# Resource-Lean Transfer Methods for Cross-Lingual Information Retrieval

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Robert Manfred Litschko
aus Herrenberg

Mannheim, 2024

| | |
|---|---|
| Dekan: | Prof. Dr. Claus Hertling, Universität Mannheim |
| Referent: | Prof. Dr. Goran Glavaš, Universität Würzburg |
| Korreferent: | Prof. Dr. Simone Paolo Ponzetto, Universität Mannheim |
| Korreferent: | Prof. Dr. Alexander Fraser, Technische Universität München |

Tag der mündlichen Prüfung: 11. Juli 2024

# Acknowledgments

This thesis would not have been possible without the support of all the people who accompanied me during my Ph.D. journey.

First and foremost, I would like to thank my primary supervisor Prof. Goran Glavaš. Thank you for being not only a dedicated supervisor, but also a great mentor who I could always trust and talk to. I am deeply grateful that I had the opportunity to do my Ph.D. under your guidance. I would also like to thank my secondary supervisor Prof. Simone Paolo Ponzetto for his constant encouragement and support. I truly enjoyed being part of the DWS NLP group and all the group activities including our Friday lunch outings at Tomate, the WÜRPAL Symposium and the retreat in Annweiler, just to name a few. I would also like to thank Prof. Laura Dietz for the wonderful time in New Hampshire, I am grateful for the invaluable experience and lasting memories I made during my research visit. Additionally, I would like to thank Prof. Alexander Fraser for his time and involvement as a member of the examination committee.

I also want to thank all my academic collaborators with whom I have worked together in the past. Here, I would like to especially thank Dr. Ivan Vulić for the many fruitful discussions and collaborations we had, and Prof. Barbara Plank for the support that allowed me to continue following my passion and pursue new research directions. A large part of my Ph.D. experience has also been shaped by the people around me. I am lucky to have been surrounded by awesome friends and colleagues who I would like to thank, in no particular order: Tommaso Green, Chia-Chien Hung, Fabian Schmidt, Max Müller-Eberstein, Leon Weber, Diego Frasinelli and everyone at the DWS NLP group and MaiNLP lab. Thank you for all the coffee chats, research discussions and good times we had together.

Zum Schluss möchte ich mich bei meinen Eltern Elisabeth und Manfred, sowie bei Albert und bei meinen Geschwistern bedanken, das sie mir immer den Rücken gestärkt und an mich geglaubt haben. Last but not least, I cannot put into words how much I want to thank you, Reme, for your love, patience and moral support throughout all these years, which especially kept me going in difficult times.

*Serendipity.*

# Abstract

Cross-Lingual Information Retrieval (CLIR) is the task of finding relevant documents written in a language different from the query language. Neural machine translation systems and CLIR models based on supervised machine learning (deep learning) are resource-hungry approaches requiring large amounts of training data, which is expensive to obtain and therefore does not scale well to a large number of languages. In this thesis, we study methods for transferring retrieval models across languages in a resource-lean way. The overarching goal is to build effective CLIR systems for languages for which we do not have access to large-scale training data. On a high level, our contributions fall into three areas.

*Unsupervised learning of CLIR models.* In the first part, we propose two fully unsupervised neural CLIR approaches for which no relevance annotations are required. In the representation-based approach, we encode queries and documents into independent semantic vector representations and use vector space similarity measures to calculate document relevance scores. Here, we obtain aligned query and document representations from static cross-lingual word embeddings (CLWEs) and contextual representations produced by multilingual text encoders. In the term-by-term query translation approach, we translate query terms by replacing their occurrences with their cross-lingual nearest neighbors found in CLWE spaces, effectively casting CLIR into a noisy variant of monolingual IR (MoIR). We conduct a large-scale evaluation and, surprisingly, find that off-the-shelf multilingual text encoders fall behind CLWE-based methods in a direct comparison, whereas further specialization for sentence-level semantics yields the best results.

*Resource-lean transfer of CLIR models.* In the second part, we focus on the standard zero-shot cross-lingual transfer (ZS-XLT) setup and use English training data to transfer cross-encoder (CE) reranking models to other languages. We first show that this approach suffers from "monolingual overfitting" where models are biased towards lexical matches between query and document tokens. To regularize this bias, we propose to train CEs on code-switched data instead. Our results show that this consistently improves the ZS-XLT performance for CLIR and maintains stable performance in MoIR. Next, we rely on parameter-efficient transfer methods to disentangle the task of learning-to-rank from learning target language semantics. We show that this modular approach improves upon the standard ZS-XLT approach in a scenario where the training and test data are in different domains.

In the third part, we present on the example task of multilingual dependency

parsing a *proof of concept* for instance-level model selection. Here, we propose cross-lingual transfer with multiple monolingual expert models by using a routing model. Moving away from a single multilingual model bypasses any capacity limits in terms of number of languages ("curse of multilinguality"). Our results pave the way for future work on CLIR involving multiple encoders (e.g. language-family specific encoders).

# Zusammenfassung

Bei sprachübergreifender Informationssuche (engl. Cross-Lingual Information Retrieval; CLIR) geht es darum, relevante Dokumente zu finden, die in einer anderen Sprache als die der Suchabfrage geschrieben sind. Neuronale maschinelle Übersetzung und CLIR-Modelle basierend auf überwachtem maschinellem Lernen (Deep Learning) sind ressourcenintensiv und erfordern große Mengen an Trainingsdaten, deren Beschaffung teuer ist und sich daher nicht gut auf eine große Anzahl von Sprachen ausweiten lässt. In dieser Arbeit untersuchen wir deshalb ressourceneffiziente Methoden, mit denen wir IR-Modelle zwischen verschiedenen Sprachen transferieren können. Das übergeordnete Ziel besteht darin, effektive CLIR-Systeme für Sprachen zu entwickeln, für die wir keinen Zugriff auf umfangreiche Trainingsdaten haben. Der Forschungsbeitrag dieser Arbeit lässt sich in folgende drei Bereiche zusammenfassen.

*Unüberwachtes Lernen von CLIR-Modellen.* Im ersten Teil stellen wir zwei unüberwachte Ansätze vor, mit denen wir CLIR-Modelle erhalten, ohne auf Relevanzannotionen zurückzugreifen. Im repräsentationsbasierten Ansatz enkodieren wir Suchabfragen und Dokumente unabhängig voneinander in semantische Vektorepräsentationen und verwenden diese, um mithilfe von Ähnlichkeitsmaßen Relevanzwerte zu berechnen. Für das Enkodieren verwenden wir sprachübergreifende Wortvektoren (engl. Cross-Lingual Word Embeddings; CLWE) und kontextualisierte Repräsentationen, die von mehrsprachigen Textkodierern erstellt werden. Im zweiten Ansatz, Term-für-Term-Abfrageübersetzung, ersetzen wir jedes Abfragewort durch seinen nächsten sprachübergreifenden Nachbarn im CLWE-Raum und überführen dadurch CLIR in ein monolinguales IR (MoIR) Problem. Wir vergleichen unsere Ansätze in einer umfangreichen Studie und stellen überraschenderweise fest, dass mehrsprachige Sprachmodelle schlechter abschneiden als CLWE-basierte Ansätze, wohingegen eine weitere Spezialisierung auf Semantik auf Satzebene die besten Ergebnisse liefert.

*Ressourceneffizienter Transfer von CLIR-Modellen.* Im zweiten Teil konzentrieren wir uns auf den Standardansatz für sprachübergreifenden Zero-Shot-Transfer (engl. Zero-Shot Cross-lingual Transfer; ZS-XLT) und verwenden ausschließlich englische Trainingsdaten, um Cross-Encoder (CE) Modelle in andere Sprachen zu transferieren. Wir zeigen zunächst auf, dass dieser Ansatz an einer "monolingualen Überanpassung" leidet, bei der Modelle zu sehr auf lexikalische Übereinstimmungen zwischen Abfrage- und Dokument-Tokens ausgerichtet sind. Um diesen Bias

zu regulieren, schlagen wir vor, CE-Modelle stattdessen auf durch Code-Switching manipulierte Daten zu trainieren. Unsere Ergebnisse zeigen, dass wir damit deren ZS-XLT-Leistung für CLIR konsistent verbessern, ohne dabei die Ergebnisse in MoIR zu verschlechtern. Als Nächstes verwenden wir parametereffiziente Transfermethoden, um die Aufgabe des Erlernens von Relevanzmerkmalen vom Erlernen der Zielsprachensemantik zu entkoppeln. Wir zeigen, dass dieser modulare Ansatz besser als der Standard-ZS-XLT-Ansatz abschneidet, wenn die Trainings- und Testdaten in unterschiedlichen Domänen vorliegen.

Im dritten Teil präsentieren wir, am Beispiel von multilingualem Dependenzparsing, eine Machbarkeitsstudie zur Modellselektion auf Instanzebene. Hierbei lernen wir ein Modell, das darauf spezialisiert ist, einzelne Instanzen an einen oder mehrere monolinguale Expertenmodelle weiterzuleiten. Dabei umgehen wir mögliche Kapazitätsgrenzen hinsichtlich der Anzahl der unterstützen Sprachen, denen multilinguale Sprachmodelle ausgesetzt sind (sog. “curse of multilinguality”). Unsere Ergebnisse ebnen den Weg für zukünftige Arbeiten an CLIR mit mehreren Textkodierern, die zum Beispiel auf Daten von verschiedenen Sprachfamilien trainiert wurden.

# Contents

# List of Publications

The work presented in this thesis (including text, figures, and tables) has been previously published at top-tier international conferences and workshops. In each chapter, we reference to the publications upon which it is based. Below, we list all publications in reversed chronological order, each publication venue is annotated according to its CORE Conference Ranking (A*, A, and B).

**Robert Litschko**, Ekaterina Artemova, and Barbara Plank. Boosting Zero-shot Cross-lingual Retrieval by Training on Artificially Code-Switched Data. In *Findings of the Association for Computational Linguistics: ACL 2023* (Findings of ACL 2023, A*), pages 3096–3108, Toronto, Canada.

**Robert Litschko**, Ivan Vulić, and Goran Glavaš. Parameter-Efficient Neural Reranking for Cross-Lingual and Multilingual Retrieval. In *Proceedings of the 29th International Conference on Computational Linguistics* (COLING 2022, A), pages 1071–1082, Gyeongju, Republic of Korea.

**Robert Litschko**, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. On Cross-Lingual Retrieval with Multilingual Text Encoders. *Information Retrieval Journal* 25.2 (2022), pages 149–183.

**Robert Litschko**, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. Evaluating Multilingual Text Encoders for Unsupervised Cross-Lingual Retrieval. In *Advances in Information Retrieval: 43rd European Conference on IR Research* (ECIR 2021, B), pages 342–358, Lucca, Italy (Online).

**Robert Litschko**, Ivan Vulić, Željko Agić, and Goran Glavaš. 2020. Towards Instance-Level Parser Selection for Cross-Lingual Transfer of Dependency Parsers. In *Proceedings of the 28th International Conference on Computational Linguistics* (COLING 2020, A), pages 3886–3898, Barcelona, Spain (Online).

**Robert Litschko**, Goran Glavaš, Ivan Vulić, and Laura Dietz. Evaluating Resource-Lean Cross-Lingual Embedding Models in Unsupervised Retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2019, A*), pages 1109–1112, Paris, France.

**Robert Litschko**, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. Unsupervised Cross-Lingual Information Retrieval Using Monolingual Data Only. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2018, A*), pages 1253–1256, Ann Arbor, Michigan (USA).

The publication listed above form the basis of this dissertation and are used in Sections 2.2 and 2.3.3 and in Chapters 4 to 9. In addition, the author also contributed to the following publications, which we list in reverse chronological order.

**Robert Litschko***, Max Müller-Eberstein*, Rob van der Goot, Leon Weber, and Barbara Plank. Establishing Trustworthiness: Rethinking Tasks and Model Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2023, A*), 193–203, Singapore.

Gretel Liz De la Peña Sarracén, Paolo Rosso, **Robert Litschko**, Goran Glavaš, and Simone Paolo Ponzetto. Vicinal Risk Minimization for Few-Shot Cross-lingual Transfer in Abusive Language Detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2023, A*), pages 4069–4085, Singapore.

Onur Galoğlu, **Robert Litschko**, and Goran Glavaš. A General-Purpose Multilingual Document Encoder. In *Proceedings of the 3rd Workshop on Multilingual Representation Learning* (MRL 2023), 37–49, Singapore.

Chia-Chien Hung, Tommaso Green, **Robert Litschko**, Tornike Tsereteli, Sotaro Takeshita, Marco Bombieri, Goran Glavaš, and Simone Paolo Ponzetto. ZusammenQA: Data Augmentation with Specialized Models for Cross-lingual Open-retrieval Question Answering System. In *Proceedings of the Workshop on Multilingual Information Access* (MIA 2022), pages 77–90, Seattle, Washington (USA).

Ivan Vulić, Edoardo Maria Ponti, **Robert Litschko**, Goran Glavaš, and Anna Korhonen. 2020. Probing Pretrained Language Models for Lexical Semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2020, A*), pages 7222–7240, Online.

Goran Glavaš, **Robert Litschko**, Sebastian Ruder, and Ivan Vulić. 2019. How to (Properly) Evaluate Cross-Lingual Word Embeddings: On Strong Baselines, Comparative Analyses, and Some Misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (ACL 2019, A*), pages 710–721, Florence, Italy.

*Both authors contributed equally.

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In this chapter, we first motivate the need for cross-lingual information retrieval and resource-lean transfer of retrieval models between languages (Section 1.1). Next, we summarize our main contributions (Section 1.2) and conclude with an outline of the remainder of this thesis (Section 1.3).

## 1.1 Motivation

Information Retrieval (IR) is the task of satisfying a users' *information need* (i.e., a users' desire to locate information about a topic) expressed in a *query*, by searching for relevant content in a large collection of unstructured material and organizing the collection in a way such that it can be searched in an efficient and effective way (Schütze et al., 2008). The most common branch of IR is text search, where unstructured material refers to large collections of text documents, also known as *document collection* or *corpus*. In ranked retrieval, IR systems present a list of documents ranked according to their estimated relevance. IR can be viewed as a natural language processing (NLP) task because of the unstructured nature of queries and documents. This distinguishes it from traditional database systems which use a structured query language (SQL) to retrieve and process data, where the structure is determined by pre-defined schemata.

IR systems and search engines are ubiquitous in our everyday life, journalists seek information from diverse sources, researchers conduct literature reviews with academic search engines, software engineers search for code on StackOverflow, jobseekers browse vacancies on job portals, and lawyers use legal search engines to search for case files and patents. Sometimes we interact with IR systems not only in an explicit but also in an implicit way, e.g. when automated fact checking tools verify claims during political debates, or when we interact with virtual assistants. Most recently, the rise of generative large language models (LLMs) such as ChatGPT and GPT-4 (OpenAI, 2022, 2023) or Llama-3 (Meta, 2024) have transformed the way we view language processing applications. LLMs are now used as general-purpose agents for a myriad of knowledge-intensive NLP tasks. This

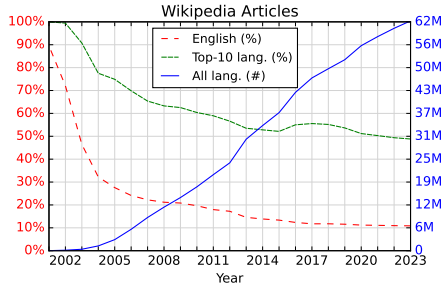


**Figure 1.1:** Number of articles on Wikipedia, fraction of English articles, and fraction of articles of the ten largest Wikipedias each year.[1]

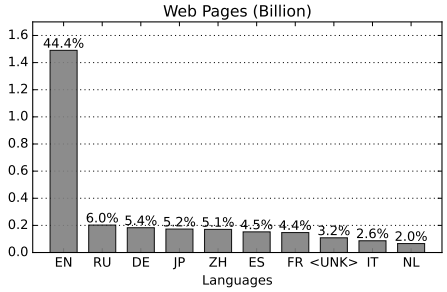


**Figure 1.2:** Number of web pages contained in the Common Crawl corpus in Dec 2023 (CC-MAIN-2023-50), and the share of each language.

raises the question of whether we still need retrieval systems to search for relevant information when we could also prompt LLMs to directly generate answers for our questions. Previous work shows that LLMs store factual knowledge in their model parameters (Mallen et al., 2022; Meng et al., 2022). However, due to their black-box nature, it is unclear how the parametric knowledge is combined to generate the model output (Litschko et al., 2023). As a result, users cannot reliably anticipate under which conditions LLMs are prone to generate factually incorrect statements, which are also known as hallucinations (Dziri et al., 2022; Mallen et al., 2022). For example, Mallen et al. (2022); Kandpal et al. (2023) find that their ability in answering factual questions about entities relates to their popularity, i.e., how well they are represented in pre-training corpora. A recent line of work on retrieval-augmented generation models (RAG) (Lee et al., 2019; Lewis et al., 2020b) shows that grounding the output generation in externally retrieved, i.e. non-parametric knowledge mitigates the hallucination problem in LLMs (Mallen et al., 2022; Semnani et al., 2023; Ram et al., 2023, *inter alia*). This goes to show that information retrieval is not only crucial for satisfying human information needs, but also for LLMs to be more accurate and trustworthy (Litschko et al., 2023).

The web is arguably the largest document collection available for retrieval. The common crawl corpus[2] is an openly available collection of web crawl data. A recent version (CC-MAIN-2023-50) contains a total of 3.35 billion web pages. At the same time, Wikipedia being one of the most visited websites is currently hosting ~62 million articles (see Appendix A). Over time, the web has become increasingly linguistically diverse. For example, as shown in Figure 1.1, in December 2001 about 17K out of 19K Wikipedia articles were written in English (89%), and twenty-two years later, its relative share has shrunk to 10.9% (6.8M articles). Today, Wikipedia supports over 300 languages, and combining the ten largest Wikipedia language editions still amounts to less than 50% of all articles.

[1] Figure 1.1 is adapted from (Wikimedia Commons, 2020).

[2] https://commoncrawl.org/

This means that more than half of all Wikipedia articles belong to the long tail of articles written in any of the other languages. Figure 1.2 shows the language distribution of all web pages contained in the Common Crawl corpus.[3] Here, too, we can see that for less than half of all web pages (44%) English has been classified to be the primary language, followed by Russian (RU), German (DE), Japanese (JP), Chinese (ZH), Spanish (ES), French (FR), Italian (IT) and Dutch (NL). Notably, these languages were identified with Google's Compact Language Detector 2,[4] which supports 83 languages. The fact that many web pages could not be categorized into any of those languages (<UNK>) indicates that a substantial portion of the web is written in an underrepresented language. The proliferation of multilingual content on the web motivates the development of IR systems capable of retrieving relevant documents across language boundaries, as discussed next.

