97/98%, regardless of what data they are trained on. Although the non-zero-shot models outperform the zero-shot models significantly for Japanese and Tagalog, the zero-shot models still show remarkable results of around 80 to 96% for P@1 and MAP. The strong performances in zero-shot cross-lingual transfer settings show that multilingual

BERT is effective at learning representations across multiple languages (Wu and Dredze, 2019). *We recommend using multilingual BERT as a starting point for future work in zero-shot cross-lingual transfer CLIR tasks.*

6.4 Discussion and Analysis

6.4.1 How much training data is needed?

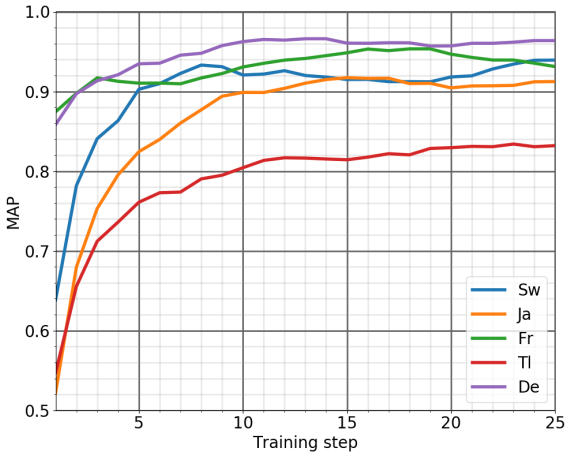


Figure 6.2: Learning curves of the BERT ranker models (Batch size = 16).

In figure 6.2, the BERT ranker models converge to their peak MAP scores on evaluation sets in just 5 to 20 training steps, with supervision from only 80 to 320 samples. This is consistent with the finding that BERT only requires a small amount of data for many downstream tasks (Devlin et al., 2019).

6.4.2 Do we actually need training data?

	Ja	De	Fr	Tl	Sw
BERT	.11/.35	.10/.36	.11/.35	.10/.27	.13/.31
LASER	.22/.32	.31/.46	.36/.49	.36/.47	.37/.50

Table 6.4: P@1/MAP performances of documents rank by the cosine similarities between queries and documents sentence embeddings.

The German and French models attain MAP of more than 85% in just one training step, while other models perform above 50% with minimal supervision. This introduces us to the possibility of approaching CLIR in an unsupervised manner. We experiment with two simple unsupervised approaches: (i) Rank documents based on the cosine similarities between the multilingual BERT embeddings of query and candidate documents. (ii) Alternatively, we can encode queries and documents with LASER, a multilingual sentence encoder trained on explicit cross-lingual signal (Artetxe and Schwenk, 2019; Schwenk et al., 2019).

The results of the unsupervised approaches are shown in table 6.4. As expected, ranking with cosine similarities between BERT embeddings performs poorly as BERT was not trained on any explicit cross-lingual signal. LASER works better but still performs significantly worse than the supervised models in table 6.2, possibly because LASER is not optimized for encoding long documents. *This shows that a small amount of training data is still necessary.*

6.4.3 Is the CLIR dataset too easy?

	Ja	De	Fr	Tl	Sw
Results	.10/.25	.05/.16	.13/.29	.25/.44	.17/.35
Samples	1.4M	2.6M	3.6M	151K	77K

Table 6.5: Results of BERT ranker models trained from scratch (1 epoch). Top shows the P@1/MAP performances on all languages. Bottom shows the number of training samples in 1 epoch.

The near-perfect performances of the BERT ranker models in Table 6.2 and 6.3 beg the question: *Is the CLIR dataset too easy?* To answer that, we retrain the BERT ranker models on the CLIR dataset from scratch without using any pre-trained model parameters. We train a randomly initialized BERT ranker model for every language pair for one epoch. As seen in Figure 6.5, these BERT ranker models perform significantly worse than the models in Figure 6.2. This is especially true for the Japanese, German and French models that perform below 30%, even when exposed to millions of training samples. The poor performances stand in stark contrast to Figure 6.2, where all models converge in less than 320 training samples. *This shows that optimizing from scratch on the CLIR dataset is inherently hard. Multilingual BERT provides an excellent initial point for further optimization (Hao et al., 2019) and is beneficial to this CLIR task.*

6.4.4 Is BERT modeling the interactions between queries and documents?

Ja	De	Fr	Tl	Sw
.05/.16	.04/.13	.03/.12	.03/.12	.03/.13

Table 6.6: P@1/MAP results of partial-input baselines.

Recently, there have been growing concerns about the partial-input baseline problem: When a partial-input model performs well on a dataset, that model might be "cheating" on the dataset and would not generalize well (Feng et al., 2019). To investigate whether the BERT ranker model suffers from the same problem, we create a partial-input dataset by masking the queries with single spaces. We then retrain BERT ranker models on the partial-input dataset. Fortunately, the partial-input baselines perform poorly as seen in Table 6.6. *This shows that the BERT ranker models do rely on the interactions between queries and documents and not just exploiting the linguistics cues in the documents.*

6.4.5 How much does BERT benefit from overlapping subword tokens across languages?

As Wikipedia documents in different languages could be referring to the same subject, having overlapping tokens in different documents is unavoidable.

Ja	De	Fr	Tl	Sw
.24/.35	.68/.75	.57/.66	.35/.46	.38/.52

Table 6.7: P@1/MAP of documents ranked by percentage of overlapping subword tokens.

For example, "Barack Obama" is written the same way in all five languages except Japanese. If "Barack Obama" appears in an English search query, it would also appear in relevant documents in other languages. We design a simple similarity function based on overlapping subword tokens' statistics to quantify how much BERT benefits from the overlapping subwords. Formally, given a query $q^X = \{\rho_1, \rho_2, \ldots, \rho_N\}$ in language X and a document $d^Y = \{\phi_1, \phi_2, \ldots, \phi_M\}$ in language Y, where ρ_i is the i-th subword token in q^X and ϕ_j is the j-th subword token in d^Y, the similarity function between q^X and d^Y is defined as:

$$f(q^Y, d^Y) = \frac{\sum_{i=1}^{N} g(\rho_i, d^Y)}{N}$$
$$g(\rho, d) = \begin{cases} 1 & \rho \in d \\ 0 & otherwise \end{cases} \tag{6.1}$$

As seen in Table 6.7, ranking the documents based on only overlapping subword tokens performs decently across all language pairs. It even manages to outperform the unsupervised LASER embeddings method in table 6.4 on almost all language pairs. The Japanese model performs worse than the other models since Japanese is the only language not written in the Latin alphabet.

Although Japanese is performing at only 24/35% using this simple similarity function, it improves drastically to 94/96% when we fine-tune the BERT ranker model on in-domain data. *This shows that although the statistics of overlapping subword tokens might be a helpful feature for some language pairs, BERT is much more sophisticated than simply comparing token overlaps between queries and documents.*

6.5 Conclusion

Results on the Large-Scale Wikipedia CLIR Dataset show that combining pre-trained language models such as the multilingual BERT with the existing neural retrieval model is effective in standard CLIR tasks and zero-shot cross-lingual transfer settings achieving good P@1/MAP with minimal supervision from training data. We also show that multilingual BERT does the heavy lifting for the CLIR task, which is challenging to optimize from scratch. We further eliminate the possibility that the BERT ranker model is performing well because of the partial-input baseline problem. This shows that the model is robust and does not overfit specific linguistic cues in the documents.

Chapter 7

CLIRMatrix: A Massively Large Collection of Bilingual and Multilingual Datasets for Cross-Lingual Information Retrieval

7.1 Introduction

Despite the growing interest in end-to-end CLIR, the lack of a large-scale, easily-accessible CLIR dataset covering many language directions in high-, mid-, and low-resource settings has detrimentally affected the CLIR community's capability to replicate and compare with previously published work. For example, among the widely-used datasets, the CLEF collection (Ferro and Silvello, 2015) covers many languages but is not large enough for training neural models. The more recent IARPA MATERIAL/OpenCLIR collection (Zavorin et al., 2020) is not yet publicly accessible. This motivates us to design and build *CLIRMatrix*, a massively large collection of bilingual and multilingual datasets for CLIR.

We construct *CLIRMatrix* from Wikipedia in an automated manner, exploiting its large variety of languages and massive number of documents. The core idea is to **synthesize relevance labels via an existing monolingual IR system, then propagate the labels via Wikidata links** that connect documents in different languages. In total, we were able to mine 49 million unique queries in 139 languages and 34 billion (query, document, label) triplets, creating a CLIR collection across a matrix of $139 \times 138 = 19,182$ language pairs. From this raw collection, we introduce two datasets:

- **BI-139** is a massively large bilingual CLIR dataset that covers $139 \times 138 =$

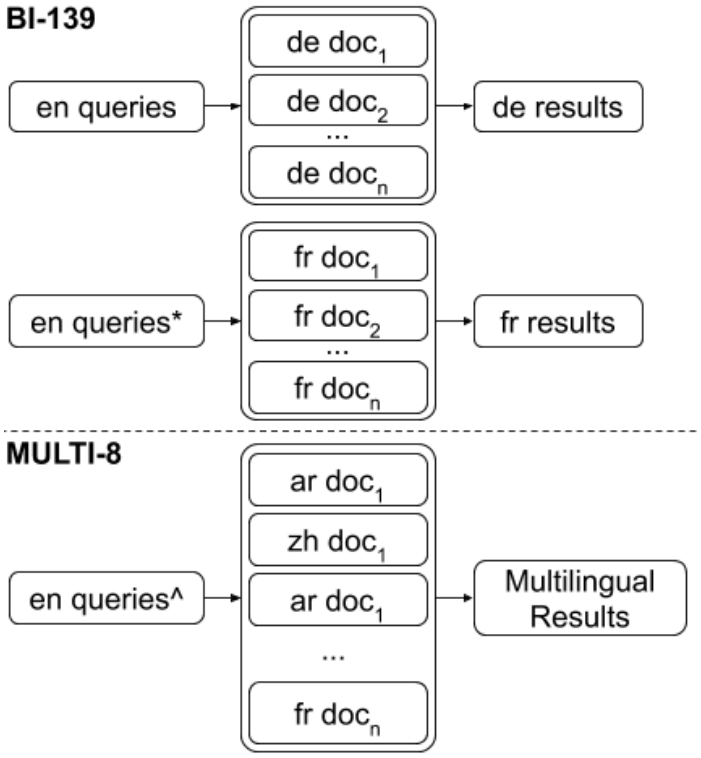


Figure 7.1: Illustration of our CLIRMatrix collection. The BI-139 portion of CLIRMatrix supports research in bilingual retrieval and covers a matrix of 139×138 language pairs. The MULTI-8 portion of CLIRMatrix supports research in multilingual modeling and mixed-language (ML) retrieval, where queries and documents are jointly aligned over 8 languages.

$19,182$ language pairs. To encourage reproducibility, we present standard train, validation, and test subsets for every language direction.

- **MULTI-8** is a *multilingual* CLIR dataset comprising of queries and documents jointly aligned 8 languages: Arabic (ar), German (de), English (en), Spanish (es), French (fr), Japanese (ja), Russian (ru), Chinese (zh). Each query will have relevant documents in the other seven languages.

See Figure 7.1 for a comparison of BI-139 and MULTI-8. The former facili-

tates the evaluation of bilingual retrieval over a wide variety of languages. At the same time, the latter supports research in mixed-language retrieval (a.k.a multilingual retrieval (Savoy and Braschler, 2019)), which is an interesting yet relatively under-explored problem. For both, the train sets are large enough to enable the training of the neural IR models.

We hope *CLIRMatrix* is useful and can empower further developments in this field of research. To summarize, the contributions of this chapter are:

1. A massive CLIR collection supporting both training and evaluation of bilingual/multilingual models.

2. A set of baseline neural results on BI-139 and MULTI-8. On MULTI-8, we show that a single multilingual model can significantly outperform an ensemble of bilingual models.

CLIRMatrix is publicly available at `https://github.com/ssun32/CLIRMatrix`.

7.2 Methodology

Let q^X be a query in language X, and d^Y be a document in language Y. A bilingual CLIR dataset consists of I triples

$$\{(q_i^X, d_{ij}^Y, r_{ij})\}_{i=1,2,...,I} \tag{7.1}$$

where d_{ij}^Y is the j-th document associated with query q_i^X, and r_{ij} is a label saying how relevant is the document d_{ij}^Y to the query q_i^X. Conventionally, r_{ij} is an integer with 0 representing "not relevant" and higher values indicating more relevant.

Suppose there are J documents in total. In the full collection search setup, the index j ranges from $1, \ldots, J$, meaning that each query q_i^X searches over the full set of documents $\{d_{ij}^Y\}_{j=1,\ldots,J}$. In the re-ranking setup, each query q_i^X searches over a subset of documents obtained by an initial full-collection retrieval engine: $\{d_{ij}^Y\}_{j=1,\ldots,K_i}$, where $K_i \ll J$. For practical reasons, machine learning approaches to IR focus on the re-ranking setup with K_i set to 10$\sim$1000 (Liu, 2009; Chapelle and Chang, 2011). We follow the re-ranking setup here.

We now describe the main intuition of our construction method and detail various components and design choices in our pipeline.

7.2.1 Intuition and Assumptions

To create a CLIR dataset, one needs to decide how to obtain q_i^X and d_{ij}^Y, and r_{ij}. We set q_i^X to be Wikipedia titles, d_{ij}^Y to be Wikipedia articles, and synthesize r_{ij} automatically using a simple yet reliable method. We argue that *Wikipedia* is the best available resource for building CLIR datasets due to two reasons: First, it is freely available and contains articles in more than 300 languages, covering various topics. Second, Wikipedia articles are mapped to entities in

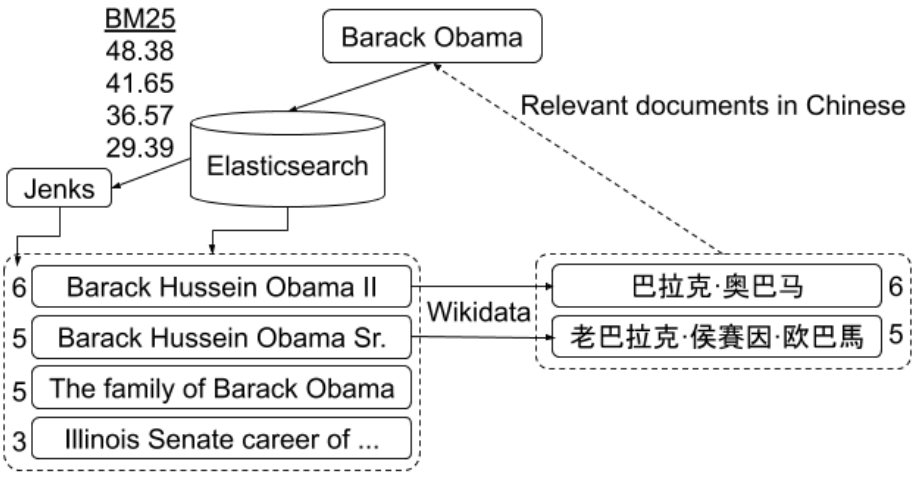


Figure 7.2: Intuition of CLIR relevance label synthesis. For the English query "Barack Obama", first a monolingual IR engine (Elasticsearch) labels documents in English; then Wikidata links are exploited to propagate the label to the corresponding Chinese documents, which are assumed to be topically similar.

Wikidata[1], which is a relatively reliable way to find the same articles written in other languages.