### 1.1.1 Cross-Lingual Information Retrieval

Sometimes relevant information is written in a language different from the query language in which the information need is expressed. To access this information, we need cross-lingual information retrieval (CLIR) systems. CLIR systems go beyond the users' perspective because retrieving external information is also an integral part of many other natural language processing (NLP) applications. These include, e.g., cross-lingual plagiarism detection (Potthast et al., 2011; Glavaš et al., 2018) and combating misinformation with cross-lingual fact checking (Gupta and Srikumar, 2021; Huang et al., 2022). Also, in question answering applications, the answer for a given question might be present in a different language (Contractor et al., 2010; Asai et al., 2021b; De Bruyn et al., 2021). We will now discuss how improving information access across language boundaries therefore not only improves user experience and contributes towards digital inclusion, but also how it improves the usefulness of digital assistants that rely on search.

**Digital Language Divide.** In the previous section, we discussed how the web has become increasingly multilingual. While one half is dominated by English and a few other high-resource languages, the other half is made up of a long tail of other languages. Similar to how English is the dominant language on the web, NLP research has long been dominated by English (Mielke, 2016; Bender, 2019; Ruder, 2020; Søgaard, 2022). In fact, Søgaard (2022) found that about "two thirds of NLP research at top venues is devoted exclusively to developing technology for speakers of English". According to the Ethnologue there exists over 7,000 languages (Eberhard et al., 2022), and most languages have very limited or no resources available to train NLP and IR models (Joshi et al., 2020a; Wang et al., 2022a). The disparity of available resources creates a *digital language divide* (Young, 2015) between speakers of few high-resource languages and speakers whose native language does

[3] https://commoncrawl.github.io/cc-crawl-statistics/plots/languages.csv
[4] https://github.com/CLD2Owners/cld2

not have a large digital footprint. While the first group benefits from access to mature information systems, the second group is left behind. Graham and Zook (2013) exemplify this by showing how people searching for restaurants on Google Maps get different search results depending on the language in which the query is formulated. In general, one might not only get different search results, with varying degrees of quality and diversity, but also a different number of search results. That is, the choice of language directly determines the experience on the web (Young, 2015). In the context of web search, speakers of underrepresented groups might be unable to interact with search engines by naturally expressing their information needs in their native language, e.g., because it leads to substantially worse search results or because it is not supported altogether. These examples show why one might prefer to use a language different from their native language to avoid a worse experience with information systems. In fact, Kornai (2013) predicted most languages in existence will gain no digital presence and, consequently, the dominance of some languages will cause gradual extinction of other languages.

The digital language divide undermines the freedom of language choice and cultivation of linguistic diversity, which is deeply embedded in our societal values. For example, the United Nations declare in the Universal Declaration of Human Rights that "everyone has the right to freedom of opinions and expression; this right includes freedom [...] to seek, receive and impart information and ideas through any media and regardless of frontiers." (Article 19). Here, the right to seek information regardless of frontiers includes information access across language boundaries. And the European Union states in the Charter of Fundamental Right of the European Union: "The Union shall respect cultural, religious and linguistic diversity" (Article 22). Thus, to make information truly accessible, IR systems need to be able to bridge the gap between the languages in which information needs are expressed and languages in which documents are written, motivating the development of cross-lingual information retrieval (CLIR) systems. CLIR enables speakers of underrepresented languages to access information relevant to their information needs and thereby promotes the linguistic diversity on the web, i.e., CLIR contributes towards narrowing the digital language divide.

**Information Asymmetry.** In addition to democratizing language technologies and information access to a larger group of people by narrowing the digital language divide, CLIR also benefits speakers of high-resource languages. CLIR specifically allows users to access information that is written by speakers with diverse cultural backgrounds and perspectives. *Information asymmetry* describes the observation that content written in different languages not only covers different topics but also different perspectives on the same topics (Roy et al., 2022). We will now illustrate specific examples of information asymmetry on the web and focus on cultural biases embedded in encyclopedic knowledge. In a recent study, Miquel-Ribé and Laniado (2019) analyze the largest 40 Wikipedia language editions and find that, on average, 25% of all content refer to local and cultural entities such as places,

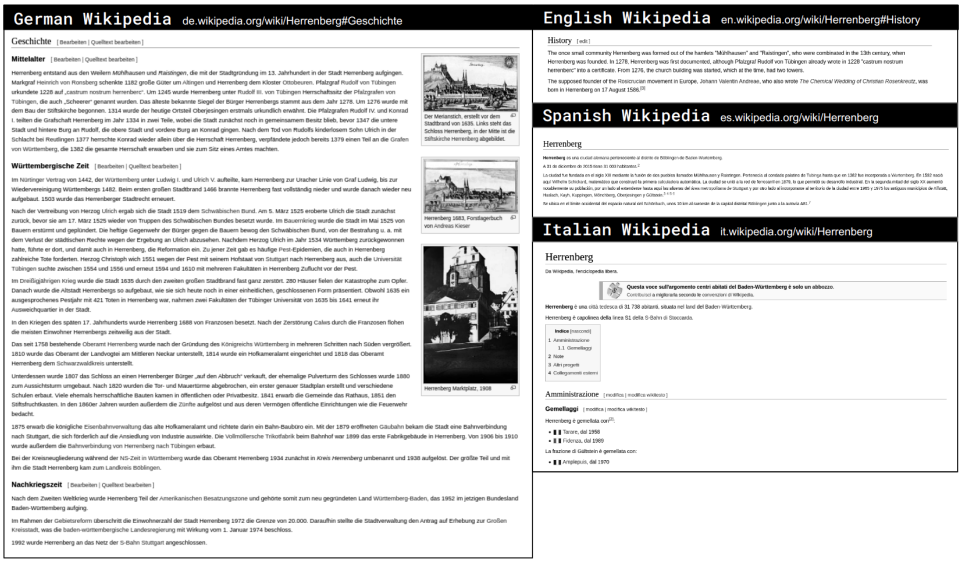


**Figure 1.3:** Information Asymmetry on Wikipedia. Searching for "Herrenberg" on the Wikipedia domain can return varying amounts of information on its history.

traditions, languages, agriculture and biographies. The authors further find that Wikipedia's language editions are limited in their cultural diversity (culture gap), i.e., most of the local cultural content cannot be found on other Wikipedia language editions. Even if two Wikipedia articles in different languages discuss the same topic, their nature and amount of content can vary. Kolbitsch and Maurer (2006); Callahan and Herring (2011) examine Wikipedia biographies about locally famous people, which they refer to as "local heroes", to understand whether those tend to be longer and more favorable compared to their version in a different language. Specifically, Kolbitsch and Maurer (2006) highlight an example where the English article about the American chess player Paul Morphy had significantly more words than the German version. This analysis was later extended by Callahan and Herring (2011), who compare biographies of fifteen English and Polish local heroes from different domains (sports, politics, music, etc.). Notably, their results highlight that articles about the same entity in different languages have unbalanced coverage of controversial information such as, e.g., extramarital affairs, problems with law enforcement and career controversies. Unbalanced coverage is not limited to biographies. In Figure 1.3, we provide a location-related example of information asymmetry. Suppose we search for the term "Herrenberg" (small town in Germany) to find information about its history. On the German Wikipedia version, we get access to multiple paragraphs of relevant information, whereas the history section of the English article contains only a single paragraph. The Spanish article contains some information in its lead paragraph, while the Italian article is missing historical information altogether. Most of Wikipedia language versions contain no article about the entity at all.

Information asymmetry origins from the interests and knowledge of the people who author and edit texts, which in turn is influenced, among others, by cultural biases and geographical factors (Callahan and Herring, 2011; Roy et al., 2022; Samir et al., 2024). It negatively impacts web search because it leads to search results that are potentially biased or fail to satisfy a users' information need due to missing information. It can also pose a challenge for high recall search applications such as patent retrieval (Piroi et al., 2011; Shalaby and Zadrozny, 2019), surveying existing methods on a topic (writing survey papers) (Wang et al., 2024), or evidence retrieval for automatically fact-checking claims (Guo et al., 2022), where it is crucial to gather as much relevant information as possible, irrespective of language boundaries. Finally, the adverse effects of information asymmetry also propagate to retrieval-augmented LLMs, because their ability to respond to a knowledge-related question relies on the information that is accessible (Mallen et al., 2023). If a model cannot access detailed information about an entity, it will not be able to answer specific questions about it.

### 1.1.2 The Case for Resource-Lean Transfer

Modern information retrieval systems rely on supervised machine learning, in particular deep learning (Goodfellow et al., 2016), to train relevance prediction models (see Section 3.4). The limited availability of resources (i.e., training data) and the large number of languages in existence present unique challenges for developing CLIR systems (Section 3.2). We therefore focus on resource-lean transfer methods for CLIR. To this end, we distinguish between three types of resources:

**R1. Language Resources** refer to monolingual data (easy to obtain) and parallel data (hard to obtain). Monolingual data allow language models to learn syntactic and semantic knowledge during pre-training (Chapter 2). Parallel data helps to align this knowledge across language boundaries (see Sections 2.2 and 2.3.3). In CLIR, parallel data can be used to specialize ranking models for interlingual semantics or to train machine translation models.

**R2. IR Resources** refer to downstream task training data (i.e., monolingual task supervision) and facilitates training supervised retrieval/ranking models. During training, IR models acquire task-specific language understanding capabilities required to interpret/represent information needs (queries) and match those against relevant documents.

**R3. CLIR Resources** refer to cross-lingual downstream task supervision (hard to obtain). In theory, training ranking models on CLIR training data facilitates learning both (R1) and (R2) in an end-to-end fashion.[5]

[5] In practice, obtaining high quality training data is important to train robust CLIR models. Similar to other language understanding tasks, CLIR models trained on low quality data can potentially suffer from learning heuristics and "shortcuts" (McCoy et al., 2019). In Chapter 7, we investigate one particular heuristic, which we refer to as "monolingual overfitting", where a model relies (too much) on overlapping keywords between queries and documents.

To motivate resource-lean transfer, we further distinguish languages by the available amount of resources. There are different ways how languages can be categorized according to their digital presence. One way is to distinguish between head and tail languages (Siddhant et al., 2022; Imani et al., 2023), where the former refers to the top hundred languages with the largest Wikipedia language versions and the latter to all other languages. Another commonly adopted approach is proposed in (Joshi et al., 2020b), who categorize languages into six classes according to the amount of labeled and unlabeled data that exist. Yong et al. (2023) group these into low-, mid- and high-resource languages. We adopt a simplified version of this notion and differentiate only between *low-resource languages* and *high-resource languages* (see Section 3.3 and Appendix B).

In Figure 1.4 we present a taxonomy of cross-lingual transfer methods for cross-lingual retrieval. As discussed below, sourcing large amounts of high-quality training data (**R1-3**) for a quadratic combination of up to 7,000 languages would be extremely costly (i.e., resource-intensive) and infeasible in practice. In this thesis, we therefore focus on resource-lean methods for transferring CLIR models to different languages. We first investigate methods for unsupervised retrieval (Chapter 4 and 5). Here, our models do not require any task-level supervision (**R2**), and only limited language resources in the form of bilingual dictionaries and monolingual corpora (**R1**). We then focus on cross-lingual transfer based on two types of limited task supervision (**R3**). In the first one, we investigate few-shot CLIR, which refers to a scenario where we assume a limited annotation budget to obtain relevance annotations for a few queries (Chapter 5). In the second one, we transfer models trained on a task that is similar to CLIR. Here, the task supervision is not limited in quantity but in quality (Chapter 6). Finally, in Chapters 6 to 9 we investigate transfer methods that rely only on monolingual task supervision (**R2**). In summary, we ground our definition of *resource-lean transfer methods for CLIR* on the availability of existing resources (**R1-3**) and include any method that enables (or improves) transferring IR models across languages without the need of collecting large-scale human-annotated data (i.e., direct supervision or parallel data). We now discuss resource-lean and resource-intensive transfer in more detail.

**Resource-Intensive Transfer Methods.** In principle, machine translation (MT) can be used to transform any cross-lingual IR task into a monolingual IR task. One can train multilingual models on translated data (*translate train*) or translate queries to the document language at inference time (*translate test*) (Artetxe et al., 2023). The advantage of MT is that it allows us to reuse large amounts of (typically English) IR resources to obtain supervised retrieval models for different target languages. However, training bilingual and multilingual MT systems requires massive amounts of parallel and multi-way parallel data (Fan et al., 2021a; Kudugunta et al., 2023). Furthermore, storing translations for many languages would lead to excessive space requirements. Furthermore, the translate-test approach increases query latency by the inference time required for translation, slowing down retrieval.

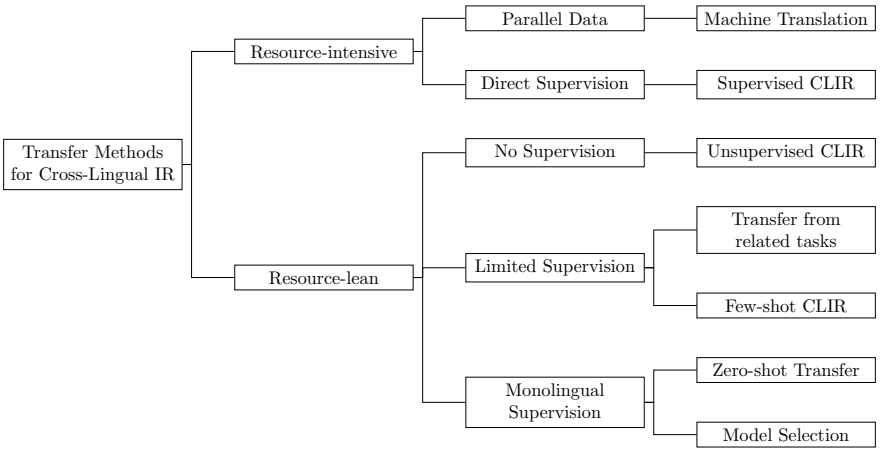


**Figure 1.4:** Taxonomy of cross-lingual transfer methods for CLIR.

Alternatively, a straightforward way to obtain ranking models in different languages is to collect training data (i.e. direct supervision) in order to specialize (or train from scratch) supervised ranking models for a given query-document language pair. However, modern IR models based on neural networks (Pang et al., 2016; Nogueira et al., 2019b; Khattab and Zaharia, 2020, *inter alia*) contain millions of parameters and thus require access to large amounts of training data. Obtaining high quality relevance labels from manual annotations (i.e. human-annotated data) is a labor-intensive task and therefore too expensive to scale up to many language pairs. Exhaustively annotating all documents in a corpus has already been shown to be impractical for monolingual retrieval tasks. For example, Voorhees (2001) illustrates how annotating relevance for a single information need (query) and all documents in a moderately sized document collection consisting of 800k documents, with a judgment rate of 30 seconds, would take over nine months. A commonly adopted practice when developing standard IR test collections is to spend a limited annotation budget on a pool of documents, which are obtained from multiple retrieval systems and likely to be relevant (see Section 3.1.1).[6] Another common approach is to obtain synthetic relevance annotations from of inter-language links on Wikipedia and synthetic queries extracted from Wikipedia titles or lead paragraphs (Sasaki et al., 2018; Sun and Duh, 2020; Ogundepo et al., 2022, *inter alia*). However, the quality of synthetic data is typically lower since its distribution differs from the data distribution a model is exposed to at test time. In Section 3.2 we further discuss the quality versus quantity trade-off between obtaining annotations from humans and automatic construction of retrieval benchmarks.

[6]This is still expensive: Faggioli et al. (2023a) show how annotating 86,000 pooled documents for 50 topics (queries) in the TREC-8 Ad hoc track (Voorhees and Harman, 1999) required over 700 assessor hours and cost about $15,000. In comparison, a commonly used dataset to train and evaluate neural retrieval models, MS MARCO (Nguyen et al., 2016), contains 6,980 test queries.

As of today, there exists no benchmark that (i) is large enough to *train and evaluate* neural CLIR models and simultaneously (ii) has a *broad language coverage*. Motivated by this, we focus on *resource-lean transfer methods*, which allow us to obtain cross-lingual retrieval models for different target language pairs without requiring access to direct supervision.

**Resource-Lean Transfer Methods.** In this thesis, we aim to obtain CLIR systems without collecting large amounts of training data. We distinguish between three resource-lean cross-lingual transfer scenarios with different levels of supervision, as shown in Figure 1.4. In the first scenario, we investigate transfer methods that require *no task supervision* (Chapters 4 and 5). These methods rely on multilingual semantic representation spaces and vector space similarity measures (see Chapter 2). Here, we rely on text representation models that are trained on large monolingual corpora in a fully self-supervised fashion, (Mikolov et al., 2013c; Devlin et al., 2019) where labels are derived from the data itself. This scales well to many languages and requires little to no bilingual supervision.

Resource-lean transfer methods with *limited supervision* break down into two branches. In the first branch, the supervision is limited in a qualitative sense where the task on which models are trained, cross-lingual semantic textual similarity (STS) (Hercig and Kral, 2021), only cover some aspects of cross-lingual relevance matching, which, according to Guo et al. (2016), can be characterized by the ability to perform similarity matching, exact lexical matching, understanding query-term importance and identifying local relevance signals in long documents. Here, we investigate how effective cross-lingual STS models perform when they are applied on CLIR (Chapter 5). In the second branch, the task supervision is limited in a quantitative sense, referring to the size of the training data. Sometimes, there is a limited annotation budget that can be spent to collect relevance judgements for a few queries (Few-shot CLIR).[7] This leads to high quality in-domain data, as it is closer to distribution as the data we expect at test time. In this regard, we investigate to what extent we can improve the performance of cross-lingual STS models when fine-tuned on little in-domain data (Section 5.5.5).

Lastly, compared to CLIR it is easier and cheaper to obtain large(r) amounts of IR resources in monolingual setups. This is because annotators do not need to be bilingual, and one can obtain synthetic relevance judgments with unsupervised lexical retrieval models (Sun and Duh, 2020; Ogundepo et al., 2022) or by exploiting links between Wikipedia articles (Schamoni et al., 2014; Sasaki et al., 2018). Also, companies and organizations can use existing user log data to derive relevance judgments (Nguyen et al., 2016). In this thesis, we study how we can use *monolingual supervision* to transfer retrieval models to (and across) different languages. Following the zero-shot cross-lingual transfer approach (Hu et al., 2020;

[7] The notion of few-shot training is usually related to the number of training instances. For example, in the context of NMT, Chen et al. (2020) use few-shot training to refer to a setup with 50 to 500 training instances. We use few-shot CLIR to refer to annotating the rankings of a *few queries*.