To synthesize relevance labels r_{ij}, we propose first to generate labels using an existing monolingual IR system in language X, then propagate the labels via Wikidata links to language Y. In other words, we assume:

1. the availability of documents d^X in the *same* language as the query, and
2. the feasibility of an existing monolingual IR system in language X to provide labels $\hat{r}_{ij}$ on (q_i^X, d_{ij}^X) pairs

Then for any d_{ij}^Y that links to d_{ij}^X, we assign the relevance label $\hat{r}_{ij}$.

[1]Wikidata is a knowledge base that contains links to parallel Wikipedia documents in different languages.

This intuition is illustrated in Figure 7.2. Suppose we wish to find Chinese documents relevant to the English query "Barack Obama". We first run monolingual IR to find English documents that answer the query. In this figure, four documents are returned, and we attempt to link to the corresponding Chinese versions using Wikidata information. When the link is available, we set the relevance label r_{ij} for Chinese documents using the English-based IR system's predictions $\hat{r}_{ij}$; all other documents are deemed not relevant. This gives us the triplet $(q_i^X, d_{ij}^Y, r_{ij})$.

7.2.2 Mining Pipeline

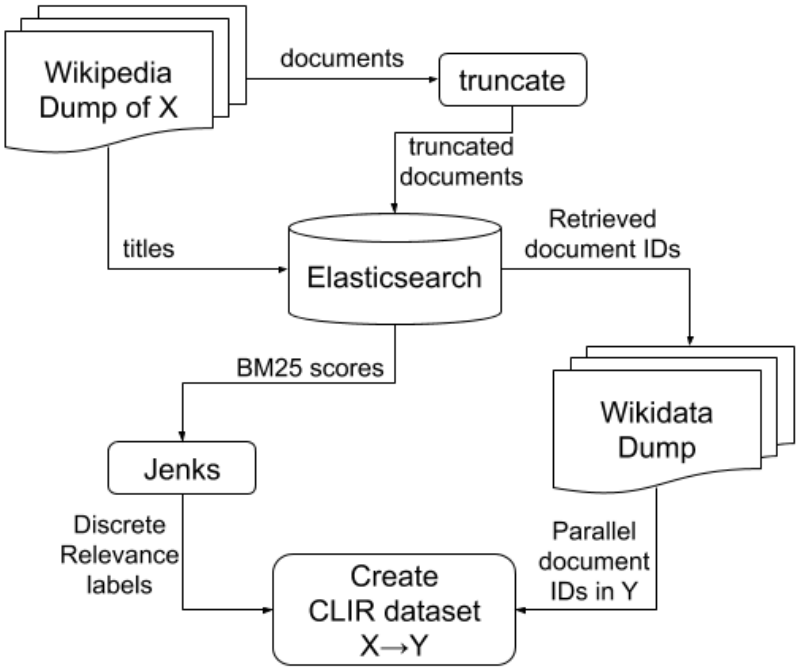


Figure 7.3: Mining pipeline for constructing a bilingual CLIR dataset with queries in language X and documents in language Y.

Figure 7.3 is our mining pipeline that implements the intuition in Figure 7.2. First, we download the Wikipedia dump of language X and extract every article's titles and document bodies. We index the documents into an Elasticsearch[2] search engine, which serves as our monolingual IR system. Using the extracted titles as search queries, we retrieve the top 100 relevant documents and their corresponding BM25 scores from Elasticsearch for every query. We then convert the BM25 scores into discrete relevance judgment labels using Jenks natural break optimization. Finally, we propagate these labels to documents in language Y linked via Wikidata.

We downloaded Wikidata, and Wikipedia dumps released on January 1, 2020. Since Wikipedia dumps contain tremendous amounts of meta-information such as URLs and scripts, it can be expensive to extract actual text directly from those dumps. Inspired by (Schwenk et al., 2019), we extracted document ids, titles, and bodies from Wikipedia's search indices[3] instead, which contain raw text data without meta-information.

WIKIPEDIA DUMPS

We discarded dumps with less than ten thousand documents, usually the dumps of Wikipedia of particular dialects and less commonly used languages. We are left with Wikipedia dumps in 139 languages, containing a good mix of

[2]https://www.elastic.co/
[3]https://dumps.wikimedia.org/other/cirrussearch/

high-, mid-, and low-resource languages. For writing systems that do not use whitespaces, such as Chinese, Japanese, and Thai, we truncated documents to approximately the first 600 characters. We kept roughly the first 200 tokens of every document for other languages. Truncating the documents is necessary for several reasons: First, shorter documents are more friendly to neural models bounded by GPU memories. Second, the first few hundred tokens of Wikipedia articles are usually the main points of the complete text, thus are more likely to be topically similar across languages. Last but not least, BM25 tends to over-penalize long documents, which can lead to sub-optimal IR performances (Lv and Zhai, 2011). We hypothesize we can get better relevant judgment labels if we use shorter documents.

WIKIDATA DUMP

We downloaded the JSON dump[4] of Wikidata, a structured knowledge base that links to Wikipedia. We designed a regex rule that efficiently obtains a list of entities IDs from the Wikidata dump. We also extracted a list of related (language code, document title) pairs for every entity ID. Using our extracted Wikipedia data, we matched the document titles to Wikipedia document IDs[5]. The extracted data allows us to construct two dictionaries: 1) A dictionary maps the document ID in some language to its Wikidata entity ID.

[4]https://dumps.wikimedia.org/wikidatawiki/entities/

[5]Note that documents in different languages do not share document IDs. This means that document N in language X does not refer to the same entity as document N in language Y.

2) A reverse dictionary maps a Wikidata entity ID to document IDs in different languages. This enables us to locate a document's counterpart in another language quickly; we use this information to find link relevant documents across languages.[6]

7.2.3 Design Choices

DOCUMENT TITLES AS SEARCH QUERIES

We considered several methods used to generate search queries. One quick way is to acquire human-generated search queries directly from search logs. However, this is not a viable option because search logs are not publicly available for most languages. Alternatively, we can engage human annotators to generate search queries manually, but this can be time-consuming, expensive, and impossible to quickly scale the process to 139 languages.

We use document titles as search queries for two reasons: (1) They are readily available in large amounts for each of the 139 languages, enabling us to build large datasets (i.e., I is large). (2) In specific real-world search settings, queries are typically short, spanning only two to three tokens (Belkin et al.,

[6]We acknowledge that there are potentially missing inter-language links in Wikidata. This implies that our method may miss the labeling of some relevant documents. Wikidata has several policies to improve its data quality, such as requests for editors to link new Wikipedia articles to entities in Wikidata. There are also automated auditing tools that periodically identify articles with missing or inconsistent Wikidata labels and ask human editors for verification. An interesting research problem for future work is finding ways to quantify these inter-language links' coverage.

2003) and informational, covering a wide variety of topics (Jansen et al., 2008). We leave the investigation of complex queries to future work. We want to emphasize that our mining pipeline is compatible with all query types; for example, we can use the first sentences of documents as queries (Schamoni et al., 2014b; Sasaki et al., 2018) if desired.

BM25 AND ELASTICSEARCH

The main step of our mining pipeline is to index documents into a monolingual IR system and then retrieve a list of relevant documents and similarity scores for every query. We assume the similarity score between a query and a document accurately reflects the degree of relevance for that document. Since many Wikipedia dumps contain millions of documents, the computations needed to retrieve relevant documents for all 139 languages are non-trivial. We need an efficient retrieval system that can handle the retrieval task efficiently and accurately. For this reason, we chose Elasticsearch[7] as our monolingual IR system.

Elasticsearch is an open-source, highly optimized search engine software based on Apache Lucene[8]. It has built-in analyzers that handle language-specific preprocessing such as tokenization and stemming. By default, Elasticsearch implements the BM25 weighting scheme (Robertson et al., 2009), a

[7]Elasticsearch is also used as the backend search engine for Wikipedia.org
[8]https://lucene.apache.org/core/

bag-of-word retrieval function that calculates similarity scores between queries and documents based on term frequencies and inverse document frequencies. BM25 is a strong baseline that frequently outperforms existing neural IR models on multiple benchmark IR datasets (Chapelle and Chang, 2011; Guo et al., 2016b; McDonald et al., 2018).

We used Elasticsearch 6.5.4 and imported the same settings as the official search indices from Wikipedia[9]. We configured Elasticsearch to search document titles and document bodies for every query, with twice the weight given to document titles. We limit Elasticsearch to return only the top 100 documents for each query and assume documents not returned by the search engine are irrelevant. We parallelized the retrieval processes by running multiple Elasticsearch instances on numerous servers and dedicated one Elasticsearch instance to every language.

DISCRETE RELEVANCE JUDGMENT LABELS

A potential pitfall of using document titles as queries is that some short queries can be ambiguous (Allan and Raghavan, 2002). For example, it is impossible to determine whether the search query "Java" refers to the Java programming language or the island in Indonesia without other context words. Fortunately, Wikipedia disambiguates different document titles by append-

[9]For example, the settings for English Wikipedia is available at https://en.wikipedia.org/w/api.php?action=cirrus-settings-dump&format=json&formatversion=2. For BM25, $b = 0.3$ and $k_1 = 1.2$.

ing category information to the titles, e.g., Java (Programming Language) and Java (Island), etc. Nevertheless, we do not want to rank retrieved documents solely based on their BM25 scores. To prevent potential ambiguity issues, we smooth out the BM25 scores into discrete relevance judgment labels. We achieve this by using the Jenks natural break optimization (McMaster and McMaster, 2002), an algorithm that finds optimal BM25 score intervals for different labels by iteratively reducing the variance within labels and maximizing the variance between labels.

More specifically, for each query q_i^X, we normalized the BM25 scores $\hat{r}_{ij}$ of d_{ij}^X to the unit range. We then used Jenks optimization to distribute the normalized scores into five different relevance judgment labels $\{1, 2, 3, 4, 5\}$. We want to emphasize that we did not run Jenks optimization globally across all BM25 scores because the scales of BM25 scores are not consistent across different queries. Additionally, documents that are not returned by Elasticsearch or not linked by any Wikidata are deemed irrelevant and given a label 0. We also assigned label 6 to the document associated with the title query. So final r_{ij} is of **a scale of 0 to 6**, with 0 being irrelevant and 6 being most relevant.

7.2.4 Bilingual and Multilingual datasets

7.2.4.0.1 BI-139

Using the pipeline, we build a bilingual dataset $\{(q_i^X, d_{ij}^Y, r_{ij})\}_{i=1,2,...,I}$ for every X→Y language direction. In the "raw" version, there are 49.28 million unique queries and 34.06 billion (query, document, label) triplets across $139 \times 138 = 19,182$ language directions. We also generated a "base" version containing standard train, validation, test1, and test2 subsets for each language direction. Train sets contain up to I=10,000 queries, while validation, test1, and test2 sets contain 1,000 queries. We ensured that queries in the train and validation/test sets of one language direction do not overlap with those in the test sets from other language directions. We ensure there are precisely K =100 candidate documents for every query by filling the shortfall with random irrelevant documents.

MULTI-8

This is a multilingual CLIR dataset covering 8 languages from various world regions (Arabic, German, English, Spanish, French, Japanese, Russian, and Chinese). First, we restricted queries to those with a relevant document ($r_{ij} = 6$) in all 8 languages. Then, for each query q_i^X, we use the monolingual IR

systems to collect 100 documents in the same language d_{ij}^{X}.[10] Similar to BI-139 base, if ElasticSearch returns less than 100 documents labels ($r_{ij} \geq 1$), then we fill-up the short-fall with random irrelevant documents with label $r_{ij} = 0$. Finally, we merge these document lists such that for any query in language X, we have 7×100 documents in the other seven languages.

Similar to the base version of BI-139, the train sets contain 10,000 queries, while validation, test1, and test2 sets contain 1,000 queries, but note that the query sets are different. This dataset supports two kinds of research: First, one can still evaluate bilingual CLIR (single-language retrieval) like BI-139 but exploit training multilingual models using more than two languages. Second, one can evaluate multilingual CLIR (mixed-language retrieval), where the document list to be re-ranked contains two or more languages. This research direction is relatively unexplored, except for early work in the 2000s in the CLEF campaign (Savoy and Braschler, 2019).

7.2.5 File Formats

{*“src_id”: ”6267”,*
“src_query”: “Cultural imperialism”,
“tgt_results”: [[“3383724”, 6], [“19028”, 5], [“6291141”, 4], [“4394682”, 2], [“138124”, 1], [”1245746”, 1], [“1004260”, 0], ...}

Figure 7.4: An example English query “Cultural imperialism” and the document IDs and labels of its relevant Chinese documents.

[10] Recall that our Wikidata entities dictionary can map a language-independent entity to query strings (Wikipedia article titles) in any language.

6499809 ⟨TAB⟩ Structured light is the process of projecting a known pattern (often grids or horizontal bars) on to a scene...

Figure 7.5: The IDs and texts of documents are stored tab-separated in a text file.

For every language direction, we store queries and their relevant document IDs and labels in the JSON Lines format (Figure 7.4). For each unique language, we store the IDs and texts of documents in TSV files (Figure 7.5). Note that we will release both the truncated and the original documents.

7.2.6 Average Number of Relevant Documents per Query

Table 7.1 , 7.2 and 7.3 present the average number of relevant documents per query for English, Chinese and Swahili queries respectively. Each table cell presents two numbers. On the left is the average number of documents with relevance labels of at least four per query. The number on the right is the average number of documents with relevance labels of at least one per query. Table 7.4 presents the average number of relevant documents per query for all language pairs from CLIRMatrix MULTI-8. Table 7.5 presents a side-by-side comparison of the statistics of Large-scale CLIR dataset and CLIRMatrix.