Liang et al., 2020), we train ranking models on top of multilingual language models (Devlin et al., 2019; Conneau and Lample, 2019) by using English training data. In the model selection approach, we present a proof of concept (Chapter 9) and use monolingual datasets in different languages to train multiple models. A routing model then forwards inputs to the models with the highest expected performance.

## 1.2 Contributions

Motivated by the resource scarcity (Section 1.1.2) and the goal of making information accessible across language boundaries (Section 1.1.1), the focus of this thesis is the study of *resource-lean cross-lingual transfer methods* and facilitating information search irrespective of language boundaries. This goal entails developing retrieval systems capable of matching queries against documents in the same language (monolingual IR; MoIR), in different languages (multilingual IR; MLIR) and across different languages (cross-lingual IR; CLIR). Contrary to retrieval within language boundaries, as done in MoIR, in CLIR models cannot rely on relevance signals based on exact lexical matches, rendering CLIR a challenging task. Neural representation learning has emerged as a powerful method for representing text in a semantic vector space (Mikolov et al., 2013c; Devlin et al., 2019), which allows us to compute soft matches based on semantic similarity between queries and documents. In this thesis, we focus on resource-lean transfer for semantic CLIR models. We now summarize our contributions, grouped into five topic areas (**C1-5**).

**C1: Unsupervised CLIR with Cross-Lingual Word Embeddings.** In the first part, we propose unsupervised CLIR models that require no relevance annotations. We focus primarily on the bi-encoder paradigm (see Section 3.4), in which queries and documents are independently projected into a latent semantic vector space.

1. *CLIR Models based on CLWEs.* Based on cross-lingual word embedding (CLWE) spaces (Ruder et al., 2019), we propose two CLIR models (see Section 4.2). The first method represent queries and documents by aggregating their constituent CLWEs. The second method uses CLWEs to translate query terms into the document language, followed by lexical retrieval. To the best of our knowledge, we are the first to propose a fully unsupervised neural CLIR pipeline. In a preliminary study, we validate the effectiveness of our approach in a fully unsupervised CLIR setup.

2. *Systematic evaluation of CLWEs on CLIR.* We conduct a systematic evaluation of different resource-lean methods for inducing CLWE spaces and compare their effectiveness on word-, sentence- and document-level CLIR (Section 4.4). We show that the best results can be obtained with a method that only requires a dictionary consisting of 5K word translation pairs.

**C2: Investigating the Role of Contextualization in CLIR.** On the example of two pre-trained language models (PLM), mBERT (Devlin et al., 2019) and XLM (Conneau and Lample, 2019), we investigate multilingual text encoders for CLIR with limited supervision and without any supervision. In Chapter 5, we investigate the impact of contextualization on sentence-level and document-level CLIR.

1. *CLIR Models based on PLMs.* We study three different types of contextualization. First, we encode tokens in isolation without any context (ISO). In the second approach, we encode tokens by their "average over contexts" (AOC) with representations extracted from their occurrences on Wikipedia. In both cases, we construct static word embeddings from mPLMs. The third approach uses mPLMs to encode queries and documents similar to sentence embeddings (SEMB) and dynamically contextualize tokens in-place with their surrounding query or document tokens (Section 5.2).
2. *Static versus in-place context.* We show for document-level CLIR that contextualization (SEMB, AOC) yield gains up to twice as large as the performance obtained with ISO representations (Section 5.5.1). On sentence-level CLIR, we find that in-place context (SEMB) dominates over static contextual representations (ISO, AOC). We further ablate over how many contexts we need to average to obtain high quality AOC representations (Section 5.5.4).
3. *Degree of contextualization.* In addition to using contextualized representations from the output layer, we also investigate the impact of different degrees of contextualization. Here, we evaluate the internal layer representations of mBERT and XLM on CLIR (Section 5.5.4). Our results show that upper layers (high degree of contextualization) work best for document-level CLIR whereas middle layers (moderate degree of contextualization) work best for sentence-level CLIR.
4. *Context outside the maximum sequence length.* Applying multilingual text encoders only on the first $n$ tokens up to the maximum sequence length ignores relevance signals appearing at later positions. Here, we first show that PLMs fall behind CLWEs if we do not account for larger context window sizes (Section 5.5.1). We address this limitation and experiment with localized relevance matching. Our results show that representing documents by multiple local embeddings leads to better CLIR performance (Section 5.5.3).

**C3: CLIR with Limited Supervision.** We empirically evaluate the impact of relying on supervision that is limited in quality and limited in quantity. Specifically, we investigate how well models perform when trained on related tasks and when trained on little in-domain CLIR data.

1. *Sentence-similarity specialized PLMs.* Semantic textual similarity (Hercig and Kral, 2021) and bi-text mining (Artetxe and Schwenk, 2019a; Zweigenbaum et al., 2018) are two closely related tasks. We evaluate multilingual PLMs specialized for sentence-level similarity (Section 5.3.1) and show

that these models can outperform both their vanilla mPLM counterparts and CLWE-based models (Section 5.5.1). Our findings validate that cross-lingual semantic matching is indeed a central part of CLIR and a strong proxy for cross-lingual relevance matching.

2. *Few-shot CLIR.* We use a cross-fold validation setup to simulate a few-shot CLIR scenario in which we have access to relevance annotations for less than sixty queries (Section 5.5.5). We show that further fine-tuning sentence-similarity specialized PLMs using a contrastive loss function consistently yields performance improvements.

**C4: Zero-shot Transfer for Cross-Lingual Re-ranking.** In this part, we utilize available relevance annotations in English to transfer retrieval models to other languages. We follow a model-based and a dataset-based transfer approach. Our results on re-ranking in CLIR show that both approaches can improve upon the standard zero-shot cross-lingual transfer approach with mPLMs.

1. *Domain effects.* In Section 6.4.1, we compare two reranking models trained on datasets that differ both in size and in their domain similarity to the test dataset. In our experiments, we find that the size of the training dataset plays a crucial role in obtaining a good transfer performance. We further show that few-shot fine-tuning (of multilingual sentence encoders) on in-domain CLIR data outperforms zero-shot transfer based on PLMs trained on large-scale out-of-domain MoIR data.

2. *Dataset-based Zero-shot Transfer.* We first show that training zero-shot CLIR models on large monolingual training data is prone to overfitting to lexical features which cannot be exploited at test time, which we refer to as "monolingual overfitting" (see Section 6.4.2 and Chapter 7). We then demonstrate the effectiveness of code-switching training data as a way to regularize monolingual overfitting. The gains are most pronounced when queries and relevant documents have no tokens in common (Section 7.4).

3. *Model-based Zero-shot Transfer.* In Chapter 8, we experiment with two parameter-efficient transfer methods (Pfeiffer et al., 2020; Ansell et al., 2022) that allow us to decouple (i) language specialization from (ii) learning to match queries against relevant documents during training. Our results show that this modular approach outperforms the standard zero-shot cross-lingual transfer in a scenario where the training and test data are in different domains. For the language pairs where we have access to machine translation (MT) models, we show that parameter-efficient transfer can further improve initial rankings obtained from a MT-based lexical retrieval system.

**C5: Expert Model Selection (Proof of Concept).** Multilingual encoders reach capacity limits as we try to pre-train them on an increasing number of languages,

which is also known as the "curse of multilinguality" (Conneau et al., 2020). We present a proof of concept where, instead of transferring a multilingual model, we transfer multiple expert models, thereby bypassing model capacity limitations of a single multilingual model.

1. *Instance-level model selection framework.* In Chapter 9, we present an instance-level model selection framework that consists of (i) independently trained monolingual expert models and (ii) a routing model which predicts for each instance-expert combination the expected performance.
2. *Proof of concept.* We use delexicalized dependency parsing as our proof-of-concept task for which we have many training and test languages in high- and low-resource languages. We train forty-two monolingual expert models and transfer instances from a set of twenty low-resource languages. We find that our approach outperforms two treebank-level model selection baselines. However, there are still large gaps compared to the performance achieved by oracle models, paving the way for future research.

## 1.3 Outline

This thesis is organized into four parts. In *Part I Theoretical Background*, we review fundamental concepts and terminology on representation learning, information retrieval and CLIR experiments. Here, we also describe our experimental protocol (languages, datasets, baselines, and measures) used in our experiments.

In *Part II Resource-lean Transfer of Bi-Encoders*, we focus on the representation-based retrieval paradigm where queries and documents are first independently projected into in a semantic embedding space and then compared with vector space similarity measures to compute relevance scores (Section 3.4). In Chapter 4, we first investigate unsupervised retrieval methods based on static cross-lingual word embeddings, which are induced with minimal cross-lingual supervision (**C1**). In Chapter 5, we study three CLIR models based on multilingual text encoders **(C2.1)** and investigate the importance of context information on CLIR (**C2.2** - **C2.4**). We then compare models based on off-the-shelf mPLMs against (i) models trained on related tasks (i.e., models specialized sentence similarity tasks; **C3.1**) and (ii) models trained on in-domain data in a few-shot setting (**C3.2**).

In *Part III Resource-lean Transfer of Cross-Encoders*, we focus on zero-shot cross-lingual transfer (ZS-XLT) for cross-lingual reranking. Here, we assume the availability of monolingual supervision. We transfer cross-encoder reranking models (see Section 3.4) trained on English queries and documents to a previously unseen language pairs in a zero-shot fashion. In Chapter 6, we first study domain effects that occur when transferring rankers to out-of-domain datasets (**C4.1**). In our experiments, we find that CE models trained on monolingual retrieval (MoIR) data are biased towards lexical matching, which we refer to as "monolingual overfitting". We find that it harms the ZS-XLT transfer performance for CLIR be-

cause, unlike MoIR, queries and documents are written in different vocabularies. In Chapter 7, we propose to regularize monolingual overfitting by training on code-switched data instead (**C4.3**). Finally, in Chapter 8, we focus on parameter-efficient ZS-XLT for CLIR. Here, we follow a modular approach and disentangle learning target language semantics from learning ranking features.

In *Part IV Resource-lean Transfer with Multiple Encoders*, we investigate the feasibility (proof of concept) using multiple models in a scenario where we have access to monolingual training data in multiple languages. More precisely, we use multiple monolingual data to train multiple monolingual expert models, instead of training a single multilingual model (Chapter 9). We present an instance-based model selection framework that routes input instances to the expert model that is expected to perform best (**C5.1**). We evaluate this approach on the proof-of-concept task of delexicalized dependency parsing (**C5.2**).

In Chapter 10, we summarize our main findings and present possible avenues for future work. At the beginning of each chapter, we link its content to our taxonomy on cross-lingual transfer methods for CLIR introduced in Figure 1.4.

# Part I

# Theoretical Background

# Chapter 2

# Representation Learning

Traditionally, information retrieval (IR) models rely on lexical input features to compute document relevance (Schütze et al., 2008). For example, the query likelihood model (Ponte and Croft, 1998) builds statistical language models to capture probability distributions of document tokens, the tf-idf model (Sparck Jones, 1972) represents queries and documents as sparse vectors capturing word-level frequency statistics (see also Section 3.3). A major shortcoming of these approaches is that they are limited to lexical token matches (i.e., exact keyword matches) between queries and documents.

In this chapter, we review representation learning methods as a way to learn dense semantic vectors (Mikolov et al., 2013c; Devlin et al., 2019), which allow us to represent queries and document based on their meaning. We first review two types of static word embeddings in Section 2.1, followed by an introduction into contextualized embeddings in Section 2.3. Finally, we discuss pre-training methods for multilingual text embeddings in Section 2.3.3. This chapter lays the groundwork for the retrieval models discussed in the rest of the thesis.

## 2.1 Word Embeddings: Bridging the Lexical Gap

As mentioned above and discussed in Section 3.2, lexical retrieval models suffer from their inability to match queries against relevant documents when lexical matches are not present. This is commonly the case in cross-lingual information retrieval (CLIR) where the query and document language use different vocabularies. Shared vocabulary tokens between two different languages are often limited to lexical items that are not translated, such as names or numbers. Semantic text representations do not represent terms by term-level statistics but by low-dimensional dense vectors, known as *word embeddings* (Mikolov et al., 2013d; Bojanowski et al., 2017a). Word embeddings represent words in a *latent semantic vector space* (Deerwester et al., 1990) where, contrary to lexical representations, vector dimensions are not directly interpretable since they encode underlying abstract concepts. Word embeddings allow IR models to be sensitive to synonymy and polysemy.

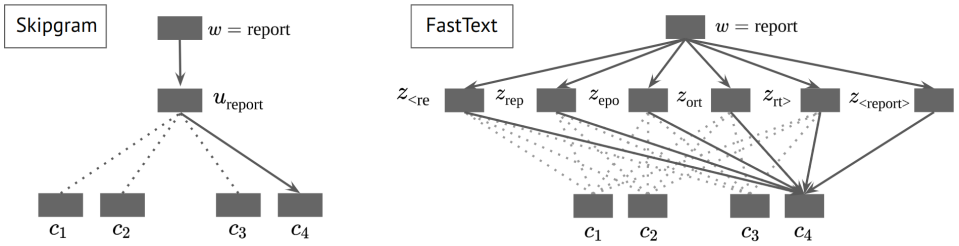


**Figure 2.1:** Illustration of a single forward pass for the Skip-gram model (left) and the fastText model (right) with the current center word $w$ = report and context word $c_4$. Dashed lines connected to different context words $c_i$ represent different forward passes, solid arrows represents a single forward pass. SkipGram illustration adapted from (Mikolov et al., 2013c).

In their seminal *word2vec* paper, Mikolov et al. (2013a) present two model architectures for learning semantic word representations: the continuous bag-of-words model (CBOW) and the SkipGram model. Different from lexical bag-of-words approaches such as tf-idf, which represent terms as high-dimensional sparse vectors, word2vec represents each word by a low-dimensional dense representation. Their method is inspired by the *distributional hypothesis* (Harris, 1954; Sahlgren, 2008), which states that words that appear in similar contexts share similar meanings. Word2vec models are trained by sliding a window of $n$ tokens through a large text corpus. Each window represents a local context and consists of a center word and multiple surrounding context words. During training, word2vec minimizes the distance (i.e., maximizes the similarity) between their respective center and context word embeddings. The similarity between two words is calculated as by the dot product between their word embeddings $sim(a, b) = w_a^\top w_b$. Word2vec follows the distributional hypothesis in the sense that if two words have similar meanings, they are expected to appear oftentimes in the same local context window, and therefore their respective word embeddings should point in a similar direction in the embedding space.

The SkipGram model is a specific word2vec implementation (Mikolov et al., 2013a). It is defined over a vocabulary of size $N$ and parameterized by $\theta = \{U, V\}$ with $U, V \in \mathbb{R}^{N \times d}$. Each vocabulary term is associated with a $d$-dimensional context word embedding $u_w$ and a center word embedding $v_w$. The model learns to predict context words $w_c$ from center words $w_t$, i.e. it maximizes the probability $p(w_c|w_t)$. In the input layer, the center word is represented by a one hot vector

$$x_i = [0, \ldots, 0, 1, 0, \ldots, 0] \tag{2.1}$$

where $i$ refer to the word's position in the vocabulary. During the forward pass, $w_t$ is first projected into a dense embedding with the lookup $u_{w_t} = x_i U$. Next, $u_{w_t}^\top V$ calculates for a given target word its similarity to each vocabulary term. The similarity distribution over the context is then transformed into a probability

| wraps | | reichstag | | german | | artist | |
|---|---|---|---|---|---|---|---|
| wrapping | 0.771 | landtag | 0.744 | austrian | 0.692 | artists | 0.747 |
| wrap | 0.751 | reichskanzler | 0.742 | germann | 0.691 | printmaker | 0.692 |
| wrapped | 0.726 | bundestag | 0.733 | germany | 0.664 | painter | 0.685 |
| wrappings | 0.694 | reichsrat | 0.732 | germans | 0.616 | watercolorist | 0.673 |
| stretchy | 0.560 | abgeordnetenhaus | 0.718 | polish | 0.612 | watercolourist | 0.637 |

**Table 2.1:** Example query terms and their nearest (cosine) neighbors in a pre-trained fastText embedding space (among 200k most frequent embedding terms).

distribution. The SkipGram model computes the probability of a context word for the current center word $p(w_c|w_t)$ by applying the softmax function:

$$p(w_c|w_t) = \frac{e^{u_{w_t} \cdot v_{w_c}}}{\sum_{j=1}^{W} e^{u_{w_t} \cdot v_j}} \tag{2.2}$$

As explained in (Goldberg and Levy, 2014), the model is trained by maximizing the log-likelihood of $p(w_c|w_t)$ for all pairs of $(w_t, w_c)$ found in the training corpus:

$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{c \in \mathcal{C}(w_t)} \log p(w_c|w_t) \tag{2.3}$$

$$= \frac{1}{T} \sum_{t=1}^{T} \sum_{c \in \mathcal{C}(w_t)} (\log e^{u_{w_t} \cdot v_{w_c}} - \log \sum_{j=1}^{W} e^{u_{w_t} \cdot v_j}) \tag{2.4}$$

The function $\mathcal{C}(w_t)$ returns for a current center word $w_t$ the surrounding context words $w_c$ in a given window. After training, one can extract the final word embeddings from the model parameters. The original word2vec implementation[1] uses $w_t$ to represent words. Exhaustively computing similarity scores for all negatives $\sum_{j=1}^{W} e^{u_{w_t} \cdot v_j}$ is intractable for large vocabulary sizes $N$ (Goldberg and Levy, 2014). Motivated by this, Mikolov et al. (2013c) propose to train the SkipGram model (1) over samples of negative examples or (2) with a hierarchical softmax formulation over the vocabulary.

The fastText word embedding model (Bojanowski et al., 2017a) is an extension of the SkipGram model. Figure 2.1 shows a single forward pass for the word "report" and a context word $c_4$ for SkipGram and fastText. In fastText, the model represents every word by its constituent $n$-gram embeddings $g \in \mathcal{G}_w$. That is, instead of a single lookup $u_{w_t} = x^{(i)}U$, each word is now represented by multiple $n$-gram embeddings $z_g$. The similarity score between a target word $w_t$ and context word $w_c$ is then computed as $\sum_{g \in \mathcal{G}_w} z_g^\top v_{w_c}$.