Table 7.1: CLIRMatrix BI-139: Average number of documents (relevance label $\geq$ 4/relevance label $\geq$ 1) per query for **English (en)** queries

af	**als**	**am**	**an**	**ar**	**arz**	**ast**	**az**	**azb**	**ba**
0.5/2.3	0.4/1.5	0.3/1.2	0.4/1.5	3.0/14.7	0.3/1.5	0.6/2.4	0.7/2.8	0.8/3.3	0.4/1.6
bar	**be**	**bg**	**bn**	**bpy**	**br**	**bs**	**bug**	**ca**	**cdo**
0.4/1.7	0.6/2.6	0.5/2.1	0.4/2.0	0.5/2.5	0.5/2.1	0.5/2.3	0.8/3.7	1.5/6.5	0.3/1.6
ce	**ceb**	**ckb**	**cs**	**cv**	**cy**	**da**	**de**	**diq**	**el**
0.8/3.3	2.3/9.2	0.5/1.6	1.1/5.2	0.4/1.5	0.6/2.4	0.8/3.6	3.8/18.5	0.3/1.3	0.7/3.1
eml	**eo**	**es**	**et**	**eu**	**fa**	**fi**	**fo**	**fr**	**fy**
0.3/1.2	0.8/3.6	3.1/15.1	0.6/2.8	1.1/4.7	2.0/9.2	1.3/5.7	0.3/1.4	4.1/19.7	0.4/1.6
ga	**gd**	**gl**	**gu**	**he**	**hi**	**hr**	**hsb**	**ht**	**hu**
0.4/1.9	0.3/1.3	0.6/2.6	0.3/1.4	0.9/3.8	0.5/2.5	0.7/3.1	0.4/1.4	0.4/1.8	1.2/5.2
hy	**ia**	**id**	**ilo**	**io**	**is**	**it**	**ja**	**jv**	**ka**
0.6/2.7	0.4/1.5	1.2/5.1	0.4/1.9	0.4/1.5	0.4/1.6	3.2/15.7	1.9/9.0	0.4/1.5	0.6/2.6
kk	**kn**	**ko**	**ku**	**ky**	**la**	**lb**	**li**	**lmo**	**lt**
0.6/2.7	0.3/1.4	1.2/5.4	0.3/1.3	0.4/1.8	0.6/2.6	0.4/1.9	0.3/1.3	0.4/1.9	0.6/2.8
lv	**mai**	**mg**	**mhr**	**min**	**mk**	**ml**	**mn**	**mr**	**mrj**
0.5/2.1	0.3/1.4	0.5/2.3	0.3/1.2	1.6/3.7	0.6/2.4	0.5/2.1	0.3/1.3	0.4/1.8	0.4/1.5
ms	**my**	**mzn**	**nap**	**nds**	**ne**	**new**	**nl**	**nn**	**no**
1.0/4.3	0.4/1.7	0.6/1.8	0.4/1.6	0.4/1.8	0.4/1.6	0.4/1.7	3.1/12.1	0.7/3.0	1.4/6.2
oc	**or**	**os**	**pa**	**pl**	**pms**	**pnb**	**ps**	**pt**	**qu**
0.6/2.3	0.4/1.4	0.3/1.2	0.4/1.6	2.7/12.5	0.4/1.4	0.5/2.0	0.3/1.3	2.3/10.9	0.3/1.4
ro	**ru**	**sa**	**sah**	**scn**	**sco**	**sd**	**sh**	**si**	**simple**
0.9/4.2	2.8/13.0	0.3/1.2	0.3/1.2	0.4/1.5	0.5/1.9	0.3/1.3	0.9/4.0	0.3/1.4	0.7/3.0
sk	**sl**	**sq**	**sr**	**su**	**sv**	**sw**	**szl**	**ta**	**te**
0.8/3.3	0.7/2.9	0.5/2.1	1.0/4.5	0.3/1.4	2.8/11.7	0.4/1.8	0.4/1.4	0.6/2.7	0.4/1.8
tg	**th**	**tl**	**tr**	**tt**	**uk**	**ur**	**uz**	**vec**	**vi**
0.6/2.8	0.6/2.6	0.5/2.3	1.0/4.4	0.4/1.7	1.8/8.2	0.8/3.6	0.6/2.5	0.3/1.3	2.1/7.5
vo	**wa**	**war**	**wuu**	**xmf**	**yi**	**yo**	**zh**		
0.8/3.5	0.3/1.2	2.2/6.9	0.4/1.6	0.3/1.3	0.3/1.3	0.4/1.5	1.8/9.0		

Table 7.2: CLIRMatrix BI-139: Average number of documents (relevance label $\geq$ 4/relevance label $\geq$ 1) per query for **Chinese (zh)** queries

af 0.6/4.3	**als** 0.4/3.0	**am** 0.3/1.8	**an** 0.4/3.8	**ar** 2.3/19.5	**arz** 0.3/4.0	**ast** 0.7/5.6	**az** 0.7/6.0	**azb** 0.9/8.2	**ba** 0.3/2.3
bar 0.4/3.1	**be** 1.0/8.3	**bg** 0.5/4.5	**bn** 0.4/3.2	**bpy** 0.6/5.5	**br** 0.6/4.5	**bs** 0.5/4.1	**bug** 0.7/7.2	**ca** 2.4/20.6	**cdo** 0.4/2.9
ce 0.9/9.2	**ceb** 3.1/26.3	**ckb** 0.4/2.5	**cs** 1.6/13.6	**cv** 0.4/2.4	**cy** 0.5/4.2	**da** 1.2/9.7	**de** 3.8/33.8	**diq** 0.3/2.0	**el** 0.9/7.4
eml 0.3/1.7	**en** 6.3/53.2	**eo** 1.4/14.2	**es** 3.9/34.9	**et** 0.9/8.1	**eu** 2.2/20.0	**fa** 2.5/22.6	**fi** 1.7/14.3	**fo** 0.3/2.1	**fr** 4.0/36.4
fy 0.4/2.8	**ga** 0.5/3.8	**gd** 0.3/2.0	**gl** 0.9/7.2	**gu** 0.2/1.8	**he** 1.2/10.0	**hi** 0.5/4.0	**hr** 0.9/7.5	**hsb** 0.3/2.2	**ht** 0.4/3.6
hu 2.1/19.7	**hy** 1.1/11.4	**ia** 0.4/3.3	**id** 1.7/13.9	**ilo** 0.4/1.9	**io** 0.4/3.4	**is** 0.5/3.5	**it** 3.9/34.7	**ja** 3.5/28.1	**jv** 0.4/2.7
ka 0.8/6.6	**kk** 1.1/10.1	**kn** 0.3/2.1	**ko** 2.4/18.9	**ku** 0.3/2.0	**ky** 0.5/4.0	**la** 1.1/11.3	**lb** 0.5/3.9	**li** 0.3/2.0	**lmo** 0.6/5.1
lt 0.9/7.6	**lv** 0.7/5.5	**mai** 0.2/1.6	**mg** 0.8/8.1	**mhr** 0.3/1.8	**min** 0.3/2.0	**mk** 0.8/5.9	**ml** 0.5/3.7	**mn** 0.3/2.4	**mr** 0.4/3.0
mrj 0.3/1.9	**ms** 1.6/15.1	**my** 0.3/2.3	**mzn** 0.3/1.8	**nap** 0.4/3.9	**nds** 0.4/3.3	**ne** 0.3/2.2	**new** 0.4/3.8	**nl** 3.6/31.1	**nn** 0.9/7.5
no 1.7/14.4	**oc** 1.0/9.5	**or** 0.3/2.2	**os** 0.3/1.7	**pa** 0.3/2.2	**pl** 3.3/30.8	**pms** 0.1/1.1	**pnb** 0.4/3.5	**ps** 0.2/1.7	**pt** 3.2/28.7
qu 0.3/2.6	**ro** 2.0/19.9	**ru** 3.7/32.6	**sa** 0.3/1.8	**sah** 0.3/1.8	**scn** 0.4/3.4	**sco** 0.6/4.8	**sd** 0.2/1.6	**sh** 1.4/13.1	**si** 0.3/2.1
simple 1.1/9.8	**sk** 1.3/12.0	**sl** 0.8/6.2	**sq** 0.6/5.1	**sr** 1.6/15.3	**su** 0.3/2.2	**sv** 3.7/31.2	**sw** 0.4/3.1	**szl** 0.3/1.9	**ta** 0.6/4.7
te 0.3/2.2	**tg** 0.4/2.8	**th** 0.9/6.2	**tl** 0.6/4.9	**tr** 1.5/12.6	**tt** 0.4/3.1	**uk** 2.9/27.3	**ur** 0.8/6.7	**uz** 1.1/10.9	**vec** 0.3/2.3
vi 2.9/25.0	**vo** 1.7/18.1	**wa** 0.2/1.6	**war** 2.4/20.1	**wuu** 0.4/3.0	**xmf** 0.3/2.2	**yi** 0.3/2.0	**yo** 0.4/5.0		

Table 7.3: CLIRMatrix BI-139: Average number of documents (relevance label $\geq$ 4/relevance label $\geq$ 1) per query for **Swahili (sw)** queries

af	**als**	**am**	**an**	**ar**	**arz**	**ast**	**az**	**azb**	**ba**
2.8/14.1	1.5/7.7	1.4/7.2	1.5/8.2	5.3/25.6	1.5/6.8	2.6/14.3	3.0/14.5	2.9/13.0	1.5/7.1
bar	**be**	**bg**	**bn**	**bpy**	**br**	**bs**	**bug**	**ca**	**cdo**
1.3/6.6	3.4/16.8	1.5/7.6	1.7/8.9	1.0/6.6	2.7/13.2	2.3/11.2	0.3/1.6	4.6/22.8	1.8/9.3
ce	**ceb**	**ckb**	**cs**	**cv**	**cy**	**da**	**de**	**diq**	**el**
1.3/6.5	9.1/36.8	1.8/7.9	4.7/23.0	1.7/7.1	2.7/13.8	3.6/18.2	5.9/27.6	1.5/6.4	3.8/19.1
eml	**en**	**eo**	**es**	**et**	**eu**	**fa**	**fi**	**fo**	**fr**
1.7/9.5	7.2/32.9	3.9/19.1	5.6/27.2	3.6/17.9	4.3/21.3	5.2/24.9	4.4/21.9	1.6/9.8	6.1/28.2
fy	**ga**	**gd**	**gl**	**gu**	**he**	**hi**	**hr**	**hsb**	**ht**
1.9/9.9	2.7/12.8	1.7/8.7	3.5/18.4	1.0/4.9	3.4/18.2	2.3/14.2	3.6/18.3	1.1/4.9	2.3/11.4
hu	**hy**	**ia**	**id**	**ilo**	**io**	**is**	**it**	**ja**	**jv**
4.5/22.8	2.6/12.9	1.4/6.4	4.5/20.7	1.7/7.7	2.3/11.3	2.1/11.1	6.5/30.3	5.2/25.4	1.9/9.2
ka	**kk**	**kn**	**ko**	**ku**	**ky**	**la**	**lb**	**li**	**lmo**
3.2/16.4	2.5/12.5	1.3/6.1	4.8/23.6	1.6/7.0	1.7/7.8	3.3/15.9	1.9/9.8	1.2/5.8	1.4/7.0
lt	**lv**	**mai**	**mg**	**mhr**	**min**	**mk**	**ml**	**mn**	**mr**
3.6/20.8	3.1/14.8	1.1/6.0	1.8/8.4	1.1/6.3	0.7/3.3	3.4/18.6	2.3/10.8	1.6/6.8	2.5/11.4
mrj	**ms**	**my**	**mzn**	**nap**	**nds**	**ne**	**new**	**nl**	**nn**
1.3/6.5	4.2/19.9	1.1/6.4	1.1/5.0	0.6/2.2	1.6/8.0	1.2/6.1	1.5/7.1	5.3/26.0	2.8/13.6
no	**oc**	**or**	**os**	**pa**	**pl**	**pms**	**pnb**	**ps**	**pt**
4.6/22.7	2.6/15.1	1.0/5.2	1.3/4.9	1.4/6.8	5.8/27.3	0.4/1.7	2.5/9.7	1.1/4.9	5.3/26.0
qu	**ro**	**ru**	**sa**	**sah**	**scn**	**sco**	**sd**	**sh**	**si**
1.8/9.2	4.3/20.4	5.6/26.5	1.1/5.5	1.2/6.3	1.7/9.2	3.0/13.7	0.9/3.8	4.1/20.3	1.1/5.2
simple	**sk**	**sl**	**sq**	**sr**	**su**	**sv**	**szl**	**ta**	**te**
3.6/17.8	3.3/16.3	3.0/15.5	2.3/11.9	4.5/23.9	1.6/8.6	5.5/26.3	1.1/4.2	2.4/12.4	1.2/5.8
tg	**th**	**tl**	**tr**	**tt**	**uk**	**ur**	**uz**	**vec**	**vi**
2.1/10.3	3.0/14.6	3.0/13.6	4.4/20.7	2.7/14.1	5.0/24.7	3.8/17.2	3.2/16.6	1.5/7.6	4.5/22.2
vo	**wa**	**war**	**wuu**	**xmf**	**yi**	**yo**	**zh**		
2.5/12.0	0.9/5.2	3.5/18.4	1.8/9.5	1.4/7.7	1.4/6.6	2.1/10.8	5.5/25.6		

Query Language	**Document Language**							
	ar	de	en	es	fr	ja	ru	zh
ar		10.2	14.8	10.1	10.4	8.7	10.0	8.8
de	4.2		8.5	5.6	6.5	4.7	5.9	4.5
en	4.6	6.6		6.2	7.0	4.8	6.0	4.9
es	5.4	7.6	11.0		8.3	5.8	7.2	5.9
fr	5.1	7.8	10.9	7.3		5.6	7.0	5.5
ja	8.6	9.8	11.6	8.9	9.2		9.9	9.2
ru	7.3	9.9	12.7	8.9	9.7	7.9		7.9
zh	5.5	7.0	9.2	6.9	7.1	6.6	6.9	

Table 7.4: CLIRMatrix MULTI-8: Average number of documents with relevance label ≥ 4 per query

7.3 Experiment Setup

7.3.1 Baseline Neural CLIR Model

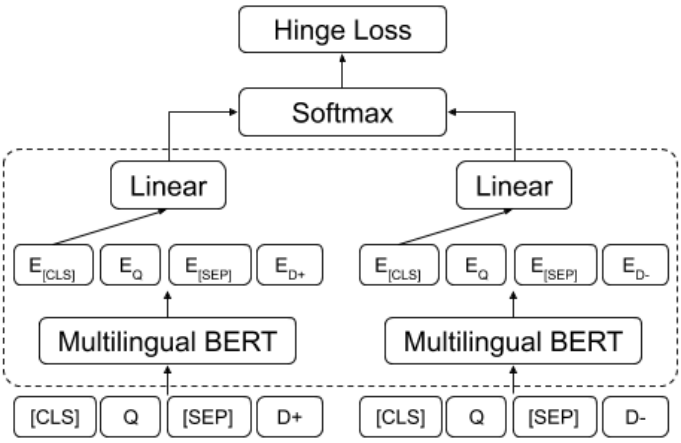


Figure 7.6: Neural architecture of our baseline CLIR model. Modules in the dotted rectangle share weights.

We follow the implementation of the vanilla BERT ranker model (MacA-

Query Lang	Document Lang	Large-Scale CLIR	CLIRMatrix
en	ar	1.60	14.67
	ca	2.84	6.54
	cs	4.09	5.19
	de	5.92	18.54
	es	3.70	15.10
	fi	3.43	5.74
	fr	4.72	19.65
	it	4.26	15.69
	ja	7.83	8.97
	ko	2.53	5.44
	nl	3.39	12.10
	nn	2.51	3.02
	no	3.21	6.17
	pl	3.56	12.49
	pt	2.85	10.94
	ro	2.26	4.23
	ru	3.49	13.02
	simple	2.19	3.02
	sv	3.24	11.66
	sw	2.53	1.79
	tl	1.48	2.31
	tr	2.05	4.44
	uk	2.62	8.25
	vi	1.73	7.51
	zh	2.00	9.04

Table 7.5: Average number of relevant documents per query for the large-scale CLIR dataset and CLIRMatrix

vaney et al., 2019), which obtained substantial results in monolingual IR. As shown in Figure 7.6, the model encodes a query-document pair with BERT (Devlin et al., 2019) and stacks a linear combination layer on top of the [CLS] token. We extended the ranker model to use multilingual BERT[11]. We sample documents pairs during training time in which the positive documents have higher relevance judgment labels than the negative ones. We obtain scores for both documents using the same BERT ranker model for each document pair. We then optimize the parameters with pairwise hinge loss and Adam optimizer. We trained all models for 20 epochs and sampled around 1,000 training pairs for each epoch. At inference time, we rerank documents based on the output scores from the BERT ranker model.

7.3.2 Evaluation Metric

We report all results in NDCG@10 (normalized discounted cumulative gain) as discussed in section 2.5.2.

7.3.3 Results on BI-139

We present results on the 138 target languages for English queries. We trained a baseline CLIR model on the base train set for each language direction and kept the checkpoint with the best NDCG@10 performance on the base

[11] We used BERT-Base, Multilingual Cased

af	**als**	**am**	**an**	**ar**	**arz**	**ast**	**az**	**azb**	**ba**
.90	.88	.56	.90	.80	.86	.88	.80	.87	.87
bar	**be**	**bg**	**bn**	**bpy**	**br**	**bs**	**bug**	**ca**	**cdo**
.89	.83	.85	.78	.85	.84	.89	.91	.88	.85
ce	**ceb**	**ckb**	**cs**	**cv**	**cy**	**da**	**de**	**diq**	**el**
.90	.89	.72	.89	.84	.87	.90	.88	.81	.83
eml	**eo**	**es**	**et**	**eu**	**fa**	**fi**	**fo**	**fr**	**fy**
.80	.87	.87	.83	.86	.85	.86	.87	.84	.90
ga	**gd**	**gl**	**gu**	**he**	**hi**	**hr**	**hsb**	**ht**	**hu**
.78	.79	.87	.78	.82	.79	.88	.86	.88	.86
hy	**ia**	**id**	**ilo**	**io**	**is**	**it**	**ja**	**jv**	**ka**
.82	.90	.00	.88	.86	.83	.84	.84	.89	.81
kk	**kn**	**ko**	**ku**	**ky**	**la**	**lb**	**li**	**lmo**	**lt**
.85	.67	.86	.76	.82	.88	.88	.85	.83	.86
lv	**mai**	**mg**	**mhr**	**min**	**mk**	**ml**	**mn**	**mr**	**mrj**
.85	.80	.88	.84	.92	.86	.87	.86	.74	.82
ms	**my**	**mzn**	**nap**	**nds**	**ne**	**new**	**nl**	**nn**	**no**
.89	.77	.85	.85	.88	.73	.75	.89	.90	.89
oc	**or**	**os**	**pa**	**pl**	**pms**	**pnb**	**ps**	**pt**	**qu**
.91	.71	.83	.76	.86	.78	.70	.72	.86	.81
ro	**ru**	**sa**	**sah**	**scn**	**sco**	**sd**	**sh**	**si**	**simple**
.89	.85	.73	.77	.81	.94	.78	.87	.48	.93
sk	**sl**	**sq**	**sr**	**su**	**sv**	**sw**	**szl**	**ta**	**te**
.86	.89	.88	.88	.91	.88	.87	.92	.85	.81
tg	**th**	**tl**	**tr**	**tt**	**uk**	**ur**	**uz**	**vec**	**vi**
.85	.81	.89	.87	.87	.85	.85	.84	0.88	0.89
vo	**wa**	**war**	**wuu**	**xmf**	**yi**	**yo**	**zh**		
0.89	0.75	0.86	0.83	0.79	0.65	0.89	0.84		

Table 7.6: Results of 138 language directions from BI-139 base with English queries. The top shows a candidate's language code in each cell, and the bottom shows the NDCG@10 score for that language direction.

validation set. We reranked the documents in the base test1 set and calculated NDCG@10. Table 7.6 lists the the baseline results.[12] The pleasant surprise is that the baseline CLIR models did pretty well on languages that multilingual BERT does not officially support. For example, the model achieved 0.65 on Yiddish (yi) and 0.75 on Walloon (Wa) when multilingual BERT was trained on neither of these languages. There are several explanations for this. We hypothesize that low resource languages such as Yiddish, a high German-derived language, and Walloon, a Romance language, benefit from their similarities to other languages within the same language families. For queries such as named entities, it is also possible that some relevant cross-language Wikipedia documents may be multilingual and contain some overlap with the query term untranslated. The details will depend on the query in question.

[12]Language codes: af:Afrikaans, als:Alemannic, am:Amharic, an:Aragonese, ar:Arabic, arz:Egyptian Arabic, ast:Asturian, az:Azerbaijani, azb:Southern Azerbaijani, ba:Bashkir, bar:Bavarian, be:Belarusian, bg:Bulgarian, bn:Bengali, bpy:Bishnupriya Manipuri, br:Breton, bs:Bosnian, bug:Buginese, ca:Catalan, cdo:Min Dong, ce:Chechen, ceb:Cebuano, ckb:Kurdish (Sorani), cs:Czech, cv:Chuvash, cy:Welsh, da:Danish, de:German, diq:Zazaki, el:Greek, eml:Emilian-Romagnol, en:English, eo:Esperanto, es:Spanish, et:Estonian, eu:Basque, fa:Persian, fi:Finnish, fo:Faroese, fr:French, fy:West Frisian, ga:Irish, gd:Scottish Gaelic, gl:Galician, gu:Gujarati, he:Hebrew, hi:Hindi, hr:Croatian, hsb:Upper Sorbian, ht:Haitian, hu:Hungarian, hy:Armenian, ia:Interlingua, id:Indonesian, ilo:Ilocano, io:Ido, is:Icelandic, it:Italian, ja:Japanese, jv:Javanese, ka:Georgian, kk:Kazakh, kn:Kannada, ko:Korean, ku:Kurdish (Kurmanji), ky:Kirghiz, la:Latin, lb:Luxembourgish, li:Limburgish, lmo:Lombard, lt:Lithuanian, lv:Latvian, mai:Maithili, mg:Malagasy, mhr:Meadow Mari, min:Minangkabau, mk:Macedonian, ml:Malayalam, mn:Mongolian, mr:Marathi, mrj:Hill Mari, ms:Malay, my:Burmese, mzn:Mazandarani, nap:Neapolitan, nds:Low Saxon, ne:Nepali, new:Newar, nl:Dutch, nn:Norwegian (Nynorsk), no:Norwegian (Bokmål), oc:Occitan, or:Odia, os:Ossetian, pa:Eastern Punjabi, pl:Polish, pms:Piedmontese, pnb:Western Punjabi, ps:Pashto, pt:Portuguese, qu:Quechua, ro:Romanian, ru:Russian, sa:Sanskrit, sah:Sakha, scn:Sicilian, sco:Scots, sd:Sindhi, sh:Serbo-Croatian, si:Sinhalese, simple:Simple English, sk:Slovak, sl:Slovenian, sq:Albanian, sr:Serbian, su:Sundanese, sv:Swedish, sw:Swahili, szl:Silesian, ta:Tamil, te:Telugu, tg:Tajik, th:Thai, tl:Tagalog, tr:Turkish, tt:Tatar, uk:Ukrainian, ur:Urdu, uz:Uzbek, vec:Venetian, vi:Vietnamese, vo:Volapük, wa:Walloon, war:Waray, wuu:Wu, xmf:Mingrelian, yi:Yiddish, yo:Yoruba, zh:Chinese

7.3.4 Results on MULTI-8

Single-language retrieval		
Model	$\{\text{BM}_{S \to T}\}$	MM
Train	$q^X = S_{train}, d^Y = T_{train}$	$q^X = A_{train}, d^Y = A_{train}$
Evaluation	$q^X = S_{test}, d^Y = T_{test}$	
Mix-language retrieval		
Model	$\{\text{BM}_{S \to T}\}$	MM
Train	$q^X = S_{train}, d^Y = T_{train}$	$q^X = A_{train}, d^Y = A_{train}$
Evaluation	$q^X = S_{test}, d^Y = A_{test}$	

Table 7.7: Different ways of using MULTI-8. A refers to the concatenation of all languages used in mixed-language retrieval. S and T refer to the queries/-documents in the source and target language under consideration for the bilingual case (i.e., single-language retrieval similar to BI-139 setups). For either, it is possible to train either bilingual models (BM) based on pairwise data or a multilingual model (MM) based on all language data.

Multilingual IR is a field that has been largely unexplored in recent years. MULTI-8 enables evaluation in two kinds of scenarios (see Table 7.7):

SINGLE-LANGUAGE RETRIEVAL

This scenario is similar to BI-139 in terms of evaluation, i.e., during test, we only have queries in source language $q^X = S_{test}$ and documents in one target language $d^Y = T_{test}$. We divide the MULTI-8 test set into $8 \times 7 = 56$ pairs.

For training, we compare **bilingual model ($\text{BM}_{\text{S} \to \text{T}}$)** trained in every language pair, against a **multilingual model (MM)** trained on data concatenated from all 56 language directions. As we can see in Table 7.8, the MM model per-

q \ d	ar	de	en	es	fr	ja	ru	zh
ar		.65▽	**.60▲**	**.65▲**	.64▽	.65▽	**.60▲**	**.64▲**
de	.75▽		**.75▲**	**.77▲**	**.72▲**	**.72▲**	**.74▲**	**.71▲**
en	**.79▲**	**.82▲**		**.83▲**	**.79▲**	.83▽	.82▽	.82▽
es	**.74▲**	**.72▲**	**.76▲**		**.75▲**	.74▽	**.74▲**	.74▽
fr	**.75▲**	**.75▲**	**.76▲**	**.79▲**		.75▽	**.74▲**	.76▽
ja	.71▽	**.68▲**	**.67▲**	**.68▲**	.67▽		.69▽	.70▽
ru	.73▽	**.71▲**	**.71▲**	**.73▲**	.73▽	.72▽		**.71▲**
zh	**.67▲**	**.67▲**	**.63▲**	**.66▲**	.66▽	**.64▲**	**.66▲**	

Table 7.8: MULTI-8 single-language retrieval results of bilingual models (BM). The rows are the source query language, and the columns are the target document language. The up arrows next to NDCG@10 scores indicate instances where the multilingual model (MM) outperforms the bilingual models.

forms better than the respective BM models in most language directions. This suggests that multilingual training is a promising research direction even for single-language retrieval.

MIX-LANGUAGE RETRIEVAL

	ar	de	en	es	fr	ja	ru	zh
BM	.52	.58	.66	.60	.63	.59	.57	.58
MM	.59	.72	.75	.73	.65	.68	.62	.68
△%	13	23	14	22	16	10	20	13

Table 7.9: MULTI-8 mix-language retrieval results. △% shows percent improvement of MM over BM z-norm.

In this scenario, at test time, we have a single source query $q^X = S_{test}$ and

wish to retrieve documents $d^Y = A_{test}$ which can be in any of the 8 MULTI-8 languages. The multilingual model (MM) can be applied directly, but the bilingual model (BM) requires some modifications. One can run multiple BM, one for each target language, then merge the resulting document lists (Savoy, 2003; Tsai et al., 2008). A common strategy, which we adopt here, is to z-normalize the output scores and rank all the test documents based on z-scores.

As seen in Table 7.9, the multilingual model performs significantly better than the ensembled/merged bilingual models. The average NDCG@10 of the multilingual model is 0.684, which is 17.1% than bilingual models with a z-score merging strategy.

7.4 Discussions

7.4.1 Is CLIRMatrix a "good" IR collection?

There are existing works that focus on evaluating the qualities of IR test collections. We first check our dataset against some general "rule of thumb" criteria that an ideal test collection would meet. According to (Jones and Van Rijsbergen, 1976), CLIRMatrix generally fulfills the conditions of a good IR dataset:

1. CLIRMatrix meets the criteria that an ideal collection should have at

least 10,000 documents because we discard languages with less than 10,000 Wikipedia articles.

2. Each dev and test set in CLIRMatrix has 1,000 queries, more than the acceptable number of at least 250 queries.

3. The queries and documents of CLIRMatrix cover a range of domains, fulfilling both variety and homogeneity criteria.

4. CLIRMatrix fulfills the conditions of varieties in type, source, origins, time, and natural language.

Nevertheless, CLIRMatrix and existing CLIR datasets such as CLEF 2000-2003 are far from perfect for two reasons (Voorhees, 2001): First, there is the issue of incompleteness, where IR collections do not have complete judgments. As IR collections can contain millions of documents, it is infeasible for human annotators to thoroughly find all relevant documents for every query. Therefore, evaluation campaigns usually use the pooling method, which gathers the top documents returned by systems from participants and only sends those subsets of documents for human annotation. CLIRMatrix uses a pseudo-pooling method that retrieves the top 100 documents with BM25, creates discrete labels for them, and assumes the other documents are irrelevant. The issue with these approaches is that any relevant document not retrieved by any participant's system would be falsely labeled as irrelevant. This is especially

problematic in the case of CLIR, where models for many language directions are relatively under-explored and suffer from low recall. Further, participants tend to submit more results for high-resource languages and fewer results for low-resource languages, which causes the sizes and diversities of pools to be unevenly distributed across different language directions, leading to potential bias in judgments.

Second, the idea of relevancy is highly subjective, and there is the issue of inconsistency where different human annotators disagree on what is relevant and what is not. This is especially challenging in the case of building a multilingual IR dataset, where different annotators with different backgrounds assess the same query in different languages. CLIRMatrix mitigates this issue by standardizing the retrieval process with BM25 and Jenks natural breaks optimization and enforcing consistent relevance judgments by propagating the same relevance judgment labels for documents in different languages.

Despite having the issues above, Voorhees (2001) empirically shows that comparative retrieval results are stable, proving the validity of using these IR test sets for comparative evaluations.

7.4.2 Limitations of Datasets

We acknowledge CLIRMatrix has potential limitations:

1. Using titles as the queries would limit its use case to ad-hoc information retrieval with short queries. Other interesting settings not covered by these datasets are ad-hoc search with longer text queries and semantic information retrieval that requires parsing the meanings of queries. For example, CLIRMatrix contains annotated documents for the "Barack Obama" but not for the "US President who plays basketball." However, our proposed algorithm is flexible and can be easily adapted to generate synthetic datasets for new query types.

2. CLIRMatrix's reliance on the inter-language information in Wikidata would subject its extraction results to the annotation errors and data incompleteness associated with Wikidata.

3. A document in one language might not be one-to-one mapped to a document in another language. For example, the English articles "Wind Power" and "Wind Energy" are mapped to the same article "Windenergie" in German. When building the German-English bilingual IR dataset and propagating the relevance labels from German documents to English documents, the article "Wind Power" would be unfairly penalized because it does not exist in the German Wikipedia.

Despite the limitations above, we believe CLIRMatrix is still a valuable resource for cross-lingual information retrieval, especially for language direc-

tions without sufficient training data.

7.5 Related Work

Information retrieval (IR) has made a tremendous amount of progress, shifting focus from traditional bag-of-world retrieval functions such as tf-idf (Salton and McGill, 1986) and BM25 (Robertson et al., 2009), to neural IR models (Guo et al., 2016b; Hui et al., 2018; McDonald et al., 2018) which have shown promising results on multiple monolingual IR datasets. Recent advances in pre-trained language models such as BERT (Devlin et al., 2019) have also led to significant improvements in IR tasks. For example, (MacAvaney et al., 2019) achieves state-of-the-art performances on benchmark datasets by incorporating BERT's context vectors into existing baseline neural IR models (McDonald et al., 2018). Training on synthetic is also a common practice, e.g., (Dehghani et al., 2017b) show that supervised neural ranking models can significantly benefit from pre-training on BM25 labels.

Cross-lingual Information Retrieval (CLIR) is a sub-field of IR that is becoming increasingly important as new documents in different languages are being generated every day. The field has progressed from translation-based methods (Zhou et al., 2012; Oard, 1998; McCarley, 1999; Yarmohammadi et al., 2019) to recent neural CLIR models (Vulić and Moens, 2015; Litschko et al.,

2018; Zhang et al., 2019) that rely on cross-lingual word embeddings. In contrast to the wide availability of monolingual IR datasets (Voorhees, 2005; Craswell et al., 2020), cross-lingual and multilingual IR datasets are scarce. Examples of the widely used CLIR datasets are the CLEF 2000-2003 collection (Ferro and Silvello, 2015), which focuses primarily on European languages, and IARPA MATERIAL/OpenCLIR collection (Zavorin et al., 2020), which focus on a few low-resource language directions. Creating a CLIR dataset for more language directions remains an open challenge.