Word embedding models such as SkipGram or fastText learn a mapping from vocabulary terms to dense and low-dimensional word vectors in a latent semantic space where individual dimensions are no longer interpretable. However, the topology of embedding spaces exhibit rich syntactic and semantic regularities between

[1] https://github.com/tmikolov/word2vec/blob/master/word2vec.c

words (Mikolov et al., 2013c,d). For example, in Table 2.1 we show for four words their five nearest neighbors found in a pre-trained fastText embedding space. Here, the distance between words is measured as the cosine similarity between their respective pre-trained fastText embeddings.[2] As mentioned earlier, lexical retrieval methods are limited to aggregating relevance signals from exact token matches between queries and documents. reprocessing steps such as stemming and lemmatization (Porter, 1980; Schütze et al., 2008) alleviate the vocabulary mismatch by normalizing different word forms (`wrapping`, `wrap`, `wrapped`, ...) into a single surface form, they fall short in capturing semantic relationships between different words (`reichstag`, `bundestag`, `abgeordnetenhaus`).

The quality of pre-trained word embeddings can be evaluated on intrinsic evaluation tasks or extrinsically on downstream tasks (Wang et al., 2019b). Intrinsic evaluations probe embeddings for linguistic regularities between words (Mikolov et al., 2013d; Drozd et al., 2016; Wang et al., 2019b). For example, the word analogy task (Mikolov et al., 2013d) uses word embeddings to construct query vectors such as, e.g., $\overrightarrow{q} = vec(\text{“}berlin\text{”}) - vec(\text{“}germany\text{”}) + vec(\text{“}france\text{”})$ and evaluate if their nearest neighbors correspond to analogous words $vec(\text{“}paris\text{”})$.[3] Extrinsic evaluations are grounded in downstream tasks. In the context of retrieval, Roy et al. (2018) evaluate the quality of word2vec and fastText under different configurations. In Chapter 4 we evaluate cross-lingually aligned fastText representations on CLIR. More generally, combining off-the-shelf pre-trained word embeddings together with task-specific model architectures has been the de facto standard in NLP applications (Conneau and Kiela, 2018), until they have been replaced by contextual encoder models (see Section 2.3). In summary, intrinsic evaluations are designed to measure how well embeddings represent syntactic and semantic word-level relationships, and extrinsic evaluation measures the effectiveness of word embeddings when used as input features in downstream tasks.

Training information retrieval (IR) models with pre-trained input features that already encode rich lexical semantics alleviates the need to learn those regularities in task-specific training. In order words, using pre-trained word embeddings decouples “language acquisition” from learning IR-specific features (Guo et al., 2016) such as exact matching signals, query term importance and topic relevance. Retrieval models typically compare query and document tokens to compute relevance signals from token-level interactions (see, e.g., full interaction and late interaction models discussed in Section 3.4). Contrary to lexical word representations such as tf-idf (Sparck Jones, 1972), which are constrained to exact matches, word embeddings allow for *soft matches* between any token-pair. This facilitates capturing more nuanced and semantic token-level interactions beyond exact matches where relevance signals are, e.g., sensitive to synonyms and related terms. Early neural ranking models such as the deep relevance matching model (DRMM; Guo

[2] https://fasttext.cc/docs/en/crawl-vectors.html

[3] We use *vec(·)* to denote a word embedding lookup and use cosine similarity to measure similarity. The example has been computed on the same FastText embedding space as in Table 2.1.

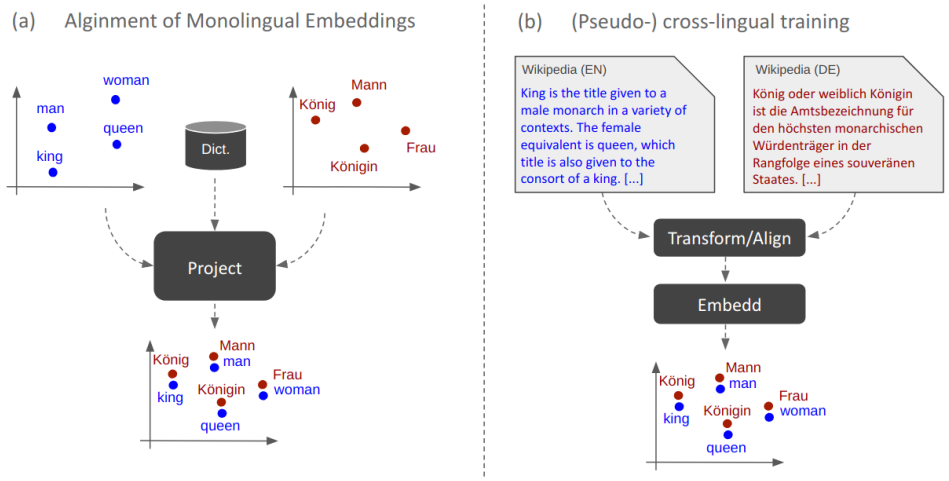


**Figure 2.2:** Overview of cross-lingual word embedding induction methods according to the classification by Ruder et al. (2019): (1) Alignment-based methods rely on monolingual word embedding spaces and use dictionaries of word translation pairs to learn projection(s) into a shared space. (2) Cross-lingual training rely on aligned corpora to obtain pseudo cross-lingual data, on which a monolingual embedding model is trained. Analogy example taken from (Mikolov et al., 2013d).

et al., 2016) or Co-PACCR (Hui et al., 2018) rely on word2vec embeddings to compute input representations from all pair-wise similarities between query and document terms, which are then used to extract higher-order similarities and predict relevance scores. In Chapter 4, we systematically evaluate the effectiveness of different (resource-lean) methods for aligning different monolingual word embedding spaces into a shared cross-lingual embedding space for cross-lingual retrieval.

## 2.2 Cross-lingual Word Embedding Spaces

[1]Cross-lingual Word Embedding (CLWE) spaces are word vector spaces, where semantic features, as described above, are further aligned across languages (Mikolov et al., 2013b; Faruqui and Dyer, 2014; Vulić and Moens, 2015; Joulin et al., 2018; Artetxe et al., 2018; Lample et al., 2018; Hoshen and Wolf, 2018, *inter alia*). According to Ruder et al. (2019), most methods for inducing CLWE spaces can be classified into projection-based approaches which map pre-trained monolingual

[1]This Section is adapted from: (1) **Robert Litschko**, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. 2018. Unsupervised cross-lingual information retrieval using monolingual data only. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pages 1253–1256; (2) **Robert Litschko**, Goran Glavaš, Ivan Vulić, and Laura Dietz. 2019. Evaluating resource-lean cross-lingual embedding models in unsupervised retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1109–1112.

word embedding spaces into a shared space and those that directly apply an embedding method on a (pseudo-) cross-lingual corpus (see Figure 2.2). CLWE methods can be further distinguished by their required level of cross-lingual supervision. Here, we focus on CLWE methods that rely only on word-level alignments as those are easier to obtain than sentence- and document-level alignments, and therefore more suitable for a *resource-lean cross-lingual transfer* of CLIR models.

### 2.2.1 Projection-Based Framework

In the projection-based framework we start from two independently pre-trained monolingual word embedding spaces ($\mathbf{X}_{L1}$ and $\mathbf{X}_{L2}$) and seek to learn the projection/mapping function(s) that either project vectors from one monolingual space to the other or vectors from both monolingual spaces to the new joint vector space (Glavaš et al., 2019). The projection(s) are learned using the dictionary of word translations pairs $D_T = \{w_{L1}^{(i)}, w_{L2}^{(i)}\}_{i=1}^{N}$. Supervised models (Section 2.2.2) use some readily available external translation dictionary usually consisting of few thousand word translation pairs, whereas unsupervised models (Section 2.2.3) induce $D$ automatically (typically iteratively), assuming that approximate isomorphism holds between the monolingual embedding spaces. Using the translation dictionary, projection-based CLWE models create word-aligned matrices – $\mathbf{X}_S = \{\mathbf{x}_{L1}^{(i)}\}_{i=1}^{N}$ and $\mathbf{X}_T = \{\mathbf{x}_{L2}^{(i)}\}_{i=1}^{N}$ – by looking up vectors for aligned words from $D$ in $\mathbf{X}_{L1}$ and $\mathbf{X}_{L2}$, respectively. In the general framework, a CLWE model uses $\mathbf{X}_S$ and $\mathbf{X}_T$ to learn two projection matrices $\mathbf{W}_{L1}$ and $\mathbf{W}_{L2}$, projecting respectively $\mathbf{X}_{L1}$ and $\mathbf{X}_{L2}$ to the shared cross-lingual space $\mathbf{X}_{CL} = \mathbf{X}_{L1}\mathbf{W}_{L1} \cup \mathbf{X}_{L2}\mathbf{W}_{L2}$. In practice, however, many of the models we evaluate learn only a single-direction projection matrix $\mathbf{W}_{L1}$ which projects vectors from $\mathbf{X}_{L1}$ to $\mathbf{X}_{L2}$. This can be seen as a special instantiation of the framework in which $\mathbf{W}_{L2} = I$, i.e., $\mathbf{X}_{CL} = \mathbf{X}_{L1}\mathbf{W}_{L1} \cup \mathbf{X}_{L2}$.

### 2.2.2 Supervised Models

In the following, we review different resource-lean CLWE methods, which we then evaluate on cross-lingual retrieval in Chapter 4. We first examine supervised CLWE models that require an externally created translation dictionary $D$.

**Canonical Correlation Analysis (CCA).** Faruqui and Dyer (2014) treat $\mathbf{X}_S$ and $\mathbf{X}_T$ as different views on the same data points and apply CCA to learn the data representations that maximize the correlation between the two views. CCA learns both projection matrices $W_{L1}$ and $W_{L2}$ and projects both monolingual spaces $\mathbf{X}_{L1}$ and $\mathbf{X}_{L2}$ to the new shared space. For a single pair of word translations $x_i \in \mathbf{X}_{L1}$ and $x_j \in \mathbf{X}_{L2}$, CCA finds projection matrices that maximizes their correlation:

$$\rho(W_{L1}x_i, W_{L2}x_j) = \frac{\text{cov}(W_{L1}x_i, W_{L1}x_j)}{\sqrt{\text{var}(W_{L1}x_i)\text{var}(W_{L1}x_j)}} \tag{2.5}$$

Here, cov$(\cdot,\cdot)$ and var$(\cdot)$ denote the correlation between and variance within latent features of pairs of aligned monolingual word embeddings. This objective is optimized for a pair of aligned word embedding matrices obtained from $D$.

**Euclidean Distance and Procrustes Problem (PROC).** Mikolov et al. (2013b) cast the CLWE induction as a problem of learning the unidirectional projection $\mathbf{W}_{L1}$ that minimizes Euclidean distance between the projected source language vectors $\mathbf{X}_{L1}$ and their corresponding target language vectors $\mathbf{X}_{L2}$:

$$\mathbf{W}_{L1} = \arg\min_{\mathbf{W}} \|\mathbf{X}_{L1}\mathbf{W} - \mathbf{X}_{L2}\| \tag{2.6}$$

By constraining $\mathbf{W}_{L1}$ to an orthogonal matrix, this minimization becomes a well-known Procrustes problem (Xing et al., 2015; Smith et al., 2017) which has the following closed-form solution:

$$\begin{aligned} \mathbf{W}_{L1} &= \mathbf{U}\mathbf{V}^\top, \text{ with} \\ \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top &= SVD(\mathbf{X}_{L2}{\mathbf{X}_{L1}}^\top). \end{aligned} \tag{2.7}$$

In Chapter 4, we evaluate two supervised models based on the solution on the Procrustes problem. First, we evaluate the PROC model that induces $\mathbf{W}_{L1}$ using a larger translation dictionary (5K word translation pairs). The second model, PROC-B, starts from a significantly smaller translation dictionary (1K word pairs): it first learns two single-directional projections – $\mathbf{W}_{L1}$ which induces the cross-lingual space $\mathbf{X}^1_{CL} = \mathbf{X}_{L1}\mathbf{W}_{L1} \cup \mathbf{X}_{L2}$ and $\mathbf{W}_{L2}$ that induces a different cross-lingual space $\mathbf{X}^2_{CL} = \mathbf{X}_{L2}\mathbf{W}_{L2} \cup \mathbf{X}_{L1}$ – and then augments the translation dictionary $D$ with pairs of words that are cross-lingual nearest neighbours according to both projections (i.e., both in $\mathbf{X}^1_{CL}$ and $\mathbf{X}^2_{CL}$). Finally, PROC-B computes the new projection matrix $\mathbf{W}_{L1}$ by solving the Procrustes problem on the augmented dictionary.

**Relaxed Cross-Domain Similarity Local Scaling (RCSLS).** The model of Joulin et al. (2018) learns the projection matrix $\mathbf{W}_{L1}$ by maximizing the ranking-based measure called Cross-Domain Similarity Local Scaling (CSLS) between $\mathbf{X}_S\mathbf{W}_{L1}$ and $\mathbf{X}_T$ (Lample et al., 2018). For a pair of aligned word embedding matrices $X_{L1}$ and $X_{L2}$ the Relaxed CSLS loss is defined as:

$$\begin{aligned} \mathbf{W}_{L1} = \arg\min_{\mathbf{W}} \frac{1}{n}\sum_{i=1}^{n} -2\cos(x^{(i)}_{L1}\mathbf{W}, x^{(i)}_{L2}) \\ + r(x^{(i)}_{L1}\mathbf{W}, X_{L2}) + r(x^{(i)}_{L2}, X_{L1}\mathbf{W}) \end{aligned} \tag{2.8}$$

where $r(x^{(i)}_{L1}\mathbf{W}, X_{L2})$ denotes the average cosine similarity between the $i^{th}$ (translated) source vector $x^{(i)}$ and its k-nearest-neighbors in $X_{L2}$. This term is applied in both directions and adjusts for the "hubness problem" (Doddington et al., 1998;

Dinu et al., 2014). CSLS, commonly used for inference in word translation (Glavaš et al., 2019), is the cosine similarity normalized with the average similarity that each of the vectors has with its cross-lingual nearest neighbors. For the maximization of CSLS to be a convex optimization problem, the constraint that $\mathbf{W}_{L1}$ is orthogonal must be relaxed. By using a BLI inference metric as its learning objective RCSLS is tailored to perform well in bilingual lexicon induction (Irvine and Callison-Burch, 2017), as shown in Section 4.4.2.

### 2.2.3 Unsupervised Models

Unsupervised CLWE models automatically induce seed translation dictionaries without any bilingual data. In Chapter 4, we include models that induce seed dictionaries using different strategies: adversarial learning (Lample et al., 2018), similarity-based heuristics (Artetxe et al., 2018), and principal component analysis (PCA) (Hoshen and Wolf, 2018). After obtaining the seed dictionary $D$, a bootstrapping procedure, similar to the one described for PROC-B, is executed. In the final step, the Procrustes problem is again solved, using the dictionary produced through bootstrapping.

**Heuristic Alignment (VECMAP).** Artetxe et al. (2018) induce the initial seed lexicon by comparing monolingual distributions of word similarities, assuming that word translations have similar distributions of similarities with other words from the same language. For a pair of two un-aligned matrices $X$ and $Z$ VECMAP builds two monolingual similarity matrices $M_x = XX^T$ and $M_Z = ZZ^T$, each row $i$ corresponds to the similarity distribution of the $i^{th}$ term to all other vocabulary terms. Next, the values in each row are sorted (independently) yielding $sorted(M_x)$ and $sorted(M_Z)$ respectively. The initial seed dictionary $D$ is computed with a $k$-nearest-neighbor-search between rows of the two matrices. Word pairs are assumed to be a translation pair (i.e., aligned) if they show similar in-vocabulary similarity distributions. $D$ then expanded in an iterative self-learning bootstrapping procedure. VECMAP's empirical robustness also crucially depends on multiple additional steps: unit length normalization, mean centering, ZCA whitening, cross-correlational re-weighting, de-whitening and dimensionality reduction.

**Adversarial Alignment (MUSE).** Lample et al. (2018) use a Generative Adversarial Network (GAN) architecture that learns a projection $\mathbf{W}_{L1}$ (generator) from $\mathbf{X}_{L1}$ to $\mathbf{X}_{L2}$ until a discriminator $D$ (a deep feed-forward network parameterized by $\theta_D$) cannot distinguish whether a vector originally comes from the target space $X_{L2}$ or has been projected from the source space (i.e., comes from $\mathbf{X}_{L1}\mathbf{W}_{L1}$ by the generator). Specifically, for a set of $n$ source and $m$ target language embeddings the two adversaries are trained successively for every instance with stochastic

gradient descent to minimize

$$\mathcal{L}_D(\theta_D|\mathbf{W}) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}(\text{src} = L2|\mathbf{W}x_{L1}^{(i)}) - \frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}(\text{src} = L1|x_{L2}^{(i)}) \quad (2.9)$$

$$\mathcal{L}_{\mathbf{W}}(\mathbf{W}|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}(\text{src} = L1|\mathbf{W}x_{L1}^{(i)}) - \frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}(\text{src} = L2|x_{L2}^{(i)}), \quad (2.10)$$

where $P_{\theta_D}(\text{src} = L1|z)$ denotes the discriminator's confidence of vector $z$ originating from $X_{L1}$. The initial projection is then improved in an iterative bootstrapping procedure (similar to those used by PROC-B and VECMAP). MUSE strongly relies on isomorphism of monolingual spaces, often leading to poor GAN initialization, particularly for distant languages.

**Iterative Closest Point Model (ICP).** Hoshen and Wolf (2018) induce the small seed dictionary by projecting vectors of $N$ most frequent words from both languages to a lower-dimensional space using PCA. They then search for translation matrices $\mathbf{W}_{L1}$ and $\mathbf{W}_{L2}$ that find the optimal alignment (Euclidean distance) between the two sets of $N$ words in this space low-dimensional space. Since both the projection matrices and optimal word alignment are initially unknown they learn them with the Iterative Closest Point algorithm. In each iteration, ICP first fixes the projections and finds the optimal alignment $D$ and then uses $D$ to update the projection matrices. Next, they employ iterative dictionary bootstrapping and produce the final projection by solving the Procrustes problem.