Extracting CLIR datasets from Wikipedia has been explored in previous work. (Schamoni et al., 2014b) build a German–English bilingual CLIR dataset from Wikipedia, which contains 245,294 German queries and 1,226,741 English documents. They convert the first sentences from German Wikipedia documents into queries and follow Wikipedia's interlanguage links to find relevant documents in English. In chapter 5, we apply the same techniques and release a larger CLIR dataset that contains English queries and relevant documents in 25 languages. Both datasets truncate the documents to the first 200 tokens and rely on bidirectional inter-article links to find partially relevant documents. The contribution of CLIRMatrix differs in three important aspects: (i) BI-139 is a significantly larger dataset, covering more languages and documents. (ii) MULTI-8 provides a new multilingual retrieval setup not previously available. (iii) We argue that our method can reliably find more rel-

evant documents by propagating search results from monolingual IR systems to other languages via Wikidata. This is, in contrast, to directly using bidirectional links extracted from Wikipedia documents to determine relevance, which is much sparser. Further, our method allows for more finer-grained levels of relevance (e.g., as opposed to binary relevance), making the dataset more challenging.

7.6 Conclusion

This chapter presents *CLIRMatrix*, the largest and the most comprehensive collection of bilingual and multilingual CLIR datasets to date. The *BI-139* dataset supports CLIR in 139×138 language pairs, whereas the *MULTI-8* dataset enables mix-language retrieval in 8 languages. Many supported language directions allow the research community to explore and build new models for many more languages, especially the low-resource ones. Our mix-language retrieval experiments on MULTI-8 show that a single multilingual model can significantly outperform the combination of multiple bilingual models.

While there are some limitations associated with the choice of queries and issues with Wikipedia and Wikidata, we believe our dataset is beneficial for future research in developing bilingual and multilingual IR models, especially

for language directions with little or no annotated CLIR dataset.

Chapter 8

Exploiting CLIRMatrix Datasets for Domain Adaptation on New Task

8.1 Introduction

In chapter 5, 6 and 7, we show that end-to-end neural CLIR systems are effective when we have sufficient training data in the language pair and domain of interest. However, previous results are primarily based on synthetic datasets and might not necessarily reflect the performances of real-world CLIR datasets. This chapter explores the effectiveness of transferring models trained on CLIRMatrix datasets to CLEF 2000-2003, a collection of multilingual datasets in the news domain. We also explore several domain-adaptation strategies when building end-to-end neural CLIR systems in scenarios with few or no training examples in the domain of interest.

The main findings of this chapter are:

- Our experiments on benchmark CLEF 2003 datasets show that we can *only* get good performance with modular CLIR systems if there are sufficient parallel sentences to train upstream translation systems. In contrast, direct modeling systems used in zero-shot settings outperform modular systems without sufficient parallel sentences.

- Continuing pre-training off-the-shelf multilingual BERT on in-domain documents and training CLIR systems on synthetic in-domain labels do not perform well.

- We observe significant improvements if we pre-train CLIR systems on

CLIRMatrix before fine-tuning those models on in-domain data.

- We shed light on the performance difference between the query and document translation approaches with state-of-the-art neural machine translation and information retrieval systems.

8.2 Experiment Setup

8.2.1 Modular CLIR Systems

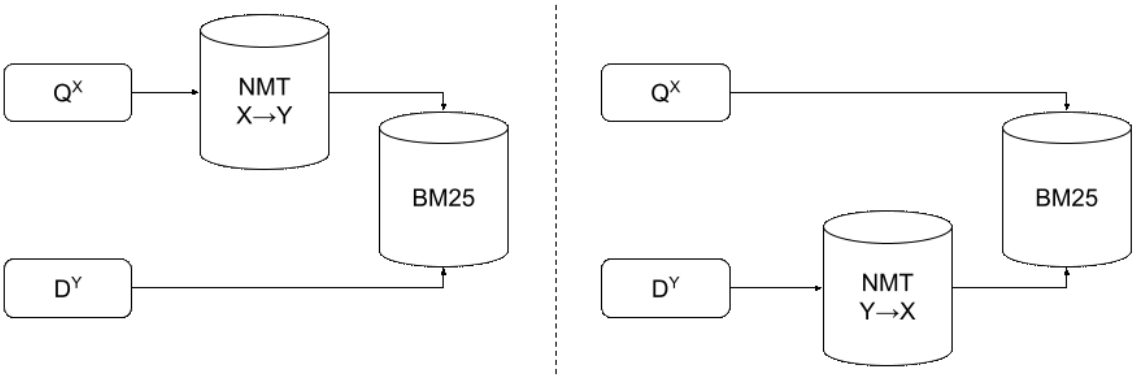


Figure 8.1: System pipelines of modular CLIR systems. (Left) The query translation approach translates the queries into the same language as the documents. (Right) The document translation approach translates the documents into the same language as the queries. Both approaches use BM25 to retrieve relevant documents.

The system pipelines of our modular CLIR systems are illustrated in Figure 8.1. A modular CLIR system consists of two components:

The first component is a machine translation system (Section 8.2.1) that translates either the queries to the same language as the documents (query

translation approach) or the documents to the same language as the queries (document translation approach). For this, we train neural machine translation systems based on the transformer architecture (Vaswani et al., 2017) on publicly available parallel corpora.

The second component is a monolingual IR system (Section 8.2.1) that ranks and retrieves the documents based on translated queries (or the translated documents based on original queries). For this, we choose Elasticsearch[1], which uses the BM25 algorithm (Section 2.2.2) to handle monolingual retrievals.

PART 1: NEURAL MACHINE TRANSLATION MODELS

We train several neural machine translation (NMT) systems on publicly available parallel sentences from the OPUS open parallel corpus (Tiedemann, 2012), in both directions for every unique language pair in our evaluation CLIR datasets.

In more detail, we download all available parallel sentences for the following language directions: Russian-English (Ru-En), Chinese-English (Zh-En), German-English (De-En), Russian-German (Ru-De), and Chinese-German (Zh-De) from https://opus.nlpl.eu/. We randomly shuffle the parallel sentences into train, validation, and test splits for each language direction in the ratio of 1000 to 1 to 1. We first train byte-pair-encoding (BPE) (Sennrich et al.,

[1] https://www.elastic.co/downloads/elasticsearch

2016b) models on the train data using the SentencePiece toolkit[2] and tokenize all parallel sentences into subword units. For all BPE models, we limit the BPE size to 32,000. We then use Fairseq (Ott et al., 2019) to train transformer-based (Vaswani et al., 2017) NMT models on the preprocessed text data, using the recommended hyperparameters:

```
fairseq-train ${DATA_DIR}
    --arch transformer_wmt_en_de --sh-decoder-input-output-embed \
    --optimizer adam --adam-betas '(0.9, 0.98)' --clip-norm 0.0 \
    --lr 5e-4 --lr-scheduler inverse_sqrt --warmup-updates 4000 \
    --dropout 0.3 --weight-decay 0.0001 \
    --criterion label_smoothed_cross_entropy --label-smoothing 0.1 \
    --max-tokens 8000 \
    --no-epoch-checkpoints \
    --skip-invalid-size-inputs-valid-test \
    --patience 20 \
    --save-dir $MODEL_DIR
```

Since we are also interested in the performance of CLIR systems in low-resource language directions, we simulate NMT models in low-resource settings by training them on sub-sampled parallel sentences. For each language direction, we train several low-resource NMT models on 10 thousand (10K), 100 thousand (100K), and 1 million (1M) random parallel sentences.

The performances of our NMT models measured in BLEU score (Post, 2018), and the statistics of different parallel sentences corpora are shown in Table 8.1.

[2] https://github.com/google/sentencepiece

LD	10K	100K	1M	All	#train	#dev	#test
Ru→En	1.54	5.84	37.77	48.15	66,392,941	65,829	65,829
En→Ru	0.77	5.84	32.54	40.15			
Zh→En	2.77	5.80	7.90	34.70	27,733,721	27,665	27,665
En→Zh	0.00	0.36	8.10	27.86			
De→En	0.74	4.75	32.41	43.18	87,597,895	87,329	87,329
En→De	0.73	3.70	29.33	38.37			
Ru→De	0.37	1.91	9.36	13.56	11,698,660	11,447	11,447
De→Ru	0.07	1.20	6.56	10.50			
Zh→De	4.15	7.27	N/A	67.47	424,864	399	399
De→Zh	0.00	0.00	N/A	3.24			

Table 8.1: NMT BLEU scores for different training settings (10K, 100K, 1M, and all sentences for each language direction (LD)) and statistics of parallel sentences corpora.

NMT models benefit from more training data, achieving higher BLEU scores when trained on more parallel sentences. The performances of Zh-De and De-Zh vary drastically, with the best Zh-De model getting a BLEU score of 67.47 while the best De-Zh gets a BLEU score of 3.24. We hypothesize that it is caused by the relatively smaller size of the train and test data.

PART 2: MONOLINGUAL IR SYSTEMS

The translation step effectively reduces a CLIR task to a monolingual IR task. We can then deploy any effective monolingual IR model to rank and retrieve the translated queries or documents. Due to its speed and effectiveness, we use the BM25 scoring mechanism to handle monolingual retrievals for all modular CLIR systems. The BM25 score of a document d given query q is de-

fined:

$$BM25(q,d) = \sum_{i=1}^{n} IDF(q_i).\frac{TF(q_i,d).(k_1+1)}{TF(q_i,d)+k_1.(1-b+b.\frac{|d|}{avgdl})}$$

where $TF(q_i, d)$ is the term frequency of q_i in document d, $IDF(q_i)$ is the inverse document frequency of q_i, $|d|$ is the length of document d in terms of number of word tokens and $avgdl$ is the average length of all candidate documents. b and k_1 are hyper-parameters which we use the default values of 0.75 and 1.2 respectively. We use the implementation of BM25 in Elasticsearch to index and retrieve documents, and use similar settings as the experiments in chapter 7.

8.2.2 Direct Modeling CLIR Systems

A strong baseline for direct modeling is the vanilla BERT ranker model (MacAvaney et al., 2019), which also performs well on cross-lingual information retrieval task (Sun and Duh, 2020). The model uses the cross-encoder architecture described in section 2.2.4 which encodes a query-document pair with multilingual BERT (Devlin et al., 2019) and stacks a linear combination layer on top of the [CLS] token. Unfortunately, this approach has a time complexity of $\mathcal{O}(MN)$ where M is the number of queries, and N is the number of documents, making it unsuitable for running on our experiments that contain millions of queries-documents pairs. Therefore, we experiment with another approach that uses the bi-encoder architecture.

NEURAL ARCHITECTURE

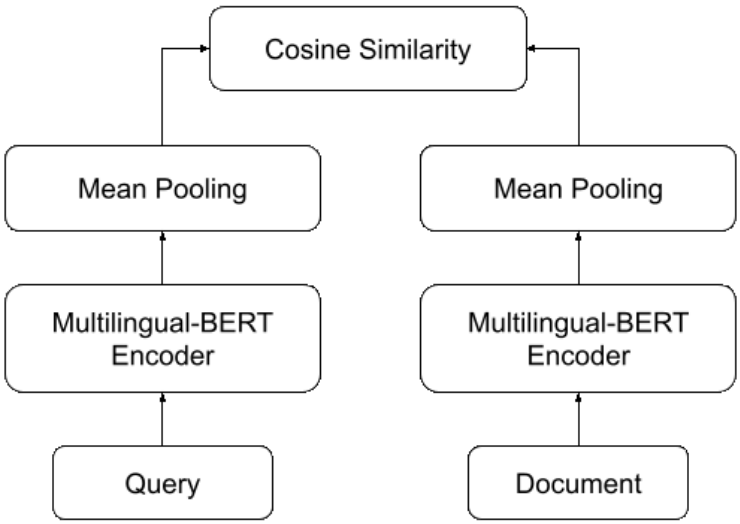


Figure 8.2: The bi-encoder neural architecture for CLIR, where query and document are encoded separately with the same multilingual BERT encoder.

The neural architecture of our bi-encoder approach is inspired by the Sentence-BERT method (Reimers et al., 2019). As shown in Figure 8.2, the main idea of this approach is to encode query and document independently with the same multilingual BERT encoder and then use a mean pooling layer to compress the list of outputs into vector representations. It then computes the cosine similarity between those vector representations to estimate the degree of relevancy between a query and a document. For all experiments, we use the Multilingual Cased version[3] of the BERT-Base contextualized language model, which supports 104 different languages. The dimension of query and document vectors is 768, the same as the hidden size of the encoder layers in BERT.

[3] https://github.com/google-research/bert/blob/master/multilingual.md

The time complexity of this method is improved to $\mathcal{O}(M + N)$ since we only need to encode M queries and N documents. However, we still have to compute cosine similarities MN times to rank and retrieve relevant documents for a given query. Fortunately, the computation overhead to calculate the cosine similarity is significantly lower than to encode query and document pairs with BERT.

OBJECTIVE FUNCTION

We use the cosine similarity loss in the Sentence Transformers[4] python package to optimize our direct modeling CLIR systems. The loss is defined as:

$$MSE\left(sim\left[E(q), E(d)\right], \hat{r}\right) \tag{8.1}$$

where E is a multilingual BERT encoder shared between the queries and documents, sim is a function that computes the cosine similarity between a query embedding and a document embedding, $\hat{r}$ is the relevance label of the query-document pair rescaled to [0, 1] and MSE is the mean squared error.

TRAINING PROCEDURES

We train our direct modeling CLIR systems in a pointwise manner where every training example is a query-document pair. As some CLIR datasets are

[4] https://www.sbert.net/

labeled with multiple degrees of relevancy, we map those relevant labels to [0, 1] using the following formula:

$$\hat{r}_i = \frac{r_i}{r_{max}}$$

where r_i is the relevance label for the i-th example and r_{max} is the max value of relevance labels in the dataset.

We convert each pair of query and document into embeddings using the same multilingual BERT, compute the cosine similarities between them, and use the cosine similarities loss defined in the previous subsection to optimize the CLIR model. We trained all models for 20 epochs with a batch size of 8 and used an early stopping patience of 20 steps on a server instance with a single GTX TI1080 GPU.

INFERENCE

Given a test CLIR dataset and a trained CLIR model, we first convert all queries and documents to vectors with the CLIR model. We rank documents based on the cosine similarities between query embedding and all document embeddings. We only keep the top 2000 documents for each query for practical reasons.

8.2.3 Train and Test Datasets

TRAIN DATA FOR DIRECT MODELING

We use the Multi-8 datasets from the CLIRMatrix collection (Chapter 7) for training. MULTI-8 is a multilingual CLIR dataset comprising queries and documents jointly aligned in 8 languages: Arabic (Ar), German (De), English (En), Spanish (Es), French (Fr), Japanese (Ja), Russian (Ru), and Chinese (Zh). The datasets come with train, dev, and test splits where each train split contains 10,000 queries, while the dev and test splits contain 1000 queries each. This dataset uses seven levels of relevance labels, from 0 to 6, where 6 means a document is highly relevant to a query and 0 means a document is not relevant.

TEST DATA FOR MODEL EVALUATION

Language Direction	Number of queries	Number of documents
Ru → En	32	166,740
Zh → En	50	
De → En	50	
En → De	50	294,805
Ru → De	50	
Zh → De	50	

Table 8.2: Statistics of selected CLEF 2003 test sets

From the Multilingual-8 dataset used in the CLEF 2003 evaluation campaign (CLEF 2000-2003, 2003), we carefully select six language directions as

the test sets for our experiments: Russian-English (Ru-En), Chinese-English (Zh-En), German-English (De-En), Russian-German (Ru-De), English-German (En-De) and Chinese-German (Zh-De). This set of language directions is the intersection between the language directions in the CLIRMatrix MULTI-8 dataset and the language directions from CLEF 2003.

60 similar topics are available for every source language, except for Russian, with only 37 topics. Each topic contains a brief title, a one-sentence description, and a more complex narrative explaining the topic's requirements. In the original CLEF 2003 evaluation campaign, participants were expected to construct queries from the given topics. In contrast, we use a more straightforward setup where each query is just the string concatenation of the topic title and its description. We randomly sample 50 topics (32 topics for Russian) for evaluation and reserve the other 10 topics for domain adaptation experiments. Care has been taken to ensure that topics in the test set of one language direction do not overlap the reserved topics of another language direction.

The documents are newspaper articles or news agency documents stored in the SGML format. As the documents might contain irrelevant tags such as account id, date, and page, we only extract and concatenate text from the following tags: title, ti, ld, text, tx, and body. In contrast to CLIRMatrix, which uses seven levels of relevance, our test set uses a binary relevance scale where 1 represents relevant and 0 means irrelevant.

8.2.4 Evaluation Metric

We report all experiment results in NDCG@100 defined as:

$$\text{DCG@100} = \sum_{i=1}^{100} \frac{2^{r'_i} - 1}{log_2(i+1)}$$

$$\text{IDCG@100} = \sum_{i=1}^{100} \frac{2^{r_i} - 1}{log_2(i+1)}$$

$$\text{NDCG@100} = \frac{\text{DCG@100}}{\text{IDCG@100}}$$

where r'_i is the relevance judgment label of the i-th document in the predicted document ranking and r_i is the relevance judgment label of the i-th document in the actual document ranking.

8.3 Baseline Results

As discussed in Section 8.2.1, our baseline results are based on the modular approach. We train eight neural machine translation (NMT) models for each language direction on 10K, 100K, 1M, and all parallel sentences. We train an NMT system for query translation and another NMT system for document translation for every language resource setting. For each set of queries and documents translated to the same language, we use BM25 to index and rank relevant documents for every query. NDCG@100 results are displayed in Table 8.3 and Figure 8.3.

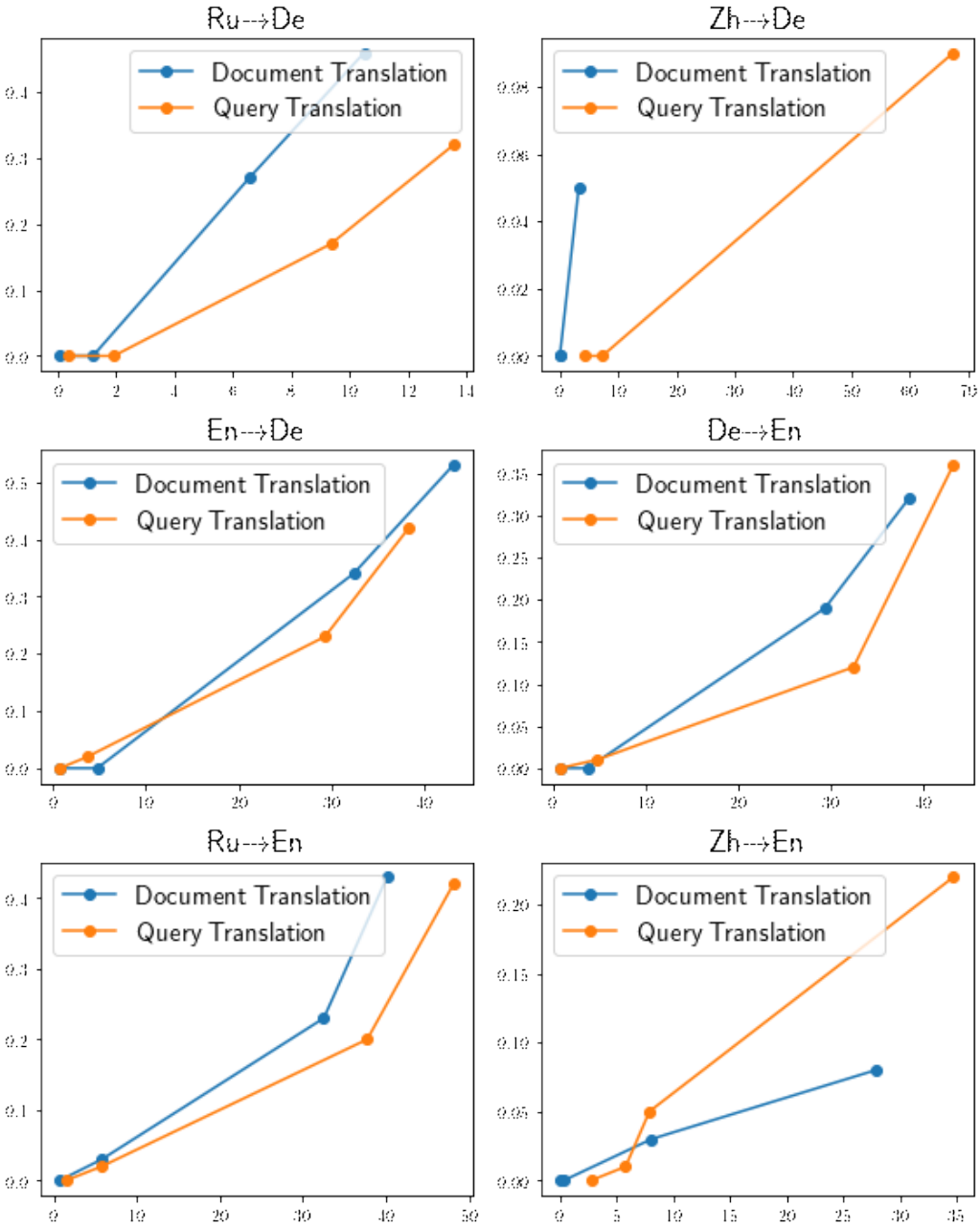


Figure 8.3: Plot of NDCG@100 against BLEU for six language directions from CLEF 2003 Multilingual-8 dataset..

LD	10K		100K		1M		All	
	QT	DT	QT	DT	QT	DT	QT	DT
Ru→De	0.00	0.00	0.00	0.00	0.17	0.27	0.32	**0.46**
Zh→De	0.00	0.00	0.00	0.00	N/A	N/A	0.09	0.05
En→De	0.00	0.00	0.02	0.00	0.23	0.34	0.42	**0.53**
De→En	0.00	0.00	0.01	0.00	0.12	0.19	**0.36**	0.32
Ru→En	0.00	0.00	0.02	0.03	0.20	0.23	0.42	**0.43**
Zh→En	0.00	0.00	0.01	0.00	0.05	0.03	**0.22**	0.08

Table 8.3: Modular approach results in six language directions from CLEF 2003 Multilingual-8 dataset. The best results are bolded.

PERFORMANCE OF DOCUMENT RETRIEVAL DEPENDS ON THE QUALITY OF TRANSLATION SYSTEM

As we can see in Table 8.3 and Figure 8.3, it is clear that translating queries and documents with better NMT systems (systems with higher BLEU scores) always lead to better performances in the downstream monolingual IR retrievals. For example, the BLEU scores for the English-German (En-De) and German-English (De-En) NMT systems are 38.37 and 43.18, respectively, when trained on all available parallel sentences. Therefore, the downstream monolingual IR systems obtain remarkable NDCG@100 scores of 0.42 and 0.53 for query-translation and document translation based on the outputs of those NMT systems. On the other hand, NMT systems trained under low-resource settings suffer from low BLEU scores. Consequently, the downstream monolingual IR systems are not effective at retrieving relevant documents, getting NDCG@100

scores close to zero.

To better understand the poor performance of the modular CLIR systems trained under low-resource settings, we examine one sample translation. For the Chinese query, "臭氧層的破壞那些臭氧層的破洞不是污染所造成的?" which should be translated to "Destruction of the Ozone Layer. Are those holes in the ozone layer not caused by pollution?", the translations from the various NMT systems are:

- ""D" they have been "D"""?" (10K)
- "How does the State be able to protect the environment?" (100K)
- "How does the State be able to protect the environment?" (1M)
- "The breakdown of the ozone layer is not caused by pollution?" (All)

The NMT system trained on 10K parallel sentences completely failed, producing a gibberish sentence that is neither adequate nor fluent. While the translations from the NMT systems trained on 100K and 1M sentences output are fluent, those translations are inadequate and fail to carry the semantic meaning of the original Chinese sentence. The last translated sentence from the NMT system trained on all parallel sentences is fluent and almost correctly translates the original sentence. Examining the sample translations above, the query translations of the first 3 NMT systems are incorrect, affecting the downstream monolingual IR systems' ability to retrieve irrelevant documents.

Our experiment results show that the success of a modular CLIR system heavily relies on the quality of the upstream translation system. However, as machine translation is not yet effective in all language directions (especially those with low availability of parallel sentences for training), we argue that the modular CLIR approach is not scalable for most language directions.

QUERY OR DOCUMENT TRANSLATION?

We use this opportunity to revisit the decades-long debate (Section 2.3.1) on whether the query translation approach is better than the document translation approach. From our empirical results shown in Figure 8.3, we argue that the better translation approach is highly situational: The document translation approach outperforms the query translation approach for Russian-German (Ru-De), English-German (En-De), and Russian-English (Ru-En). In contrast, the query translation approach is better for Chinese-German (Zh-De), German-English (De-En), and Chinese-English (Zh-En). Chinese-German (Zh-De) is an extreme case where the NMT system for query translation has a much higher BLEU score than that of the NMT system for document translation, explaining the significant difference in performance between the query and document translation approaches.

8.4 Domain Adaptation on New Task

The previous section shows that we can potentially build modular CLIR systems that do decently well on the CLEF 2003 Multilingual-8 dataset under ideal situations in which we have many quality parallel sentences for the language pair of interest. However, we also show that modular CLIR systems cannot retrieve relevant documents with NMT models trained under low-resource settings.

This section now focuses on the direct modeling approach, where we build end-to-end neural CLIR models that avoid the need to train machine translation models. Unfortunately, The CLEF 2003 dataset is small, with only 30-60 examples, which means we do not have enough in-domain examples for training. Therefore, we train end-to-end models on large-scale synthetic datasets (Sun and Duh, 2020) and explore strategies that can adapt these models to the data from the news domain of CLEF 2003.

We examine domain adaptation in 4 different scenarios:

1. No in-domain data
2. No in-domain queries and relevance labels, have in-domain documents
3. Have some in-domain queries and documents, but no relevance labels
4. Have some in-domain labeled datasets

Table 8.4 provides a high-level summary of the different scenarios and the domain adaptation.

Scenario	Queries	Documents	Labels	Approach
1	X	X	X	Zero-short transfer learning
2	X	✓	X	Fine-tune BERT on documents
3	✓	✓	X	Fine-tune on synthetic labels
4	✓	✓	✓	Fine-tune on in-domain data

Table 8.4: Summary of various scenarios and the approach we are exploring.

8.4.1 Results and Analysis

SCENARIO 1 (S1)

LD	Modular		Direct Modeling
	Low-Resource	High-Resource	
Ru→De	0.00	**0.46**	0.20
Zh→De	0.00	0.09	**0.12**
En→De	0.00	**0.53**	0.28
De→En	0.01	**0.36**	0.10
Ru→En	0.03	**0.43**	0.21
Zh→En	0.01	**0.22**	0.10

Table 8.5: Results on the 6 language directions from the CLEF 2003 test set for modular and direct modeling approaches. For the modular approach, we show the best NDCG@100 score for **Low-Resource** modular systems (using NMT trained on either 10K or 100K parallel sentences) and **High-Resource** modular systems (using NMT trained on either 1M or all parallel sentences).

We assume we have no in-domain training data in this scenario, i.e., zero

query, document, and labels related to the CLEF 2003 dataset. For this, we explore a simple zero-shot transfer learning strategy where we train a bi-encoder variant of the Multilingual-BERT ranker model on the CLIRMatrix MULTI-8 dataset for every language direction and directly evaluate them on the CLEF 2003 test sets.

From the results in Table 8.5, the direct modeling models generally perform worse than the high-resource modular models. The only exception is Chinese-German (Zh-De), where the direct modeling model gets an NDCG@100 of 0.12, while the best modular model only gets an NDCG@100 of 0.09. The low scores for Zh-De are not surprising since there are only 424,864 parallel sentences available to train NMT models for that language direction. On the other hand, the direct modeling CLIR systems *significantly outperform the modular CLIR systems trained in low-resource language settings*. Combining with the analysis in Section 8.3, we conclude that modular CLIR systems are only effective if we have enough parallel sentences to train their upstream translation systems. Zero-shot transfer learning is the better solution for low-resource language settings.

ERROR ANALYSIS

We hope to gain insights into what kind of queries benefit from CLIRMatrix. For this, we extract the top three English, Chinese, or German queries with

LD	NMT + BM25	S1(DM)	S2 0.6M/18M	S3 QT/DT	S4
Ru→De	0.46	0.20	0.00/0.00	0.00/0.26	**0.57**
Zh→De	0.09	0.12	0.00/0.00	0.00/**0.14**	0.13
En→De	**0.53**	0.28	0.00/0.05	0.00/0.00	0.52
De→En	0.36	0.10	0.00/0.03	0.00/0.00	**0.38**
Ru→En	0.43	0.21	0.00/0.00	0.11/0.23	**0.51**
Zh→En	0.22	0.10	0.00/0.00	0.00/0.00	**0.50**

Table 8.6: NDCG@100 results on six language directions from CLEF 2003 for various scenarios.

the highest NDCG@100 scores and the bottom three queries with the lowest NDCG@100 scores for English (Table 8.8 and German (Table 8.7) documents.

To our surprise, the performances of the queries do not depend on the source or target language. Instead, they rely more on the context of the queries. For example, query 143 is always among the queries with the top three NDCG scores, regardless of language directions. Queries 176 and 181 also appear in the top three for many language directions. We observe similar trends in the queries with the lowest NDCG@100 scores: Query 144 appears at the bottom of three language directions. Language directions with the same target languages have overlapping queries with NDCG@100 scores of zero.

We derive two hypotheses explaining the phenomenons: First, CLIR models based on multilingual BERT have aligned texts in different languages into similar regions of the multilingual embedding space, especially after fine-tuning on the CLIRMatrix datasets. This explains why similar queries have simi-

Q Lang	QID	Query Text	N@100
Chinese	176	蘇梅克－李維慧星及木星找出有關蘇梅克－李維慧星的撞擊及其對木星的影響。	0.51
	143	北京婦女領導人會議受到多數會議代表的爭議，北京婦女領導人會議面臨失敗的危機。	0.50
	181	法國核試找出有關國際壓力促使法國停止核試的報導。	0.43
	146	日本的速食那些北美速食連鎖業在日本經營許多聯營餐廳？	0.00
	145	日本稻米進口找出討論日本第一次稻米進口的原因和結果的文章。	0.00
	144	塞拉利昂叛亂及鑽石叛亂及政治不穩定對於塞拉利昂鑽石工業所帶來的影響？	0.00
English	176	Shoemaker-Levy and Jupiter Find reports on the break-up of the Shoemaker-Levy comet and its impact on the planet Jupiter.	0.60
	143	Women's Conference Beijing Controversial positions by a number of delegates meant that the Women's Conference in Beijing risked failure.	0.55
	199	Ebola Epidemic in Zaire Find reports on preventive measures taken after the outbreak of the Ebola epidemic in Zaire.	0.52
	160	Scotch Production Consumption Documents will discuss the amount of scotch consumed by Scots relative to the amount of scotch that is exported from Scotland.	0.00
	146	Fast Food in Japan What North American fast food chains have a large number of franchise restaurants in Japan?	0.00
	144	Sierra Leone Rebellion and Diamonds What have been the effects of rebellions and other political instability on the Sierra Leone diamond industry?	0.00

Table 8.7: Queries with highest and lowest NDCG@100 scores for **German** documents.