**Bilingual Word Embeddings Skip-Gram (BWESG).** Vulić and Moens (2015) propose a model that exploits large document-aligned comparable corpora (e.g., Wikipedia).[4] BWESG first creates a merged corpus of bilingual pseudo-documents by intertwining pairs of available comparable documents, as shown in Figure 2.2 (right). It then applies a standard monolingual log-linear Skip-Gram model with negative sampling (SGNS) on the merged corpus in which words have bilingual contexts instead of monolingual ones (Mikolov et al., 2013c).

## 2.3 Contextual Representations

A significant limitation of pre-computed word embeddings is their static nature. Representing each vocabulary term with a single embedding conflates different meanings, rendering word embeddings inherently ineffective in dealing with polysemy. Several works attempt to mitigate this shortcoming with so called multi-prototype embeddings, which represent words with multiple sense embeddings (Reisinger and Mooney, 2010; Tian et al., 2014; Cao et al., 2017; Arora et al., 2018, *inter alia*). The idea of pre-training static word embeddings has later been

[4] We refer the reader to the survey by Ruder et al. (2019) for a broad overview.

superseded by pre-training deep language models (LM) as universal text encoders that compute dynamic representations based on the context of surrounding words (Howard and Ruder, 2018; Peters et al., 2018; Brown et al., 2020; Devlin et al., 2019). Early encoder-based language models such as ULMFit (Howard and Ruder, 2018) and ELMo (Peters et al., 2018) pre-train variants of long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997) as encoders. In the following, we focus on *Transformer-based text encoder models* (Vaswani et al., 2017; Devlin et al., 2019), which represent the state-of-the-art architecture for modelling text representations and are still used in large language models (LLM) (Touvron et al., 2023; OpenAI, 2023; Meta, 2024).

### 2.3.1 Transformers

**Subword Tokenization.** Most Transformer-based language models tokenize their input text into subwords (Devlin et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020; Wang et al., 2020a, *inter alia*). Sharing subword representations across different words leads to fewer model parameters and thereby reduces the overall model size. They also allow models to represent a larger vocabulary space and to obtain representations for previously unseen morphological variations of words and rare terms, similar to fastText (Bojanowski et al., 2017a). In this regard, subwords strike a good balance between the expressiveness of (1) fully character-based input representations which lead to very long input sequences and (2) word-based input representations which are not able to represent out-of-vocabulary (OOV) terms (Sun et al., 2023). In retrieval tasks, excessively long character-based representations would have a negative impact in query latency and slow down retrieval. On the other hand, word-based representations fail to capture relevance signals embedded in previously unseen tokens. Especially unseen words often still contain seen subwords that carry important information. For example, mBERT's WordPiece tokenizer (Devlin et al., 2019)[5] splits the word `bundestagsabgeordneter` (eng. member of the parliament) into the following subword tokens: `bundestag`, `##sa`, `##b`, `##geordnete` and `##r`. In this example, a word embedding model that has never seen the word `bundestagsabgeordneter` is unable to represent the word in a meaningful way. A Transformer-based language model with a subword tokenizer, however, might still be able to use information encoded in the seen token `bundestag` and contextualize it with the suffix tokens. A downside of subword tokenizers is that they lack robustness (i.e., they over-segment words) when applied under domain shifts (Sun et al., 2023) and when exposed to noisy text or text containing typos (Kumar et al., 2020; Sun et al., 2023).

Subword tokenizers are extracted from (i.e., trained on) large corpora prior to pre-training language models (Wu et al., 2016; Sennrich et al., 2016; Kudo and Richardson, 2018; Kudo, 2018). WordPiece (Wu et al., 2016) and byte pair encoding (BPE) (Sennrich et al., 2016) are among the most widely used tokenizers. Both

[5] We use HuggingFace's `bert-base-multilingual-cased` tokenizer (Wolf et al., 2020).

models first pre-tokenize a given corpus into words to obtain a word frequency distribution. Next, the initial vocabulary is derived from all observed characters. The vocabulary is then iteratively updated by merging the most frequent neighboring tokens (token bi-grams) based on co-occurrence statistics.[6] This process is repeated until a desired vocabulary size is reached. At test time, the input text is tokenized again first into words and characters, which are then successively merged into subwords, either by following a greedy approach (WordPiece) or by applying learned merge rules (BPE). Repeatedly merging frequently co-occurring token bigrams into new vocabulary tokens has the effect that high frequent words are more likely to be represented by a single vocabulary entry, whereas rare words (long tail) are more likely to be decomposed into multiple tokens. A trained Subword tokenizer can therefore be seen as a compression of the corpus it was trained on (Gage, 1994; Sennrich et al., 2016). Next, we discuss the model architecture used to learn dynamic, i.e. context-sensitive subword representations.

**Model Architecture.** In their seminal paper, Vaswani et al. (2017) present a multilayer neural encoder-decoder model architecture. The underlying building blocks, also known as Transformer layers, have become the *de facto* standard method for encoding text in NLP and IR applications. In the following, we focus on the encoder part (henceforth, *Transformer*). Like word embeddings (WE) discussed in Section 2.1, Transformers are trained to represent text in a latent semantic space. Different from WE models, token representations are computed dynamically for each input. That is, the representation of a given token is influenced (i.e. *contextualized*) by all other tokens present in the same input sequence, whereas static WEs are invariant to the context in which they are used at inference time. On a high level, the Transformer model consists of $L$ layers, which are recursively applied to transform a sequence of $N$ subword tokens $t_1, t_2 \dots t_N$ into a sequence of semantic token embeddings $h_i^{(l)} \in \mathbb{R}^{d_{\text{model}}}$, where $d_{\text{model}}$ is the model dimensionality. It is also known as the embedding size or hidden size of a Transformer model.

$$Transform(t_1, t_2 \dots t_N) = h_1^{(L)}, h_2^{(L)} \dots h_N^{(L)} \tag{2.11}$$

In the input layer (a.k.a. *embedding layer*), each token $t_i$ is represented as the sum of its static subword embedding $w_i = WE(t_i)$ and a vector that encodes its position $p_i = PE(i)$.[7]

$$h_1^{(0)}, h_2^{(0)} \dots h_N^{(0)} = w_1 + p_1, w_2 + p_2 \dots w_N + p_N, \tag{2.12}$$

Each subsequent layer is a parameterized function that recursively refines the token representations of the previous layer:

$$h_1^{(l)}, h_2^{(l)}, \dots, h_N^{(l)} = Layer(h_1^{(l-1)}, h_1^{(l-1)}, \dots, h_N^{(l-1)}). \tag{2.13}$$

[6] Here, BPE and WordPiece differ in how token bi-grams are scored. Please refer to the original papers for further details on their scoring function.

[7] We refer the reader to (Vaswani et al., 2017) for more details on position encoding.

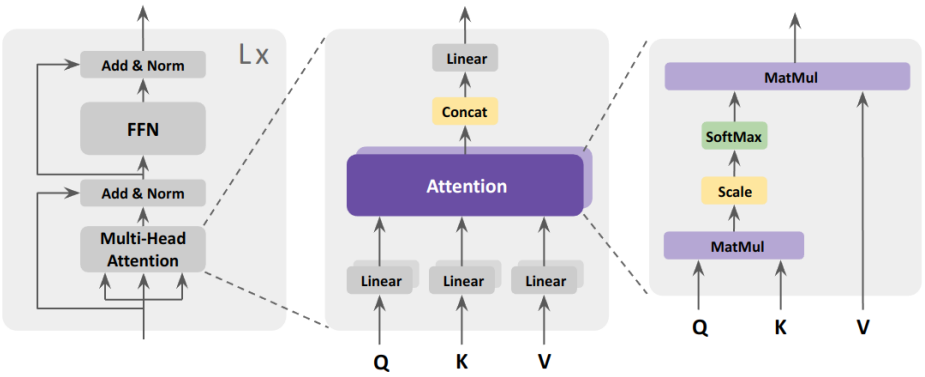


**Figure 2.3:** Overview of a single Transformer encoder block/layer. Left: The encoder block is applied to each input token. Middle: Multi-head attention components. Right: Decomposition of the attention mechanism. Image adapted from (Vaswani et al., 2017).

Processing token representations from bottom layers up until the output layer increasingly contextualizes their representations. Prior work suggests that pre-trained language models (PLM) have learned to encode lexical features in lower layers and semantic features in upper layers (Tenney et al., 2019a; Jawahar et al., 2019; Niu et al., 2022; Aoyama and Schneider, 2022).[8] In principle, lower layers in PLMs could refine the representation of `bundestag` with lexical information stored in the representations of `##sa`, `##b`, `##geordnete` in order to represent the term `bundestagsabgeordneter`. Upper layers could further contextualize this with information from surrounding words to signify, e.g., a specific person or political party the term refers to. The output representations of the last layer are typically used as general-purpose text features in downstream tasks (Devlin et al., 2019; Nogueira and Cho, 2019). In the rest of this section, we describe the different Transformer sub-layers and use the same notation as in (Vaswani et al., 2017).

*Attention mechanism* pre-dates Transformers and describes a class of functions that contextualize token-level representations with information distributed across a pre-specified context of input tokens (Bahdanau et al., 2015; Luong et al., 2015; Brauwers and Frasincar, 2023). In Transformer (encoder) models, the context corresponds to the input token sequence in which a word appears. This can be implemented in different ways; early and widely adopted approaches include additive attention (Bahdanau et al., 2015) and multiplicative attention (Luong et al., 2015). On a high level, both approaches can be described as follows: **(1)** First, compute pair-wise token alignments between a given target token and a set of context tokens (i.e., scalar attention scores). **(2)** Next, normalize attention scores into a probability distribution (attention weights). **(3)** Finally, compute an updated representation

[8] In Section 5.5.3 we validate this observation for CLIR and show that the optimal layer and degree of contextualization differ between sentence-level and document-level retrieval.

of the target token by aggregating information from its surrounding tokens (i.e., compute a weighted sum over all input embeddings). The token alignments computed in the first two steps dynamically control how much each neighboring token informs the representation of a given target token.

We now describe the *scaled dot-product attention* mechanism introduced in the Transformer model (Vaswani et al., 2017) and shown in Figure 2.3 (right). For a given a sequence of $N$ tokens, each token $t_i$ is represented by a query, key, and value vector $q_i, k_i, v_i \in \mathbb{R}^d$, which are obtained from a linear transform of the input representation. Query and key vectors are used to measure pairwise token alignments, i.e. attention scores, as follows. **(1)** In the first step, the alignment between the $i^{th}$ target token and $j^{th}$ context token is expressed as the dot product between their query and key vector $q_i k_j^T$. Stacking all key vectors $K = \{k_1, \ldots, k_N\} \in \mathbb{R}^{N \times d}$ for a given sequence of length $N$ allows us to compute all attention scores at once with $q_i K^T$. **(2)** These scores are then normalized into a probability distribution (i.e., attention weights) with $\alpha = softmax(qK^T)$, as shown in Equation 2.2. **(3)** Finally, the updated representation for the $i^{th}$ token, $h_i$, is computed as a weighted sum over all value vectors: $h_i = \sum \alpha_i v_i$. The Transformer model concatenates all query, key and value vectors into their respective matrices $Q, K, V \in \mathbb{R}^{N \times d}$, computing the scaled dot-product attention with

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{2.14}$$

where $d_k$ is the hidden size of key vectors and $\sqrt{d_k}$ is a normalizing constant scaling down large dot-product values (Vaswani et al., 2017).

*Multi-head attention*, which has also been introduced in the Transformer paper, uses $h$ different attention heads. This allows the model to "pay attention to different aspects" (Figure 2.3, middle). For example, Clark et al. (2019) show that attention heads of trained LMs capture (i) general patterns such as attending the whole sequence, attending punctuation, or positional offsets; and (ii) syntactic patterns where attention heads attend to objects of verbs, determiners of nouns, and coreference mentions. Each attention head captures a different aspect in its vector $\text{head}_i$. A single attention head is parameterized by the matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$. These are used to project a sequence of input embeddings into sequences of query, key and value embeddings respectively.

$$\text{head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.15}$$

Finally, the multi-head attention sub-layer concatenates all attention heads into a single matrix $N \times hd_v$, which are then projected back to the original model dimensionality $d_{\text{model}}$ with a linear transformation matrix $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O. \tag{2.16}$$

After contextualization, Transformer layers apply a position-wise *feed-forward network* (FFN) (Rumelhart et al., 1986). Each input token $x$ is passed through a two-layered neural network with a ReLU activation function (Glorot et al., 2011).

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{2.17}$$

While Transformer-based LMs use attention to contextualize token representations, prior work shows that FFNs encode internal knowledge (i.e., parametric knowledge) that is acquired during pre-training (Mallen et al., 2023; Neeman et al., 2023). For example, Meng et al. (2022) find that factual and linguistic knowledge can be localized in FFN parameters of pre-trained LMs.

Input vectors $x$ and output vectors Sublayer$(x)$ of both sub-layers (multi-head attention, FFN) are connected with *residual connections*: $y = x + \text{Sublayer}(x)$ (He et al., 2016). Finally, *layer normalization* (Ba et al., 2016) re-scales token vectors $y_i$ by subtracting the vector mean from each value and dividing it by the vector's standard deviation. Residual connections and layer normalization are used to improve the training efficiency of deep neural networks with many layers. In summary, $Transform(\cdot)$, as shown in Equation 2.11, constructs the input embeddings from static subword embeddings and position encodings, which are then contextualized in each layer $l = 1 \dots L$ with

$$h_1^{(l)}, h_2^{(l)} \dots h_N^{(l)} = \text{FFN}_l(\text{MultiHead}_l(h_1^{(l\text{-}1)}, h_2^{(l\text{-}1)} \dots h_N^{(l\text{-}1)})), \tag{2.18}$$

until the final output layer $L$.[9] Next, we describe how Transformer-based language models are pre-trained to learn to encode tokens into semantic representations.

### 2.3.2 Pre-trained Language Models

The original Transformer architecture is implemented as an encoder-decoder model and trained and evaluated on neural machine translation (NMT) and constituency parsing (Vaswani et al., 2017). In the following years, a plethora of Transformer-based pre-trained language models (PLM) have been proposed. Most language models can be grouped into "encoder-only" models (Devlin et al., 2019; Conneau and Lample, 2019; Liu et al., 2019; Zhang et al., 2019; Conneau et al., 2020) and autoregressive "decoder-only" models (Brown et al., 2020; Yang et al., 2019b; Raffel et al., 2020; Xue et al., 2021). In the following, we focus on the first type of PLMs due to their widespread application in information retrieval (see Section 3.4).

BERT (Bidirectional Encoder Representations from Transformers) is the first Transformer-based PLM and has been trained with a bidirectional objective (Devlin et al., 2019). Similar to word embeddings discussed in Section 2.1, the goal is to learn semantic text representations. Contrary to word embeddings, PLMs encode input text dynamically, i.e. the same token in different contexts is represented with

[9] For brevity, we omit layer normalization and residual connections.

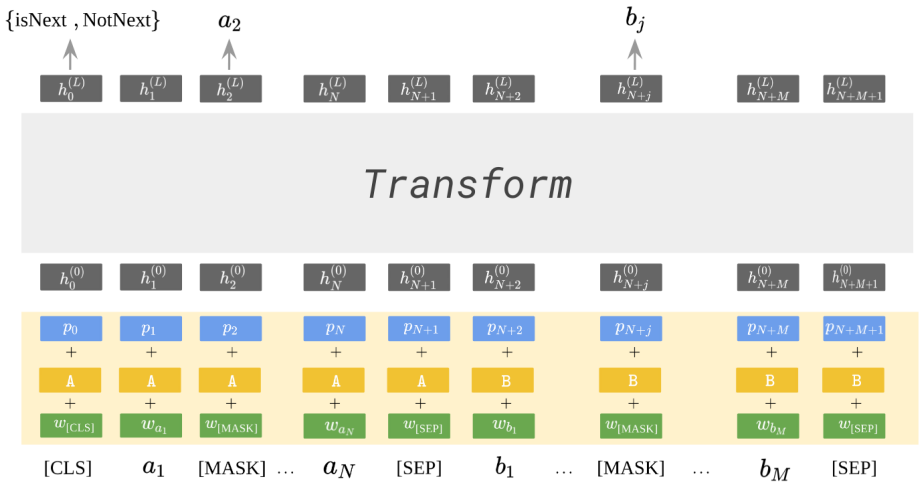


**Figure 2.4:** Overview of pre-training BERT models: A training instance is constructed by pairing two text segments (sentences) `A` with $N$ subword tokens $a_i$ and `B` with $M$ subword tokens $b_j$, which is augmented by special tokens (`[CLS]`, `[SEP]`) and perturbed by randomly masking tokens (`[MASK]`). The input layer aggregates subword, segment and position embeddings, leading to input token embeddings $h_i^{(0)}$. These are then contextualized with a Transformer encoder model. The pre-training task is to reconstruct masked-out tokens from their contextualized representations $h_i^{(L)}$ in the final layer (MLM) and predict whether sentence `B` is a randomly sampled sentence or the sentence that follows `A` in the training corpus (NSP). Image adapted from (Devlin et al., 2019).

different embeddings. This allows encoders to be sensitive towards the meaning of words based on their context. BERT consists of an input layer and a Transformer encoder, as shown in Figure 2.4. The input layer uses a WordPiece tokenizer (Wu et al., 2016) to tokenize text into subwords. Each subword is represented as the sum of its token embedding, position embedding and segment embedding. The model is pre-trained on sentence pairs `A` and `B` extracted from a large text corpus. Here, BERT uses segment embeddings to encode whether a given token belong to the first or second sentence. The model also uses a special separator token `[SEP]` to mark sentence boundaries and a sequence classification token `[CLS]` token to allow the model to learn sequence-level representations when it is fine-tuned on different downstream tasks. For a given sentence pair, the input token sequence is formatted as `[CLS] A [SEP] B [SEP]`. Next, the input embeddings are contextualized with a Transformer encoder

$$\text{BERT}(\texttt{A}, \texttt{B}) = h^{(L)}_{\texttt{[CLS]}}, h^{(L)}_{\texttt{a}_1}, \ldots h^{(L)}_{\texttt{[SEP]}}, h^{(L)}_{\texttt{b}_1} \ldots h^{(L)}_{\texttt{[SEP]}}. \tag{2.19}$$

Devlin et al. (2019) train BERT on a large training corpus consisting of the English Wikipedia and the BookCorpus (Zhu et al., 2015). For this, the authors intro-

duce two novel pre-training objectives: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM and NSP are examples of *self-supervised learning* objectives (Gui et al., 2024) where the labels for the task (i.e., the task supervision) is obtained from the data itself, as described next.

MLM is a de-noising objective where the model is trained to reconstruct the original text from perturbed text, where tokens are randomly masked out or replaced with other tokens. To train BERT, Devlin et al. (2019) replace 15% of the tokens in a sentence with the `[MASK]` token (80% of the times), a random token (10%) or not at all (10%). At each mask position $i$, the model is tasked to predict the original token from its contextualized token embedding $h_i^{(L)}$. BERT reuses the subword embedding matrix to compute the output probability distribution over the vocabulary $o = softmax(h_i^{(L)} W_{\text{emb}}^T)$. The MLM loss $\mathcal{L}_{\text{MLM}}$ is the cross-entropy between the predicted output distribution $o$ and the one-hot vector of the original token (before masking). In this regard, MLM is similar to the CBOW word embedding model (Mikolov et al., 2013a) where target words are predicted from context words, following the distributional hypothesis (Harris, 1954).

NSP is a self-supervised binary classification task. Given a sentence pair `A` and `B`, the task is to predict if `B` is a randomly sampled sentence (50% of the times) or if it is the actual next sentence of `A`. Here, the `[CLS]`-embedding is used as input for a binary softmax classifier (i.e., classification head) $softmax(h_{\texttt{[CLS]}}^{(L)} W_{\text{head}}^{\top})$. The rationale behind NSP is that many NLP tasks can be framed as sentence-pair prediction tasks (Devlin et al., 2019), and teaching BERT to encode the relationship between sentence pairs in the `[CLS]`-embedding is expected to be helpful when the model is fine-tuned on downstream tasks. Both pre-training objectives are jointly optimized $\mathcal{L}_{\text{BERT}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}}$. Devlin et al. (2019) show that fine-tuning BERT exhibits state-of-the-art results (at the time) on the popular general language understanding evaluation (GLUE) benchmark (Wang et al., 2019a).

In a follow-up work, Liu et al. (2019) find that the original BERT model is undertrained and present RoBERTa, which is a BERT model that has been trained under improved conditions. Those include upscaling the training data from 13GB to 160GB, increasing the batch size from 256 to 8K, omitting the NSP loss and masking input dynamically during training (Liu et al., 2019). The authors show that RoBERTa significantly improves upon BERT on the GLUE benchmark.

**Fine-tuning pre-trained language models.** After language models are pre-trained once on large corpora, they are then fine-tuned on much smaller task-specific datasets (Peters et al., 2018; Devlin et al., 2019). To do so, PLMs are augmented by task-specific parameters stacked on top of the Transformer encoder. These are jointly fine-tuned with all other model parameters on a downstream task. MLM and NSP, as discussed above, are examples of token-level and sequence-classification tasks. In Section 3.4, we describe how retrieval models rely on PLMs to (i) frame predicting the relevance of documents as a classification task or (ii) specialize their output representations to reflect semantic similarity.

**BERTology.** Following the success of BERT and RoBERTa, a line of research dubbed BERTology emerged (Rogers et al., 2020a), which studies the language understanding capabilities of PLMs by probing their internal representations. For example, prior work finds that PLMs encode syntactic knowledge (i.e., structural information) such as parts-of-speech, dependency structures and co-references (Tenney et al., 2019b,a; Hewitt and Manning, 2019; Wu et al., 2020). Evidence for other language understanding capabilities found in PLMs include lexical knowledge (Vulić et al., 2020), conceptual/ontological knowledge (Peng et al., 2022; Wu et al., 2023) and common-sense knowledge (Davison et al., 2019; Lin et al., 2020). Furthermore, Wiedemann et al. (2019) find that contextualization allows PLMs to represent different senses of polysemous words into different semantic subspaces. In summary, PLMs such as BERT encode rich semantics in their representations. These are acquired at pre-training time and can be utilized by task-specific models that are fine-tuned on downstream tasks.

### 2.3.3 Multilingual Representations

[1]In this section, we first discuss multilingual pre-trained language models (mPLM) and models specialized for sentence-level similarity, which we later investigate for CLIR (Chapter 5). We then discuss the zero-shot cross-lingual transfer (ZS-XLT) paradigm, which we adopt in Chapters 6 and 7.

**Multilingual Language Model Pre-training.** The multilingual extension of BERT (mBERT) follows the same training approach and is trained on the concatenation of the 104 largest Wikpedias (Devlin et al., 2019). During training, the model is exposed to batches with sentences mixed from different languages. Accordingly, mBERT's vocabulary size has been increased from 30K to 110K subwords. To control for the imbalance of the different languages, the authors apply a smoothed sampling approach which under-samples instances from high-resource languages and over-samples instances from low-resource languages (during model training and tokenizer training).[10] Conneau and Lample (2019) present two cross-lingual language model (XLM) pre-training objectives: causal language modeling (CLM) and translation language modeling (TLM). In CLM, the task is to predict the next token given all previous tokens. TLM is functionally similar to MLM, except it feeds the model with sentence translation pairs, which allows it to attend tokens in one language while predicting masked-out tokens in the other language (Conneau and Lample, 2019). Conneau et al. (2020) later introduced with XLM-RoBERTa (XLM-R) a model that is trained on a corpus of 2TB common crawl and Wikipedia data spanning 100 languages.

[1]Our discussion on multilingual sentence encoders is adapted from: **Robert Litschko**, Ivan Vulic, Simone Paolo Ponzetto, and Goran Glavaš. 2021. Evaluating multilingual text encoders for unsupervised cross-lingual retrieval. In *Advances in Information Retrieval: 43rd European Conference on IR Research (ECIR)*, pages 342–358, Lucca, Italy (Online)

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

**Training Multilingual Sentence Encoders.** An extensive body of work focuses on specializing multilingual encoders to capture sentence meaning. In Artetxe and Schwenk (2019a), the multilingual encoder of a sequence-to-sequence model is shared across languages and optimized to be language-agnostic, whereas Guo et al. (2018) rely on a dual Transformer-based encoder architecture instead (with tied/shared parameters) to represent parallel sentences. Rather than optimizing for translation performance directly, their approach minimizes the cosine distance between parallel sentences. A ranking softmax loss is used to classify the correct (i.e., aligned) sentence in the other language from negative samples (i.e., non-aligned sentences). In (Yang et al., 2019a), this approach is extended by using a bidirectional dual encoder and adding an additive margin softmax function, which serves to push away non-translation-pairs in the shared embedding space. The dual-encoder approach is now widely adopted (Guo et al., 2018; Yang et al., 2020b; Feng et al., 2022; Reimers and Gurevych, 2020; Zhao et al., 2020a), and yields state-of-the-art multilingual sentence encoders which excel in sentence-level NLU tasks. Other recent approaches propose input space normalization and using parallel data to re-align mBERT and XLM (Zhao et al., 2020a; Cao et al., 2020), or using a teacher-student framework where a student model is trained to imitate the output of the teacher network while preserving high similarity of translation pairs (Reimers and Gurevych, 2020). In (Yang et al., 2020b), authors combine multi-task learning with a translation bridging task to train a universal sentence encoder. We benchmark a series of representative sentence encoders in this thesis; their brief descriptions are provided in Section 5.3.1.

**Zero-shot Cross-Lingual Transfer.** Being able to represent text from multiple languages in a shared input space allows us to (1) fine-tune a model on a language where we have access to task-specific training data and (2) apply it in a different language in which we have no training data. This is also known as *zero-shot cross-lingual transfer* (ZS-XLT) (Hu et al., 2020) where, due to the availability of training data, models are typically transferred from English as the source language to other target languages. Multilingual Transformers have shown to exhibit state-of-the-art performance on standard cross-lingual language understanding benchmarks such as XCOPA (Ponti et al., 2020), XTREME (Hu et al., 2020) and XGLUE (Liang et al., 2020). Wu and Dredze (2019) show for different cross-lingual classification and sequence labelling tasks that mBERT, which has been trained without any explicit cross-lingual supervision (i.e., parallel data) performs surprisingly well in ZS-XLT. This raises the question of which factors contribute to mBERT's multilinguality. Prior work on understanding mBERT's multilinguality can be grouped into different lines of research, as discussed next.

*(i) Representation space topology:* To understand mBERT's multilinguality, several studies investigated the topology of the vector representations extracted from mBERT (Wu and Dredze, 2019; Pires et al., 2019; Roy et al., 2020; Lim et al., 2024, *inter alia*). Wu and Dredze (2019) probe mBERT's internal repre-

sentations and find that all layers perform well at the language identification task, suggesting that each layer indeed encodes language-specific information. Cao et al. (2020) study mBERT's contextualized word representations, captured in the words' last subword embeddings, and find that they are weakly aligned. That is, even though word representations of translation pairs are close to another, there still exists language clusters. This finding has been independently confirmed by Pires et al. (2019); Roy et al. (2020); Lim et al. (2024). For example, Pires et al. (2019) evaluate mBERT's multilingual representational alignment on a sentence similarity task using parallel data consisting of 5K sentence translation pairs. The authors claim, for mBERT to be truly multilingual, the distance between any sentence translation pair should be invariant to the sentences themselves. To test this hypothesis for a given language pair, the authors first encode each sentence into a sentence embedding by averaging its constituent wordpiece embeddings (excluding `[SEP]` and `[CLS]`). The authors use the average distance over all sentence translation-pairs to represent a global language offset, which is then used to translate (i.e., shift) sentence vectors from one language subspace to another. They find that in more than 50% of the cases, when representations are extracted from the middle and upper layers, the nearest neighbors of the shifted vectors correspond to translations.

*(ii) Lexical similarity:* Many languages share some common vocabulary terms such as named entities, anglicism and numerals. Pires et al. (2019) test the hypothesis that mBERT's strong ZS-XLT performance can be attributed to simply memorizing the label-token relationships of shared tokens. To do so, the authors first measure the extent to which the fine-tuning and test languages share common sets of entity wordpieces. They then compare this overlap with the downstream performance on named entity recognition (NER). Their results suggest that mBERT's transfer performance goes beyond simple vocabulary memorization, and that the model is even able to transfer to languages written in entirely different scripts. Lastly, Wu and Dredze (2019), Dufter and Schütze (2020) and Deshpande et al. (2022) show for different NLP tasks that overlapping tokens between train and test languages has a strong impact on zero-shot XLT performance.

*(iii) Source language selection effects:* In the ZS-XLT paradigm one can compare the test language against (1) the pre-training languages used in (masked) language modelling, and (2) the source language on which a PLM is subsequently fine-tuned. Malkin et al. (2022) investigate (1) on ZS-XLT for NER and part-of-speech (POS) tagging. The authors show empirically that the pre-training languages influence downstream performance. They find that some languages are good "donor" languages (i.e., they improve a model's ZS-XLT performance) while other languages are good "recipient" languages (i.e., they benefit most from donor languages). Lauscher et al. (2020) investigate (2) and study English as a source transfer language for syntactic and semantic NLP tasks. Their findings show that the ZS-XLT performance correlates with the typological proximity between the source and target language. Turc et al. (2021) find that other source languages often outperform English, even when the data is machine translated from English. Other studies investigate the synergies of transferring from multiple source lan-

guages (Chen et al., 2019; Lim et al., 2024), and transferring between languages from the same language family (Snæbjarnarson et al., 2023; Senel et al., 2024).

*(iv) Structural similarity:* K et al. (2020) define structural similarity as "anything that is invariant to the script of the language" and include morphology, word ordering and word frequency as structural properties of languages. To disentangle the effects of vocabulary overlap and structural similarity, the authors generate fake English data $EN_{Fake}$ and perturb English training data by shifting the Unicode of all characters by a large offset.[11] The generated data preserves structural properties of EN but has no vocabulary overlap with any other language. Using three target languages (TgT), K et al. (2020) train bilingual BERT models and compare the performance between transferring from perturbed data ($EN_{Fake}$→TgT) and transferring from original English data (EN→TgT). Contradictory to prior work (Pires et al., 2019; Wu and Dredze, 2019), their results show no significant difference between the two, i.e. removing wordpiece overlap with the target language does not drastically impact downstream performance. On the other hand, they found that perturbing $EN_{Fake}$ data by shuffling wordpieces leads to large performance drops. Pires et al. (2019) investigate the transfer performance with respect to structural similarity between the training and the test language in two ways. First, using WALS features (Dryer and Haspelmath, 2013) related to grammatical ordering, the authors show that a larger degree of structural similarity comes with performance improvements on POS tagging. Secondly, the authors group languages according to the two typological features **S**ubject/**O**bject/**V**erb order and **A**djective/**N**oun order. They then compare the performance between transferring to languages with the same features (e.g. SVO→SVO and AN→AN) against transferring to languages with different features (e.g., SVO→SOV and AN→NA). Their experimental results show that the best performance is achieved when the source and target language share the same structure.

In this thesis, we investigate these aspects in the context of cross-lingual retrieval and cross-lingual transfer for IR. For *(i)*, we empirically evaluate the suitability of representation spaces induced by mBERT and XLM when used as off-the-shelf query and document encoders in Chapter 5. For *(ii)*, we study the impact of vocabulary overlap on zero-shot cross-lingual rerankers and propose to reduce the vocabulary overlap in the training data as a way to improve their generalization performance (Chapter 7). For *(iii)*, throughout this thesis we experiment with a diverse set of languages from different language families including high-resource languages and low-resource languages, see Section 3.3 for an overview and Chapter 10 for a discussion of our findings. For *(iv)*, in the context of cross-lingual transfer with multiple models, we compare predicting the best source languages against a baseline that uses selects the source language based on structural similarity (Chapter 9).

[11] A similar method has been applied in (Dufter and Schütze, 2020).

## 2.4 Conclusion

In this chapter, we reviewed representation learning methods that allow us to encode text into semantic representations. We first discussed monolingual static word embedding methods (Section 2.1) and different approaches of aligning them into shared cross-lingual embedding spaces (Section 2.2). In Chapter 4, we present two CLIR models based on static cross-lingual embeddings and evaluate their performance on CLIR downstream tasks. We also reviewed pre-trained language models (PLM) and their multilingual variants in Sections 2.3 and 2.3.3. PLMs are used in current neural IR models, which we discuss in the next chapter. In the main part of this thesis, we first investigate the suitability of PLM as multilingual query and document encoders for representation-based CLIR (Chapter 5) and then focus on zero-shot cross-lingual transfer for CLIR (Chapters 6 to 8).

# Chapter 3

# Cross-Lingual Information Retrieval

To facilitate information access beyond language boundaries, we need retrieval systems capable of interpreting information needs expressed in any language and matching them with relevant content written in any language. This goal entails two tasks, namely monolingual IR *within* different languages (MoIR) and cross-lingual IR *across* different languages (CLIR). Compared to MoIR, CLIR is arguably the more challenging task, as models cannot rely on exact keyword matches when the query and document language vocabularies do not overlap. In this thesis, we study *ad-hoc document-level CLIR*. That is, we assume every search interaction to be independent and do not use any data from past user interactions. We specifically focus on resource-lean methods that enable us to transfer CLIR models to other language pairs without relying on expensive large scale training data.

In this chapter, we first introduce CLIR and provide an overview of standard evaluation benchmarks (Section 3.1). In Section 3.2, we discuss why lexical retrieval models cannot be directly applied due to the lexical gap (between languages), and how the lack of training resources makes it difficult to scale CLIR to many languages. In Section 3.3, we then introduce our evaluation protocol including datasets, evaluation metrics and lexical retrieval baselines used throughout most of the following chapters. We finally discuss different neural retrieval paradigms in Section 3.4.

## 3.1 Introduction and Overview

In this section, we first introduce the standard evaluation paradigm used in information retrieval research (Section 3.1.1). Next, we provide a historical overview of different CLIR evaluation campaigns (Section 3.1.2), followed by an overview of cross-lingual retrieval in NLP applications (Section 3.1.3).

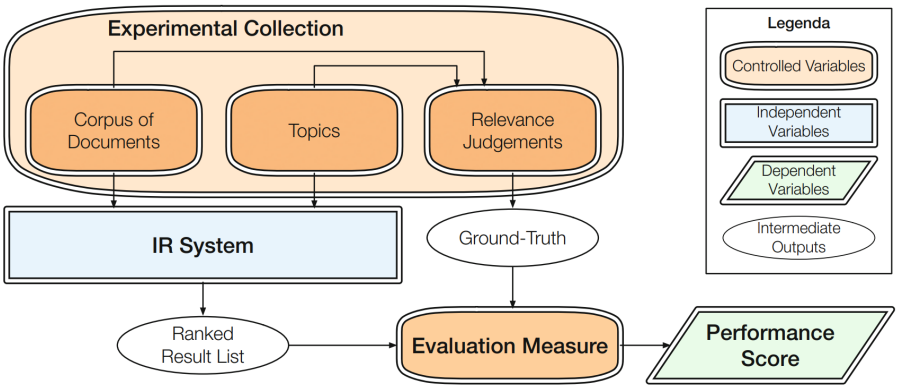


**Figure 3.1:** The Cranfield Evaluation Paradigm. Here, the experimental collection is also known as *test collection*. Image source: (Ferro and Peters, 2019, p.7). Reproduced with permission from Springer Nature.

### 3.1.1 Cranfield Evaluation Paradigm

**Online and Offline Evaluation.** IR research is oftentimes concerned with comparing the performance of two or more retrieval systems. For example, we might want to test if a proposed model outperforms a baseline system (alternative hypothesis $H_1$) or whether there is no significant difference (null hypothesis $H_0$). As discussed in (Ferro and Peters, 2019), to test such a hypothesis in a controlled experiment, researchers define the components they can control and manipulate (*independent variables*) and measure their observable effect on the *dependent variables* that capture the quality of the retrieval systems (Cook et al., 2002; Hofmann et al., 2016, p.11). They run experiments to collect data and perform statistical tests (Carterette, 2017) to either accept $H_1$, concluding that observed differences between two competing systems are significant, or reject $H_1$, attributing observed differences to random chance. With online evaluation (Hofmann et al., 2016) and offline evaluation (William, 1967), there exist two widely used evaluation paradigms.