Q Lang	QID	Query Text	N@100
Chinese	181	法國核試找出有關國際壓力促使法國停止核試的報導。	0.46
	143	北京婦女領導人會議受到多數會議代表的爭議，北京婦女領導人會議面臨失敗的危機。	0.33
	176	蘇梅克－李維慧星及木星找出有關蘇梅克－李維慧星的撞擊及其對木星的影響。	0.24
	149	教宗訪問斯里蘭卡找出有關教宗先前對於佛教發表的言論，在訪問斯里蘭卡時所引起的抗議或問題的報導。	0.00
	148	臭氧層的破壞那些臭氧層的破洞不是污染所造成的？	0.00
	144	塞拉利昂叛亂及鑽石叛亂及政治不穩定對於塞拉利昂鑽石工業所帶來的影響？	0.00
German	181	Französische Atomtests Finde Berichte über den internationalen Druck zur Beendigung französischer Atomtests.	0.45
	143	Frauenkonferenz in Peking Wegen umstrittener Positionen einiger Delegationen drohte die Frauenkonferenz in Peking zu scheitern.	0.40
	189	Hubble und Schwarze Löcher Welche Rolle spielte das Hubble-Teleskop beim Nachweis der Existenz von Schwarzen Löchern?	0.28
	149	Papstbesuch in Sri Lanka Finde Berichte über die Proteste oder Probleme während des Papstbesuches in Sri Lanka, die mit seinen vorangegangenen Erklärungen über den Buddhismus zusammenhängen.	0.00
	148	Schäden der Ozonschicht Welche Löcher in der Ozonschicht sind nicht durch Umweltverschmutzung verursacht worden?	0.00
	141	Briefbombe für Kiesbauer Finde Informationen über die Explosion einer Briefbombe im Studio der Moderatorin Arabella Kiesbauer beim Fernsehsender PRO7.	0.00

Table 8.8: Queries with highest and lowest NDCG@100 scores for **English** documents.

lar NDCG@100 scores, regardless of their languages and the languages of the candidate documents. The second reason is that the bilingual CLIR models are exposed to the same set of queries because we trained them on the CLIRMatrix MULTI-8 datasets, where the queries are jointly aligned in different languages.

SCENARIO 2 (S2)

In this scenario, we assume we do not have any queries and relevance labels from CLEF 2003, but we do have access to the candidate documents. We explore a popular strategy where we continue training a pre-trained multilingual BERT model on all documents in German and English. We first do sentence segmentation on all German and English documents using the syntok[5] library and then use the transformers[6] package to continue the training on pre-trained BERT models on those in-domain sentences. The intuition of this approach is that fine-tuning the pre-trained language models can improve the representations of the in-domain sentences.

We saved the best checkpoint after 600 thousand (0.6M) training steps and the best checkpoint after 18 million (18M) training steps. For each language direction and each fine-tuned multilingual BERT model, we train a DM CLIR model on the CLIRMatrix dataset and evaluate its performance in the same language direction from CLEF 2003. In the *S2* column of Table 8.6, we see that

[5] https://github.com/fnl/syntok

[6] https://huggingface.co/docs/transformers/index

instead of improving the performances of bi-encoder CLIR models fine-tuned on in-domain sentences, those models deteriorate and get NDCG@100 scores close to zero. The trend holds for both the 0.6M checkpoint and 18M checkpoint, and we cannot obtain any good results despite our best efforts. Therefore, based on our empirical results, we would not recommend continuing to train BERT on in-domain documents.

SCENARIO 3 (S3)

This scenario assumes we have some in-domain queries and documents, but we do not have their relevance labels. We experiment with a strategy where we generate synthetic labels with BM25 and machine translation. We reuse the BM25 labels from the system in Section 8.3 and use them to fine tune the CLIR models from scenario 1 for three epochs. In the *S3* column of Table 8.6, we show the NDCG@100 results for models fine-tuned on BM25 labels from query translation (QT) and BM25 labels from document translation (DT). As we can see from the results, systems fine-tuned on the BM25 labels from query translation systems are not effective. On the other hand, systems fine-tuned on BM25 labels from document translation systems are slightly better, seeing improved performances in 3 of the 6 language directions (compared to S1). However, this approach is unstable as the models fail to retrieve relevant documents for the other 3 language directions.

SCENARIO 4 (S4)

This scenario assumes we have some labels from the CLEF 2003 Multilingual-8 dataset to train our models. As mentioned in Section 8.2.3, we reserve around 10 queries for each language direction for training purposes. Therefore, We fine-tune the DM models from scenario 1 on these reserved in-domain (query, document, label) triplets for 3 epochs. In the *S4* column of Table 8.6, the DM models fined tuned on some in-domain data outperform the DM models in scenario 1. Further, these fine-tuned models outperform the best modular models in 4 out of 6 language directions. For the other language directions, the fine-tuned models still perform at levels close to the best modular models. These results show that it is effective to train CLIR models using just a little in-domain data.

DOES PRE-TRAINING CLIR MODELS ON CLIRMATRIX DATA HELP?

LD	BERT + CLIRMatrix + CLEF 2003	BERT + CLEF 2003
Ru→De	**0.57**	0.11
Zh→De	**0.13**	0.04
En→De	**0.52**	0.13
De→En	**0.38**	0.30
Ru→En	**0.51**	0.04
Zh→En	**0.50**	0.12

Table 8.9: NDCG@100 results for CLIR models with and without fine-tuning on CLIRMatrix.

The previous experiment shows we can achieve the best results in some language directions by further fine-tuning CLIR models trained on CLIRMatrix datasets with some in-domain CLEF 2003 data. The question now is how much do those models benefit from the prior training on the CLIRMatrix dataset? To answer this question, we would directly fine-tune the original multilingual BERT model on in-domain training data. We would refer to these models as **BERT + CLEF 2003** and the previous models as **BERT + CLIRMatrix + CLEF 2003**. The difference between these 2 models is shown in Figure 8.4.

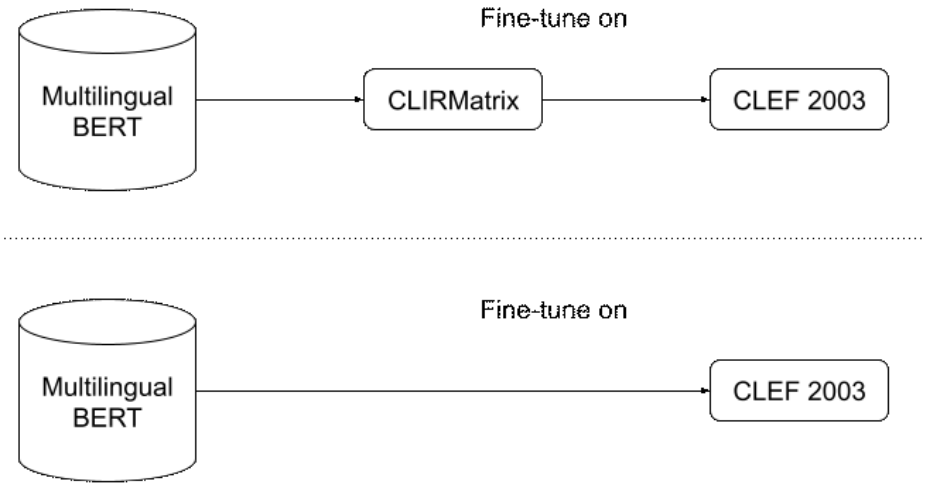


Figure 8.4: (top) BERT + CLIRMatrix + CLEF 2003 (bottom) BERT + CLEF 2003

Results in Table 8.9 show that the **BERT + CLIRMatrix + CLEF 2003** models always perform better than the **BERT + CLEF 2003** models. For example, Ru-De and Ru-En achieve NDCG@100 above 0.50 when fine-tuned on CLIRMatrix, but perform at NDCG@100 of 0.11 and 0.04 without the fine-tuning. The results show that it is beneficial to first pre-train CLIR models

on a synthetic dataset such as CLIRMatrix before fine-tuning these models on some in-domain datasets.

8.5 Conclusion

This chapter explores several strategies when dealing with scenarios with few or no training examples in the domain of interest. When sufficient parallel sentences exist, we can get good performance by building modular CLIR systems that first translate queries and documents into the same language.

In Section 8.4.1, we further investigate different scenarios with varying amounts of in-domain data. In scenario 1, we explore a setup that has no in-domain data. We show that direct modeling systems evaluated in zero-shot transfer learning settings outperform modular systems without sufficient parallel sentences for NMT training. In scenario 2, we assume we only have in-domain documents. We tried continuing pre-training off-the-shelf BERT models on in-domain documents before fine-tuning them on CLIRMatrix datasets but did not observe any significant improvements. In scenario 3, we assume we have queries and documents but no labels. We observed mixed results when we fine-tuned CLIR models on synthetic labels. In scenario 4, we assume we have some in-domain (query, document, label) triplets. We show it is beneficial to first train CLIR systems on synthetic CLIR datasets such as CLIRMatrix be-

fore fine-tuning on in-domain data.

Chapter 9

Conclusions

9.1 Contributions

This dissertation made several contributions to the field of cross-lingual information retrieval. We design and publicly release two synthetic CLIR datasets to the research community and propose and several neural architectures that might be useful for cross-lingual information retrieval.

In Chapter 4, we propose the regularized self-attention ranking network (RSARN), which is a listwise neural approach to the learning to rank problem. We show that we can significantly outperform state-of-the-art ensemble tree-based methods by carefully controlling the weights of self-attention layers over the vector representations of query-document pairs.

In Chapter 5, we release the large-scale CLIR dataset and explore a CLIR model that uses convolutional neural networks to encode and predict the relevance of a document to a query. We show that we can bootstrap bilingual IR models for languages with less training data by using parameter sharing among different language pairs. For example, using the training data for Japanese-English CLIR, we can improve the Mean Average Precision (MAP) results of a Swahili-English CLIR system by 5-7 points.

In Chapter 6, we empirically explore the Multilingual-BERT Ranker Model based on the cross-encoder architecture and show that it outperforms state-of-the-art systems with minimal supervision. We further show that these BERT

ranker models are robust and do not suffer from the partial-input baseline problems observed in other tasks (Poliak et al., 2018; Gururangan et al., 2018).

In Chapter 7, we present CLIRMatrix, a massively large collection of bilingual and multilingual datasets for Cross-Lingual Information Retrieval extracted automatically from Wikipedia. We further show that a single multilingual ranker model trained on multiple language pairs significantly outperforms an ensemble of bilingual ranker models.

In Chapter 8, we show that when sufficient parallel sentences exist, we can get good performance by building modular CLIR systems. We further show that direct modeling systems evaluated in zero-shot transfer learning settings outperform modular systems with insufficient parallel sentences to train decent NMT systems. Finally, we show that it is beneficial to first train CLIR systems on synthetic CLIR datasets such as CLIRMatrix and fine-tune on in-domain data.

9.2 Future Work

This dissertation proposes various datasets and models to improve information retrieval in cross-lingual and multilingual settings, especially for language directions with few or no labeled datasets. Nonetheless, future work that can further advance the field of cross-lingual information retrieval remains.

In Chapter 4, we show that the Regularized Self-Attentive Ranking Network (RSARN) achieves state-of-the-art performances on feature-based learning to rank datasets in both monolingual and multilingual settings. One potential follow-up work is to examine the effectiveness of stacking RSARN on recent transformer-based CLIR models. In contrast to most existing systems that use the pointwise or pairwise approach where each training example is either a (query, document, relevance judgment label) triple or a (query, document 1, document 2, label) quadruplet, the input to a listwise neural IR model is (query, $\{d_1, d_2, d_3, \ldots, d_n\}$, $\{r_1, r_2, r_3, \ldots, r_n\}$) where $\{d_1, d_2, d_3, \ldots, d_n\}$ is a set of candidate documents and $\{r_1, r_2, r_3, \ldots, r_n\}$ contains the relevance judgment labels for those documents. One way to stack RSARN on recent transformer-based CLIR models is shown in Figure 9.1: We first preprocess a list of candidate documents by concatenating each document with their corresponding query before feeding them to the same multilingual BERT encoder. The output is a list of vectors, where each vector represents one query-document pair. The list of vector representations is fed into a second-level RSARN, which focuses on learning the interactions between the document encodings. We can then use listwise objective functions such as ListNet to optimize the whole neural architecture end-to-end. A pitfall of Transformer-based CLIR models is they are computationally expensive models that require GPU accelerators for inference, and GPUs usually only have enough memory for a small set of docu-

ments. Therefore, we should primarily explore these models on CLIR tasks in the re-ranking setup, where we shortlist a much smaller collection of candidate documents using simpler models such as BM25.

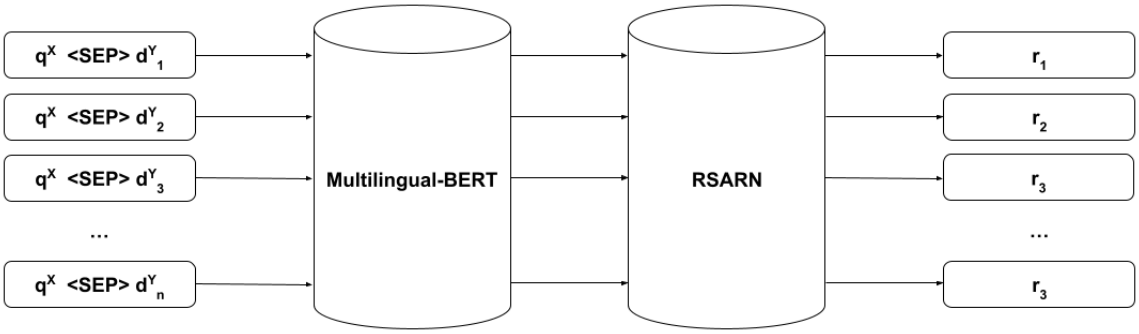


Figure 9.1: Proposed method to stack RSARN on a CLIR model based on Multilingual-BERT

Chapter 5 and Chapter 7 introduce the largest CLIR datasets to date: large-scale CLIR dataset and CLIRMatrix. One potential future work is to extend these datasets to even more languages, especially for extreme low-resource languages such as Tigrinya and Twi. This might require figuring out how to extract queries and documents from text sources beyond Wikipedia. For example, we can explore and identify other useful text sources, such as African websites with cross-language links to articles written in other languages. We can then index and retrieve articles in one language with the same pipeline as CLIRMatrix and propagate relevance labels to articles in other languages.

Given that the extraction pipeline of CLIRMatrix is flexible, another line of work is to collect and re-run the CLIRMatrix extraction pipeline with actual user-generated queries instead of synthetic queries such as Wikipedia titles.

However, search engine companies generally do not release query logs for privacy reasons. One workaround is incentivizing users to install browser plugins that log search queries and only collect data from users who consent.

Chapter 6, 7 and 8 show that building CLIR systems on pre-trained contextualized language models lead to state-of-the-art performances on different CLIR datasets. However, these CLIR models contain millions of parameters and might be too computationally expensive for real-world applications. For example, a CLIR system based on a pre-trained language model with 12 encoder layers could take at least 120 milliseconds to encode just a pair of multilingual sentences on the CPU. Another CLIR system based on a larger pre-trained language model with 24 encoder layers could take more than 377 milliseconds to do the same job (Sun et al., 2021). Based on the numbers above, a naive solution that encodes a list of query-document pairs sequentially could easily take minutes or even hours to rank and retrieve documents for just one query. Therefore, interesting future work is to apply model compression techniques to these neural CLIR models.