*Online evaluation* is a user-centric approach that compares IR systems in a live environment, where effects are measured by analyzing the behavior of real users (Hofmann et al., 2016). A/B testing randomly separates user traffic into two groups, the treatment group is exposed to a new/modified IR system and the control group is exposed to the current system (Kohavi et al., 2009; Hofmann et al., 2016, p.20). The effect of the group assignment (independent variable) is then measured with metrics that capture user behavior (dependent variable). Another example of online evaluation is interleaving (Chapelle et al., 2012; Radlinski and Craswell, 2013; Hofmann et al., 2016, p.26). Here, users are exposed to search results where individual items are interleaved from two (or more) retrieval systems. Using implicit feedback (Kelly and Teevan, 2003) from user behavior such as click data or dwell time as a proxy for document relevance has a *limited re-usability*, because experiments are tied to a dynamic environment.

```
<top>

<num> 142

<title> Christo wraps German Reichstag

<desc> Find reports on the wrapping of the German
Reichstag in Berlin by the artist Christo.

<narr> The wrapping artist Christo took two weeks
in June 1995 to wrap the Berlin German Reichstag
in lengths of material.  Find reports on this
artistic event.  Any information on either
its preparation or its execution is relevant,
including political debates and decisions and
technical preparations.

</top>
```

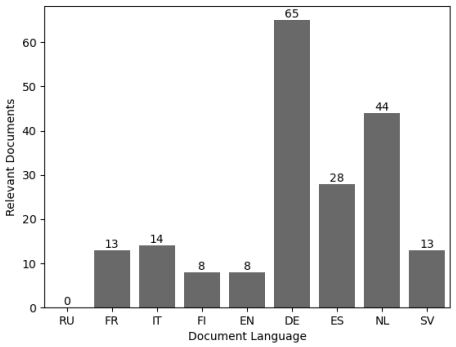


**Figure 3.2:** Left: Example information need taken from the CLEF 2003 dataset (Braschler, 2004). Right: Distribution of relevant documents across document languages.

*Offline evaluation.* The Cranfield evaluation paradigm (William, 1967) is the standard approach used in most evaluation initiatives (Section 3.1.2). It formalizes standard test environments for offline evaluation of retrieval systems in so-called test collections (see Figure 3.1). A test collection consists of a document corpus, a set of information needs (topics) and relevance judgments. Offline evaluation is the standard approach adopted by TREC (Voorhees et al., 2005) and CLEF (Voorhees, 2019; Ferro and Peters, 2019), two of the largest IR evaluation campaigns. Contrary to online evaluation, the relevance of documents is not implicitly inferred from user behavior but explicitly annotated by experts (ground truth). Evaluation measures compare the ground truth with rankings produced by IR systems to compute a performance score. Controlling for the test collection and evaluation measure facilitates re-usability and allows for comparing the effectiveness of IR systems (independent variables) under same the test conditions. Next, we describe each component of a test collection in more detail.

**Information Needs.** Interactions between an IR system and a user typically begin with an *information need*, which is a psychological state (Cooper, 1971) referring to a users' desire to find out more about a certain topic (Schütze et al., 2008). Information needs can arise in a conversational context or as part of accomplishing tasks, which in turn can trigger multiple information needs. For example, planning a holiday trip might involve looking for flights, researching different travel destinations and learning about local cuisines. Information needs are then verbalized into queries, e.g., "When are flight tickets cheapest?" or "Best city trips in Europe". Historically, TREC formalizes information needs into structured representations known as *TREC topic statements* (Voorhees et al., 2005). As shown in Figure 3.2 (left), each topic describes an information need at different levels of granularity, which are stored in different fields. The title field represents the information need as a short query limited to a few words, similar to how one would per-

```
<DOC>

<DOCID>GH950620-000097

<TITLE>It's a wrap in the name of art

<TEXT> WORKERS lower a giant panel of cloth
over the entrance to the Reichstag in Berlin,
helping Hungarian artist Christo to fulful
a dream of 24 years.  Christo and his wife
Jeanne Claude are using a #4.6m loan secured
on their private art collection to fund the
work of covering the former German Parliament
in silver fabric.  [...]

</DOC>
```

```
<DOC>

<DOCID>FR940123-000072

<TITLE>Christo

<TEXT> Des Künstlers Plan, den Berliner
Reichstag zu verpacken, ist bei
Umweltschützern auf Kritik gestoßen.  Unter
Umweltgesichtspunkten, so urteilt Michael
Braungart, Vorsitzender des Hamburger
Umweltinstituts, ist das Projekt eine
Schweinerei.  [...]

</DOC>
```

**Figure 3.3:** Two relevant CLEF documents in German and English for the information need shown in Figure 3.2. For brevity, we only show the first two sentences and exclude unused metadata. Both documents are part of the CLEF 2003 dataset (Braschler, 2004).

form a keyword search. The description field provides more context and describes the information need in one sentence. The narrative is a complete description of the information need and allows annotators to determine whether a document is relevant and meets an information need. In CLEF test collections, topics are developed to cover events in a broad range of domains such as politics, culture, sports, science and translated into different languages (Womser-Hacker, 2001; Braschler, 2004). Topic development and relevance annotations are performed by hired and trained native speakers also known as *assessors*. For example, most assessors employed by NIST are retired intelligence analysts (Soboroff, 2021). In the NTCIR-1 and NTCIR-2 evaluation campaigns, assessors were researchers in scientific domains (Kando, 2000). As discussed in Section 1.1.2, the high costs associated with hiring and training assessors, developing topics and collecting relevance judgments make it impractical to create large-scale CLIR datasets for many languages.

**Document Collection.** Relevant information can be present in different modalities such as text, image, and video. In this thesis, we focus on text retrieval where individual sentences, passages or entire articles can be relevant. We refer to a single searchable text unit as *document* and to the set of all documents as *document collection* or *corpus*. The CLEF 2003 benchmark (Braschler, 2004) is a popular test collection used for ad-hoc document-level CLIR. Here, the document collection consists of news articles written in different languages. Similar to topic statements, documents are organized in structured representations, each document has a document ID, a title and a text field. In Figure 3.3 we show two relevant documents for our earlier example query (Figure 3.2). The English document discusses the event of wrapping the "Reichstag in Berlin" on a rather high level, and the German document discusses political aspects related to its environmental impact. This is an example of information asymmetry, as discussed in Chapter 1. It shows how CLIR enables access to not only more information but also more diverse and localized information.

**Relevance Judgments.** The notion of relevance emerges from a users' information need and from the intent behind a query. For example, the intent behind the query *"implementing batch normalization in Python"* might be asking for passages describing how to do the task (Answer Retrieval) or looking for specific code examples (Code Retrieval).[1] The main difference between the Cranfield paradigm and online user studies is that in Cranfield relevance is not implicitly inferred from user behavior but explicitly annotated by experts. Those annotations are also known as relevance judgments or relevance assessments.

Relevance can be assessed on a binary scale or on a scale of different degrees of relevance (Cooper, 1971). As mentioned in Section 1.1.2, manually assessing an entire corpus for each topic is too expensive for any reasonably large document collection. In many IR applications such as web search most queries only have a few relevant documents, skewing the label distributions towards non-relevant. Annotating a random sample is therefore likely to miss relevant documents. In other words, it is non-trivial to obtain documents that are likely relevant. *Pooling* (Spark-Jones, 1975) is a commonly applied method for creating a pool of promising candidate documents to be manually annotated. For any given topic, the idea is to collect $n$ different search rankings $\mathcal{R}_i$, e.g., from teams participating in the same shared task. The pool depth $k$ corresponds to the top-$k$ documents taken from each ranking $\mathcal{R}_i(k)$. Finally, the candidate document pool to be annotated is the union of all documents that appear at a high rank in any participating ranking:

$$\bigcup_{i=1}^{n} \mathcal{R}_i(k). \tag{3.1}$$

The pool size and thus the annotation budget for a single topic bounded by $n$ rankings times $k$ documents (Voorhees, 2019, p.82).

In CLIR, we assume the notion of relevance to be language agnostic, which means that relevant documents can be present in any language. The CLEF 2003 benchmark (Braschler, 2004) adopts a binary relevance scale, where a document is judged as relevant if it contains any relevant portion. Due to information asymmetry, content in different languages may cover the same topic to different extends (Section 1.1.1). Naturally, the number of relevant documents written in a language may be influenced by whether the topic, e.g., refers to a local event where the language is spoken. For example, the topic shown in Figure 3.2 (right) is covered more often in German news articles than in other languages (see also Appendix B.1).

### 3.1.2 Historical Test Collections

Cross-lingual IR (CLIR) has been studied for a long time in different research communities. In the following, we provide an overview of some CLIR evaluation initiatives and their development in chronological order.

[1] This example is taken from (Asai et al., 2023) and shows that IR goes beyond lexical matching. In fact, Fan et al. (2021b) show that different retrieval tasks require different linguistic skills to model relevance.

**Text REtrieval Conference (TREC).** In 1992 the US National Institute of Standards and Technology (NIST) initiated TREC, which is a leading information retrieval workshop that regularly organizes shared tasks related to text retrieval, also known as TREC Tracks. The earliest official TREC CLIR tracks were organized in the years between 1997 and 1999: TREC-6 (Schäuble and Sheridan, 1998), TREC-7 (Braschler et al., 1999) and TREC-8 (Braschler et al., 2000).[2]

Initially developed independently of TREC, the Microsoft MAchine Reading COmprehension (MS MARCO) dataset (Nguyen et al., 2016) is a large-scale English retrieval benchmark that contains labels, documents and queries from real user interactions. Queries correspond to user questions and are sampled from Bing's search log, documents are passages extracted from websites retrieved with Bing and relevance is manually annotated by editors. The MS MARCO dataset has been used in the TREC Deep learning Tracks from 2019 until 2022 (Craswell et al., 2020a, 2021a, 2022, 2023). Bonifacio et al. (2021) later created a multilingual version of MS MARCO, dubbed mMARCO, where queries and documents are machine translated into thirteen different languages.

In 2022 CLIR has returned to TREC with the recent NeuCLIR Track (Lawrie et al., 2023). The organizers use the HC4 dataset (Lawrie et al., 2022), which includes Common Crawl News documents in Chinese, Persian and Russian. Importantly, different from prior TREC tracks, NeuCLIR also includes a large training split. In the following year, TREC NeuCLIR also included retrieval tasks of technical documents written in Chinese and multilingual news retrieval (Lawrie et al., 2024). It's current iteration, NeuCLIR 2024,[3] extends prior shared tasks by report generation where CLIR systems are evaluated in the context of retrieval augmented generation (Lewis et al., 2020b).

**Cross-Language Evaluation Forum (CLEF).** The CLEF initiative emerged as a separate community in Europe to focus on cross-lingual and multilingual retrieval evaluation for European languages (Voorhees, 2019). The first CLEF CLIR evaluation campaign (CLEF 2000) involved seven different query languages and English news articles (Braschler, 2001). In the following years, CLEF also included news articles from different countries and written in different languages (Braschler, 2003, 2004). Specifically, the CLEF 2003 benchmark[4] has expanded to eighth European languages (Braschler, 2004) and for example included news articles, among others, from the German newspaper *Frankfurter Rundschau* and news magazine *Der Spiegel*, and the English newspaper *Glasgow Herald*. CLEF corpora were used to organize tracks for monolingual, cross-lingual and multilingual retrieval. In the years 2004 to 2009 CLEF continued to expand their evaluation campaigns to different retrieval tasks and additional languages, for a comprehensive historical review of CLEF we refer to (Ferro and Peters, 2019).

[2] In 1994 TREC-4 organized a shared task on multilingual retrieval (Davis and Dunning, 1995).

[3] https://neuclir.github.io/2024

[4] https://catalogue.elra.info/en-us/repository/browse/ELRA-E0008/

Later in 2010 the CLEF initiative has been rebranded to Conference and Labs of the Evaluation Forum and broadened the scope to include, e.g., retrieval in other modalities such as image, speech and video (Voorhees, 2019). Most recently, Bonab et al. (2019) extended the CLEF 2000-2003 test collections with new query translations into the low-resource African languages Swahili and Somali. To further advance research in low-resource languages, we contribute three additional CLEF query translations into Kyrgyz, Uyghur and Turkish (see Chapter 8).

**NII Testbeds and Community for Information access Research (NTCIR).** [5] Next to the US and European IR research communities, the Japanese National Institute of Informatics (NII) established the NTCIR initiative[6] to hold regular workshops on IR tasks. Their focus lies on CLIR involving Asian languages. The first workshop (NTCIR-1) took place in 1999 and included CLIR with Japanese queries and English document collections (Kando et al., 1999). Documents consisted of scientific abstracts from a broad range of disciplines and domains (Ferro and Peters, 2019). Later NTCIR iterations in 2002 to 2007 (i.e., NTCIR-3 to NTCIR-6) also included the languages Chinese, English, Japanese, Korean (Chen et al., 2002; Kishida et al., 2004a; Abdou and Savoy, 2005; Kando, 2007).

**Forum for Information Retrieval (FIRE)** is an initiative established by the Indian Statistical Institute in 2008 to promote the development of retrieval systems for Indian languages (Hindi, Bangla, Marathi, Tamil, Telugu, Punjabi, Malayalam) (Mitra and Majumdar, 2008).[7] In the following years, FIRE organized different CLIR shared tasks. FIRE-2010 (Majumder et al., 2013b) was the second iteration of ad-hoc cross-lingual document retrieval. FIRE-2011 (Palchowdhury et al., 2013) additionally included the tasks of Cross-Language Indian Text Reuse (CL!TR), i.e., plagiarism detection (Barrón-Cedeno et al., 2013) and SMS-based Cross-lingual FAQ Retrieval with noisy text coming from "SMS language" (Contractor et al., 2013). FIRE-2013 (Majumder et al., 2013a) included the Cross-Language !ndian News Story Search (CL!NSS) Track (Gupta et al., 2013).

### 3.1.3 Cross-Lingual Retrieval for NLP

In the previous section, we discussed with TREC, CLEF, NTCIR and FIRE four major CLIR initiatives. For a comprehensive survey including CLIR initiatives from other different IR communities, we refer the reader to (Galuščáková et al., 2021). CLIR has also been studied in the natural language processing (NLP) community. We now describe three related research and application areas where cross-lingual retrieval is used.

[5] Formerly called National Center for Science Information Systems (NACSIS).
[6] https://www.nii.ac.jp/dsc/idr/en/ntcir/ntcir.html
[7] http://fire.irsi.res.in/fire/static/data

**CLIR for Machine Translation (MT).** Word retrieval (Cao et al., 2020) and bilingual lexicon induction (Irvine and Callison-Burch, 2017) are two tasks where cross-lingual retrieval is evaluated on the word-level. The motivation for these tasks is two-fold. First, they allow us to measure the lexical alignment of multilingual representation spaces (see Sections 2.2 and 2.3.3), and, secondly, they allow us to obtain translation dictionaries, which, e.g., can be used to induce cross-lingual embedding spaces (Section 2.2.1). This resource-lean approach is suitable when parallel data (i.e. sentence translation-pairs) is difficult to obtain. Early approaches in CLIR rely on dictionaries for query translation and query augmentation (Ballesteros and Croft, 1996; Adriani and Van Rijsbergen, 1999; Peters et al., 2012). In this thesis, we use bilingual dictionaries to translate queries (Chapter 4) and code-switch queries and documents (Chapter 7).

Cross-lingual sentence retrieval (a.k.a. bi-text mining) is motivated by the goal of automatically extracting parallel data from large multilingual corpora to train machine translation models (Zweigenbaum et al., 2018; Artetxe and Schwenk, 2019b). Regular shared tasks are organized by the *Workshop on Building and Using Comparable Corpora* (BUCC) (Zweigenbaum et al., 2018).[8] In addition to BUCC corpora, the Tatoeba dataset (Artetxe and Schwenk, 2019b) is also commonly used to evaluate sentence-level retrieval. Finally, a closely related task, cross-lingual semantic textual similarity (Hercig and Kral, 2021).

**CLIR for Question Answering (QA).** Cross-lingual answer retrieval is an integral part of question answering systems (Asai et al., 2021b; Roy et al., 2020; Limkonchotiwat et al., 2022; Zheng et al., 2022; Albalak et al., 2023). The retriever-reader framework (Asai et al., 2021b) divides the task of QA into (i) answer retrieval,[9] i.e. finding the passage that contains the answer, and (ii) machine reading comprehension where answers are extracted from retrieved text (Lewis et al., 2020a; Ni et al., 2019). The recent *Workshop on Multilingual Information Access (MIA)* involved a shared task to evaluate this paradigm in the cross-lingual domain (Asai et al., 2022). Similar approaches have been applied in multilingual fact checking (Gupta and Srikumar, 2021) and cross-lingual fake news detection (Dementieva and Panchenko, 2021), where, e.g., CLIR is used to retrieve supporting evidence in order to predict the truthfulness of claims (Huang et al., 2022).

In retrieval-augmented generation (RAG) (Lee et al., 2019; Lewis et al., 2020b; Mallen et al., 2022), support passages are not used to extract or predict the answer but instead to condition the answer generation process, e.g., with a pre-trained sequence-to-sequence (seq2seq) model (Lewis et al., 2020b). Following seq2seq RAG, Shi et al. (2022) propose a framework for cross-lingual Text-to-SQL generation, where the model is provided with a foreign language user question and instructed to generate a SQL statement based on English database schemata. Here,

[8] https://comparable.limsi.fr/bucc2023/

[9] This is also known as Cross-Lingual Retrieval Question Answering (CL-ReQA) (Limkonchotiwat et al., 2022) and Language agnostic Retrieval QA (LaReQA; see Roy et al., 2020)

CLIR is used to find the closest English examples (pairs of user questions and SQL statements) in order to condition the generation of the SQL statement. Augmenting prompts with additional examples is also known as in-context learning (Brown et al., 2020) and widely used in decoder-only large language models (LLMs) such as ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023; Bubeck et al., 2023). In in-context learning, LLMs solve a task formalized by a task description (instruction), a few in-context examples of expected input-output pairs and a test instance to be solved. In the context of cross-lingual retrieval, Nie et al. (2023) and Lin et al. (2024) study CLIR as means to improve in-context learning in low-resource languages by retrieving semantically similar examples in high-resource languages.

Finally, a task that is closely related to answer retrieval is Frequently Asked Question (FAQ) retrieval (Mass et al., 2020). The goal is to retrieve those question-answer pairs that best match the intent and semantics of a user question. In cross-lingual FAQ retrieval questions are formulated in a language that is different from the language in which an existing collection of question-answer pairs are written (Contractor et al., 2010; De Bruyn et al., 2021).