Model compression is a popular field of work that focuses on reducing the number of parameters and computations in big neural models without significantly hurting performance. There are several genres of compression techniques: Knowledge distillation is a model compression technique that uses the outputs of a larger teacher model to train a smaller student model. In the con-

text of CLIR, we can pre-train larger CLIR models on CLIRMatrix datasets and then use the large models to train the learning of smaller CLIR models with the same CLIR datasets. Pruning is another popular sub-field of model compression which focuses on removing redundant computations from the architecture of neural models. Some techniques worth trying are layer pruning which removes encoder layers from transformer-based networks (Sajjad et al., 2020) and token pruning which removes redundant tokens before feeding them into the next encoder layer (Goyal et al., 2020). Other popular strategies to achieve speedups are neural architecture search, which we can use to find the best neural architecture for the CLIR task, and quantization, where we use less number of bits to encode model parameters.

Bibliography

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6):734–749, 2005.

Qingyao Ai, Keping Bi, Jiafeng Guo, and W Bruce Croft. Learning a deep listwise context model for ranking refinement. *In Proceedings of SIGIR '18*, 2018.

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. Applying BERT to document retrieval with birch. 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): System Demonstrations*, pages 19–24, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-3004. URL https://aclanthology.org/D19-3004.

BIBLIOGRAPHY

James Allan and Hema Raghavan. Using part-of-speech patterns to reduce query ambiguity. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, 2002.

Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *stat*, 1050:21, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Weinberger. Learning to rank with (a lot of) word features. *Information Retrieval*, 13(3):291–314, June 2010. ISSN 1386-4564.

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, 2020.

BIBLIOGRAPHY

Nicholas J Belkin, Diane Kelly, G Kim, J-Y Kim, H-J Lee, Gheorghe Muresan, M-C Tang, X-J Yuan, and Colleen Cool. Query length in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 205–212, 2003.

Tianchi Bi, Liang Yao, Baosong Yang, Haibo Zhang, Weihua Luo, and Boxing Chen. Constraint translation candidates: A bridge between neural query translation and cross-lingual information retrieval. *arXiv preprint arXiv:2010.13658*, 2020.

Hamed Bonab, James Allan, and Ramesh Sitaraman. Simulating clir translation resource scarcity using high-resource languages. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 129–136, 2019.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. Learning to rank using gradient de-

scent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pages 89–96, 2005.

Christopher Burges, Krysta Svore, Paul Bennett, Andrzej Pastusiak, and Qiang Wu. Learning to rank using an ensemble of lambda-gradient models. In *Proceedings of the Learning to Rank Challenge*, pages 25–35, 2011.

Christopher J Burges, Robert Ragno, and Quoc V Le. Learning to rank with nonsmooth cost functions. In *Advances in neural information processing systems*, pages 193–200, 2007.

Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. 2010.

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. ACM, 2007.

Olivier Chapelle and Yi Chang. Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge*, pages 1–24, 2011.

Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630. ACM, 2009.

BIBLIOGRAPHY

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

CLEF 2000-2003. *The CLEF Test Suite for the CLEF 2000-2003 Campaigns – Evaluation Package*, 2003. https://catalog.elra.info/en-us/repository/browse/ELRA-E0008/.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

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. *arXiv preprint arXiv:1911.02116*, 2019.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard 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, 2020.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*, 2020.

BIBLIOGRAPHY

Domenico Dato, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. Fast ranking with additive ensembles of oblivious and non-oblivious regression trees. *ACM Transactions on Information Systems (TOIS)*, 35(2):15, 2016.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74. ACM, 2017a. ISBN 978-1-4503-5022-8.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74, 2017b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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://www.aclweb.org/anthology/N19-1423.

Susan T Dumais and Todd A Letsche. Automatic cross-language retrieval using latent semantic indexing. 1997.

Shi Feng, Eric Wallace, and Jordan Boyd-Graber. Misleading failures of partial-input baselines. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Nicola Ferro and Gianmaria Silvello. CLEF 2000-2014: Lessons learnt from ad hoc retrieval. In Paolo Boldi, Raffaele Perego, and Fabrizio Sebastiani, editors, *Proceedings of the 6th Italian Information Retrieval Workshop, Cagliari, Italy, May 25-26, 2015*, volume 1404 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015. URL http://ceur-ws.org/Vol-1404/paper_5.pdf.

Martin Franz, J Scott McCarley, and Salim Roukos. Ad hoc and multilingual information retrieval at ibm. *NIST special publication SP*, pages 157–168, 1999.

Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

BIBLIOGRAPHY

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

Petra Galuščáková, Douglas W. Oard, and Suraj Nair. Cross-language information retrieval, 2021.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pages 3690–3699. PMLR, 2020.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

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*, pages 55–64. ACM, 2016a.

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*, pages 55–64, 2016b.

BIBLIOGRAPHY

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Visualizing and understanding the effectiveness of bert. 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 4134–4143, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997a.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997b.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050, 2014.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2333–2338. ACM, October 2013a.

BIBLIOGRAPHY

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using click-through data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2333–2338. ACM, 2013b.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. PACRR: A position-aware neural ir model for relevance matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1049–1058. Association for Computational Linguistics, September 2017.

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*, pages 279–287, 2018.

Bernard J Jansen, Danielle L Booth, and Amanda Spink. Determining the informational, navigational, and transactional intent of web queries. *Information Processing & Management*, 44(3):1251–1266, 2008.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

BIBLIOGRAPHY

Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1997.

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, 2020.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

K Sparck Jones and Cornelis Joost Van Rijsbergen. Information retrieval test collections. *Journal of documentation*, 1976.

Nikolaos Katris, Richard Sutcliffe, and Theodore Kalamboukis. Using a cross-language information retrieval system based on OHSUMED to evaluate the Moses and KantanMT statistical machine translation systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 368–372, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1057.

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

BIBLIOGRAPHY

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://aclanthology.org/N03-1017.

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, 2018.

Toshitaka Kuwa, Shigehiko Schamoni, and Stefan Riezler. Embedding meta-textual information for improved learning to rank. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5558–5568, 2020.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

Gina-Anne Levow, Douglas W Oard, and Philip Resnik. Dictionary-based techniques for cross-language information retrieval. *Information processing & management*, 41(3):523–547, 2005.

Constantine Lignos, Daniel Cohen, Yen-Chieh Lien, Pratik Mehta, W. Bruce

Croft, and Scott Miller. The challenges of optimizing machine translation for low resource cross-language information 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 3497–3502, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1353. URL https://aclanthology.org/D19-1353.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

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*, pages 1253–1256, 2018.

Jiapeng Liu, Xiao Zhang, Dan Goldwasser, and Xiao Wang. Cross-lingual document retrieval with smooth learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3616–3629, 2020.

Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, March 2009. ISSN 1554-0669. doi: 10.1561/1500000016. URL https://doi.org/10.1561/1500000016.

BIBLIOGRAPHY

Tie-Yan Liu. Learning to rank for information retrieval. 2011.

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, 2019.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

Yuanhua Lv and ChengXiang Zhai. When documents are very long, bm25 fails! In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1103–1104, 2011.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. Cedr: Contextualized embeddings for document ranking. In *SIGIR*, 2019.

Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. Simplified data wrangling with ir_datasets. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2429–2436, 2021.

MATERIAL. *Machine Translation for English Retrieval of Information in Any*

Language (MATERIAL), 2017. https://www.iarpa.gov/index.php/research-programs/material.

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 on Computational Linguistics*, pages 208–214. Association for Computational Linguistics, 1999.

Ryan McDonald, George Brokos, and Ion Androutsopoulos. Deep relevance ranking using enhanced document-query interactions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1849–1860, 2018.

Robert McMaster and Susanna McMaster. A history of twentieth-century american academic cartography. *Cartography and Geographic Information Science*, 29(3):305–321, 2002.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information*

Processing Systems - Volume 2, pages 3111–3119. Curran Associates Inc., 2013b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013c.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299. International World Wide Web Conferences Steering Committee, 2017. ISBN 978-1-4503-4913-0.

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. *arXiv preprint arXiv:2201.08471*, 2022.

Jian-Yun Nie. *Cross-Language Information Retrieval*. Morgan & Claypool Publishers, 2010.

Vassilina Nikoulina, Bogomil Kovachev, Nikolaos Lagos, and Christof Monz. Adaptation of statistical machine translation model for cross-lingual information retrieval in a service context. In *Proceedings of the 13th Conference*

of the European Chapter of the Association for Computational Linguistics, pages 109–119, 2012.

Erik Novak, Luka Bizjak, Dunja Mladenić, and Marko Grobelnik. Why is a document relevant? understanding the relevance scores in cross-lingual document retrieval. *Knowledge-Based Systems*, 244:108545, 2022. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2022.108545. URL https://www.sciencedirect.com/science/article/pii/S0950705122002416.

Douglas W Oard. A comparative study of query and document translation for cross-language information retrieval. In *Conference of the Association for Machine Translation in the Americas*, pages 472–483. Springer, 1998.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. A study of match pyramid models on ad-hoc retrieval. In *Neu-IR'16 SIGIR Workshop on Neural Information Retrieval*, 2016.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. Deeprank: A new deep architecture for relevance ranking in information

retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 257–266. ACM, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014a. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014b.

Ari Pirkola, Turid Hedlund, Heikki Keskustalo, and Kalervo Järvelin. Dictionary-based cross-language information retrieval: Problems, methods, and research findings. *Information retrieval*, 4(3):209–230, 2001.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, 2018.

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*, pages 275–281, 1998.

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

Tao Qin and Tie-Yan Liu. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597, 2013. URL http://arxiv.org/abs/1306.2597.

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, 2018.

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance frame-

work: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

David Graff Roukos, Salim and Dan Melamed. Hansard french/english ldc95t20. *Philadelphia: Linguistic Data Consortium*, 1995.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *arXiv preprint arXiv:2004.03844*, 2020.

Shadi Saleh and Pavel Pecina. Document translation vs. query translation for cross-lingual information retrieval in the medical domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6849–6860, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.613. URL https://aclanthology.org/2020.acl-main.613.

Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

Shota Sasaki, Shuo Sun, Shigehiko Schamoni, Kevin Duh, and Kentaro Inui. Cross-lingual learning-to-rank with shared representations. In *Proceedings*

of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 458–463, 2018.

Jacques Savoy. Report on clef-2003 multilingual tracks. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 64–73. Springer, 2003.

Jacques Savoy and Martin Braschler. *Lessons Learnt from Experiments on the Ad Hoc Multilingual Test Collections at CLEF*, pages 177–200. Springer International Publishing, Cham, 2019. ISBN 978-3-030-22948-1. doi: 10.1007/978-3-030-22948-1_7. URL https://doi.org/10.1007/978-3-030-22948-1_7.

Shigehiko Schamoni, Felix Hieber, Artem Sokolov, and Stefan Riezler. Learning translational and knowledge-based similarities from relevance rankings for cross-language retrieval. In *Proceedings of the 52 Annual Meeting of the Association for Computational Linguistics*, 2014a.

Shigehiko Schamoni, Felix Hieber, Artem Sokolov, and Stefan Riezler. Learning translational and knowledge-based similarities from relevance rankings for cross-language retrieval. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 488–494, 2014b.

BIBLIOGRAPHY

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*, 2019.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016a. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://aclanthology.org/P16-1009.

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 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information

retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110. ACM, 2014. ISBN 978-1-4503-2598-1.

Artem Sokolov, Laura Jehl, Felix Hieber, and Stefan Riezler. Boosting cross-language retrieval by learning bilingual phrase associations from relevance rankings. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1688–1699. Association for Computational Linguistics, October 2013.

Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321, 1999.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, January 2014. ISSN 1532-4435.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

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, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.340. URL https://aclanthology.org/2020.emnlp-main.340.

Shuo Sun, Ahmed El-Kishky, Vishrav Chaudhary, James Cross, Lucia Specia, and Francisco Guzmán. Classification-based quality estimation: Small and efficient models for real-world applications. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5865–5875, 2021.

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. ACM, 2008.

Jörg Tiedemann. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer, 2012.

Ming-Feng Tsai, Yu-Ting Wang, and Hsin-Hsi Chen. A study of learning a merge model for multilingual information retrieval. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you

need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

Ellen M Voorhees. The philosophy of information retrieval evaluation. In *Workshop of the cross-language evaluation forum for european languages*, pages 355–370. Springer, 2001.

Ellen M Voorhees. The trec robust retrieval track. In *ACM SIGIR Forum*, volume 39, pages 11–20. ACM New York, NY, USA, 2005.

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*, pages 363–372, 2015.

Runchuan Wang, Zhao Zhang, Fuzhen Zhuang, Dehong Gao, Yi Wei, and Qing He. Adversarial domain adaptation for cross-lingual information retrieval with multilingual bert. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3498–3502, 2021.

Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. 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 833–844, 2019.

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*, pages 55–64. ACM, 2017. ISBN 978-1-4503-5022-8.

Linlong Xu, Baosong Yang, Xiaoyu Lv, Tianchi Bi, Dayiheng Liu, and Haibo Zhang. Leveraging advantages of interactive and non-interactive models for vector-based cross-lingual information retrieval, 2021. URL https://arxiv.org/abs/2111.01992.

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. *arXiv e-prints*, pages arXiv–2204, 2022.

Liang Yao, Baosong Yang, Haibo Zhang, Weihua Luo, and Boxing Chen. Exploiting neural query translation into cross lingual information retrieval. *arXiv preprint arXiv:2010.13659*, 2020.

Mahsa Yarmohammadi, Xutai Ma, Sorami Hisamoto, Muhammad Rahman, Yiming Wang, Hainan Xu, Daniel Povey, Philipp Koehn, and Kevin Duh. Robust document representations for cross-lingual information retrieval in low-resource settings. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 12–20, Dublin, Ireland, August 2019. Eu-

ropean Association for Machine Translation. URL https://www.aclweb.org/anthology/W19-6602.

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*, pages 1637–1640, 2020.

Ilya Zavorin, Aric Bills, Cassian Corey, Michelle Morrison, Audrey Tong, and Richard Tong. Corpora for cross-language information retrieval in six less-resourced languages. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 7–13, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-55-9. URL https://www.aclweb.org/anthology/2020.clssts-1.2.

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.

Chengxiang Zhai and John Lafferty. Model-based feedback in the language

modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, 2001.

Fuwei Zhang, Zhao Zhang, Xiang Ao, Dehong Gao, Fuzhen Zhuang, Yi Wei, and Qing He. Mind the gap: Cross-lingual information retrieval with hierarchical knowledge enhancement. *arXiv preprint arXiv:2112.13510*, 2021.

Hang Zhang and Liling Tan. Textual representations for crosslingual information retrieval. In *Proceedings of The 4th Workshop on e-Commerce and NLP*, pages 116–122, 2021.

Rui Zhang, Caitlin Westerfield, Sungrok Shim, Garrett Bingham, Alexander Richard 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, 2019.

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 *57th Annual Meeting of the Association for Computational Linguistics, ACL 2019*, pages 3173–3179. Association for Computational Linguistics (ACL), 2020.

BIBLIOGRAPHY

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, 2019.

Dong Zhou, Mark Truran, Tim Brailsford, Vincent Wade, and Helen Ashman. Translation techniques in cross-language information retrieval. *ACM Computing Surveys (CSUR)*, 45(1):1–44, 2012.