**CLIR on Wikipedia data.** There are numerous Wikipedia-based CLIR benchmarks focusing on different languages and aspects of cross-lingual document retrieval (Schamoni et al., 2014; Sasaki et al., 2018; Sun and Duh, 2020; Ogundepo et al., 2022; Li et al., 2022, *inter alia*). WikiCLIR (Sasaki et al., 2018) covers CLIR in twenty-five language pairs. To obtain relevance labels, the authors derive monolingual rely on links between articles, which are propagated to other languages with inter-language links. Queries and documents are extracted from the first sentence and first 200 words of the Wikipedia articles. CLIRMatrix (Sun and Duh, 2020) follows a similar approach and extends CLIR evaluation to 139 language pairs. Here, relevance labels are derived from lexical retrieval scores instead (see BM25 in Section 3.3), which are discretized and propagated to foreign language documents with inter-language links. AfriCLIRMatrix (Ogundepo et al., 2022) is created the same way and covers English queries and 15 diverse African document languages. Finally, MuSeCLIR (Li et al., 2022) is derived from disambiguation pages and focuses on lexical translation ambiguity.

As discussed in (Huang et al., 2023), an advantage of automated approaches of creating CLIR benchmarks based on Wikipedia is that articles are authored by native speakers and therefore are of high quality. A downside is that queries and relevance judgements are synthetically derived and thus do not reflect the distribution of real information needs and query formulations. For the MIRACL dataset, Zhang et al. (2023c) hired native speakers to obtain realistic questions and relevance labels based on Wikipedia paragraphs. Their focus, however, lies on monolingual retrieval in eighteen different languages. The authors use Wikipedia article snippets to inspire human annotators to formulate queries. This approach has also been adopted in TyDi QA (Clark et al., 2020), which has later been extended to CLIR with the XOR QA dataset (Asai et al., 2021a).

## 3.2 Main Challenges

In this section, we first discuss one of the fundamental challenges of CLIR: bridging the lexical gap between languages, which is the reason why unsupervised lexical retrieval methods ineffective for CLIR. We then discuss the challenge of obtaining high quality training data for neural CLIR models and the limitations of using machine translation.

**Lexical Gap (Between Languages).** The lexical gap describes a phenomenon where queries and documents express the same concept with different terms (Berger et al., 2000). Lexical retrieval models, which are based on exact term matches, fall short in dealing with linguistic variety such as for example polysemy and synonymy. Polysemous words have different meanings and can belong to different word classes.[10] Matching words only based on their surface form can therefore inadvertently overestimate the relevance of non-relevant documents and lead to *false positives*. The opposite holds for synonyms, where words with different surface forms have the same meaning. Failing to match query terms with synonymous document terms leads to *false negatives*. Different heuristics exist to improve the robustness of lexical retrievers towards lexical variations: Term-level normalization approaches such as stemming ("reports" → "report") and lemmatization ("took" → "take") translate different morphological surface forms to a common canonical form (Porter, 1980; Schütze et al., 2008). Rather than normalizing tokens, queries can also be augmented with new terms (query expansion) to better reflect the term distribution of the document collection and bridge the lexical gap (Rocchio Jr, 1971; Abdul-Jaleel et al., 2004; Azad and Deepak, 2019). In summary, lexical methods suffer from the *vocabulary mismatch problem* because of their inability to model semantic relatedness, linguistic variation and topical similarity. This issue is further amplified in cross-lingual retrieval scenarios, where the *lexical gap between languages* is caused by differences in language vocabularies. In Section 2, we discussed how neural language models representing terms in a semantic space (Devlin et al., 2019; Conneau and Lample, 2019). This allows neural retrieval models (Section 3.4) to compute "soft matches" between any query-document term pair, capturing their semantic similarity.

**Training Resources: Quantity vs. Quality Trade-off.** Access to a large amount of relevance judgments is key for training (and evaluating) neural CLIR models. As discussed in Section 1.1.2, human-annotated relevance assessments are of high quality but are expensive to obtain. Because of this reason, many traditional ad-hoc document retrieval benchmarks (Section 3.1.2) contain a limited number of queries and are mainly used to evaluate CLIR systems. For example, the CLEF

[10] For example, the description field in our example (cf. `<desc>` in Figure 3.2) contains the word "report", for which WordNet 3.1 (Miller, 1995) lists thirteen different meanings and two different word classes (noun, verb).

2003 dataset contains only 60 queries (Braschler, 2004). There are different ways to reduce the high cost associated to relevance assessments (Faggioli et al., 2023a), we now discuss two widely adopted approaches.

A commonly adopted low-cost approach in document-level CLIR is to use Wikipedia as a resource to automatically derive benchmarks (Section 3.1.2). For example, the CLIRMatrix benchmark supports 139 language pairs with a total of 49.3M queries and 50.5M documents (Sun and Duh, 2020). Wikipedia-based CLIR datasets use article links (Sasaki et al., 2018) or discretized lexical matching scores (Sun and Duh, 2020; Ogundepo et al., 2022) to obtain relevance labels, which are then propagated to other languages with inter-language links. The shortcoming of (automatically derived) Wikipedia-based benchmarks is that they are limited to encyclopedic knowledge with synthetic queries and relevance assessments, which might not be representative of how humans express natural information needs. Other benchmarks such as MS MARCO (Craswell et al., 2021b) and MIRACL (Zhang et al., 2023c) contain queries written by humans and relevance judged by humans. However, these benchmarks are created to evaluate monolingual instead of cross-lingual retrieval. As of today, there exist no large scale CLIR dataset with human-annotated relevance judgments and natural queries.

Recent works use generative models to generate synthetic data (Bonifacio et al., 2022; Dai et al., 2023; Mayfield et al., 2023; Thakur et al., 2023). For example, InPars (Bonifacio et al., 2022) and Promptagator (Dai et al., 2023) generate synthetic queries from corpus documents. Similarly, Askari et al. (2023) generate passages for queries with ChatGPT (OpenAI, 2022). Mayfield et al. (2023) adopts this approach for CLIR and generate for pairs of target language documents English queries such that one document is relevant, and the other is non-relevant. The authors discuss error categories such as generating under- and overspecified queries and hallucinations. Empirically, their proposed method shows mixed results when compared to machine translation, which we discuss next.

**Limitations of Machine Translation (MT).** There are two standard ways of using MT to close the language gap in CLIR. One can either translate queries at test time (*translate test*) or translate training data to obtain supervised ranking models in our target language pair of interest (*translate train*) (Artetxe et al., 2023). In practice, however, MT-based CLIR approaches relying on commercial translation systems such as Google Translate (Li and Cheng, 2018; Shi et al., 2021) are limited by its language coverage. Furthermore, Bonifacio et al. (2021) show that translation quality and retrieval effectiveness are weakly correlated. In the context of low-resource African languages, Ogundepo et al. (2023) show that MT-based CLIR models still perform poorly. Similarly, our qualitative analysis in Section 8.4 reveals that MT can cause unwanted artifacts such as topic shifts, repetition and hallucinations. We refer the reader to (Guerreiro et al., 2023) for a comprehensive classification of different types of hallucinations in neural MT.

Moreover, people in different geographic regions are naturally interested in

different entities such as local organizations, places and people. Callahan and Herring (2011) investigate how different Wikipedia versions contain varying amounts of information on "local heroes". Such cultural biases play a crucial role in CLIR and other knowledge-intensive NLP applications (Ponti et al., 2020; Peskov et al., 2021; Asai et al., 2021a) and cause a domain mismatch between translated training data and test data (Shen et al., 2021). For example, consider a hypothetical world where we have access to an oracle translation model. Translating English training data would adapt it to lexical and syntactic properties of the target languages; however, it would not change the content (i.e., topic and entity distribution) of the source language (Peskov et al., 2021; Asai et al., 2021a). Consequently, cultural and topical biases propagate to CLIR models trained on translated data. One way to account for local interests and cultural differences is to involve non-English native speakers from different geographic regions in the development of CLIR datasets, as done for example in the recent question answering benchmarks XOR-QA (Asai et al., 2021a) and AfriQA (Ogundepo et al., 2023).

## 3.3 Evaluation Protocol

**Ranking Notation.** In this thesis, we focus on ad-hoc document-level ranked retrieval. The task input is a list of queries $Q$ derived from a respective list of information needs (i.e., topics), a document collection $\mathcal{C}$ and relevance assessments for query-document pairs. We adopt the notation from Lin et al. (2021b) and denote with $q \in Q$ a single query with its constituent query tokens $t_i^q$ and with $d$ a single document with the tokens $t_i^d$. In ranked retrieval, the task is to return a ranked list of documents $R = [(d_1, s_1), (d_2, s_2) \dots (d_n, s_n)]$. This ranking is created such that documents are sorted according to their relevance scores in descending order from most relevant to least relevant $s_1 > s_2 > \cdots > s_n$. We denote the number of queries as $|\mathcal{Q}|$ and the set of rankings after evaluating each query as $\mathcal{R} = \{R_i\}_{i=1}^{|\mathcal{Q}|}$.

Relevance scores $s_i$ can be computed in different ways. Vector space retrieval methods (Schütze et al., 2008) first encode queries and documents independently of each other into vector representations $\overrightarrow{q}$ and $\overrightarrow{d}$, and then use vector space similarity measures to compute relevance scores $s = sim(\overrightarrow{q}, \overrightarrow{d})$. Lexical retrieval models such as tf-idf (Sparck Jones, 1972) encode queries and documents as sparse vectors, which capture term-level statistics extracted from corpora. In the Chapters 4 and 5 we investigate semantic text encoders known as bi-encoders (Karpukhin et al., 2020), where query and documents are represented with dense semantic embeddings (see Section 3.4). A different way to obtain relevance scores is to frame it as a machine learning task, also known as learning-to-rank (Liu, 2009). In the Chapters 6 to 8 we follow the cross-encoder approach first proposed by Nogueira et al. (2019c) and treat scoring a documents' relevance as a classification task. For this, we train a supervised model $s = f(q, d)$ to jointly encode query-document pairs and predict the document relevance. This paradigm is also known as pointwise learning-to-rank (Liu, 2009, p.33).

**Lexical Relevance Matching Baselines.** Relevance scores $s_i$ are computed from relevance signals. Different retrieval paradigms can be distinguished by the way they aggregate relevance signals and score documents. *Lexical retrieval* models (Sparck Jones, 1972; Robertson et al., 1995; Ponte and Croft, 1998) compute relevance signals based on *exact term matches* between queries and documents. Lexical scoring functions such as the tf-idf model (Sparck Jones, 1972) use term weighting schemes involving variants of term frequency $\text{tf}(t, d)$ and inverse document frequency $\text{idf}(t)$ (Schütze et al., 2008). The simplest weighting scheme uses the raw frequency of a term $t$ in a document and calculates idf as $\text{idf}(t) = \frac{N}{\text{df}(t)}$, where $N$ is the corpus size and df(t) the document frequency, i.e. the number of documents containing $t$. Intuitively, if an exact matching term appears more often in a given document, then a higher term frequency $\text{tf}(t, d)$ increases its relevance signal. Conversely, if it also appears in a large number of documents, then a lower $\text{idf}(t)$ reduces its relevance signal. Using tf-idf weights and a vocabulary of size $|V|$, we can represent documents and queries in a $|V|$-dimensional vector space $\overrightarrow{q}, \overrightarrow{d} \in \mathbb{R}^{|V|}$ and compute the relevance score by their inner product:

$$s^{\text{tf-idf}}(Q, D) = \sum_{t \in q \cap d} \text{tf-idf}(t, d) = \langle \overrightarrow{q}, \overrightarrow{d} \rangle \tag{3.2}$$

Among lexical scoring models, BM25 is arguably one of the most widely used scoring function (Robertson et al., 1995):

$$s^{\text{BM25}}(q, d) = \sum_{t \in q \cap d} \log \frac{N - \text{df}_t + 0.5}{\text{df}(t) + 0.5} \cdot \frac{\text{tf}(t, d) \cdot (k_1 + 1)}{\text{tf}(t, d) + k_1 \cdot \left(1 - b + b \cdot \frac{l_d}{L}\right)} \tag{3.3}$$

In BM25 each relevance signal is the product of two components: The first component is a smoothed variant of $\text{idf}(t)$ and captures a global perspective, terms that appear in very few documents are useful in discriminating between documents. On the other hand, terms that appear in all documents, such as words that function as grammatical building blocks, do not contribute towards contrasting relevant from non-relevant documents. The second factor captures a local perspective of relevance. Here, the term frequency $\text{tf}(t, d)$ is additionally scaled by the document length $l_d$ relative to the average document length $L$. The impact of the normalization factor $\frac{l_d}{L}$ and $\text{tf}(t, d)$ is controlled by two parameters $b$ and $k_1$.

The query likelihood model (QLM) is a probabilistic lexical relevance matching model (Ponte and Croft, 1998). QLM scores documents according to their probability of being relevant to a given query $P(d|q)$. Instead of directly estimating this probability, the authors propose to apply the Bayes rule to reformulate the probability as $P(d|q) = P(q|d)P(d)/P(q)$. The probability $P(q)$ is independent of documents and thus dropped, because it does not impact on the document ranking. $P(d)$ can be used to encode a priori relevance criteria such as "authority, length, genre, newness and number of previous people who have read the document" (Schütze et al., 2008). Assuming each document to be equally likely relevant, we can drop $P(d)$, which reduces QLM to estimating $P(q|d)$. The QLM

model estimates $P(q|d)$ with document-side unigram language models $\mathcal{M}_d$. Each model captures document-specific term probabilities based on their term frequencies and the document length $P(t|\mathcal{M}_d) = \text{tf}(t,d)/l_d$. The relevance score with the QLM model $s^{\text{QLM}}$ breaks down into query term probabilities:

$$s^{\text{QLM}}(q,d) = P(q|d) = \prod_{t \in q} P(t|\mathcal{M}_d) = \prod_{t \in q} \frac{\text{tf}(q,d)}{l_d} \tag{3.4}$$

QLM scores each document according to $s^{\text{QLM}}(q,d)$ and ranks documents according to the likelihood of the query being generated by their language models $\mathcal{M}_d$. In Chapter 4, we use lexical matching as a baseline for CLIR, where it is limited to shared tokens between languages such as numbers and named entities. In the Chapters 6 to 8, we use lexical matching as a baseline in (i) monolingual IR in different languages and (ii) in a pipelined approach based on machine translation.

**Evaluation Metrics.** In most parts of this thesis (Chapters 4 to 8) we use Mean Average Precision (MAP) (Harman, 1992) to evaluate our retrieval models. Following the notation from (Lin et al., 2022), the Average Precision (AP) for a single query is defined as

$$\text{AP}(R,q) = \frac{\sum_{(k,d) \in R} \text{Precision@}k(R,q) \cdot \text{rel}(q,d)}{\sum_{d \in \mathcal{C}} \text{rel}(q,d)}. \tag{3.5}$$

where $R = [d_1, d_2, \ldots, d_N]$ is the ranking for a single query $q$. The binary function $\text{rel}(q,d)$ indicates whether $d$ is relevant for $q$. Precision@$k$ evaluates the fraction of relevant documents out of all top-$k$ documents. Computing the mean of all AP-values queries yields the Mean Average Precision (MAP):

$$\text{MAP}(\mathcal{R}, \mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum \text{AP}(R,q) \tag{3.6}$$

In this thesis, we also use the Mean Reciprocal Rank (MRR) (Kantor and Voorhees, 2000). MRR is applicable when there is only one relevant document for a given query, or when the user seeks a specific document that is known to exist or has been seen before (Craswell, 2009). For a given set of $N$ queries $Q = \{q_i\}_{i=1}^{N}$ and corresponding set of document rankings $\mathcal{R}$, MRR is defined as

$$\text{MRR}(\mathcal{R}, Q) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i}, \tag{3.7}$$

where $rank_i$ denotes the position of the first relevant document in the ranking $R$ and $\frac{1}{rank_i}$ denotes the reciprocal rank. In Chapter 7, we use the mMARCO dataset (Bonifacio et al., 2021), for which the official evaluation metric is MRR@10 (Kantor and Voorhees, 2000). MRR@10 evaluates the reciprocal rank only up to the first ten documents found in a ranking $R = [d_1, d_2, \ldots, d_{10}]$, and relevant documents appearing at later positions are assigned a reciprocal rank of zero.

| Language | Class | Family | Branch | CLEF | mMARCO |
|---|---|---|---|---|---|
| *High-resource Languages* | | | | | |
| English (EN) | 5 | Indo-European | Germanic | ✓ | ✓ |
| German (DE) | 5 | Indo-European | Germanic | ✓ | ✓ |
| Finnish (FI) | 4 | Uralic | Baltic-Finnic | ✓ | |
| Italian (IT) | 4 | Indo-European | Romance | ✓ | ✓ |
| Russian (RU) | 4 | Indo-European | Balto-Slavic | ✓ | ✓ |
| Dutch (NL) | 4 | Indo-European | Germanic | ✓ | |
| Turkish (TR) | 4 | Turkic | Oghuz | ✓ | |
| Arabic (AR) | 5 | Afro-Asiatic | Semitic | | ✓ |
| Chinese (ZH) | 5 | Sino-Tibetan | Sinitic | | ✓ |
| French (FR) | 5 | Indo-European | Romance | | ✓ |
| Portuguese (PT) | 4 | Indo-European | Romance | | ✓ |
| Vietnamese (VT) | 4 | Austronesian | Vietic | | ✓ |
| *Low-resource Languages* | | | | | |
| Indonesian (ID) | 3 | Austronesian | Malayic | | ✓ |
| Swahili (SW) | 2 | Niger-Congo | Sabaki | ✓ | |
| Somali (SO) | 1 | Afro-Asiatic | Cushitic | ✓ | |
| Kyrgyz (KG) | 1 | Turkic | Kipchak | ✓ | |
| Uyghur (UG) | 1 | Turkic | Karluk | ✓ | |

**Table 3.1:** Overview of CLIR languages used. The class column refers to the language taxonomy proposed by Joshi et al. (2020b), who distinguish between languages "exceptionally limited resources" (**0**), languages with "some amount of unlabeled data" (**1**), languages with "a small set of labelled datasets" (**2**), languages "with a strong web presence" (**3**), languages with "large amount of unlabelled data" and those with lesser amounts of data (**4**) and languages with "a dominant online presence" (**5**).

**Evaluation Benchmarks.** We evaluate resource-lean transfer methods for ad-hoc document-level CLIR primarily on the standard test collections from the CLEF 2001-2003 benchmark (Braschler, 2002, 2003, 2004).[11] We additionally experiment on the low-resource query languages Swahili and Somali provided by Bonab et al. (2019) and, as an additional contribution, we also published three new query translations into Uyghur, Kyrgyz and Turkish (Chapter 8). With CLEF, we have access to (1) natural, i.e. human-annotated relevance labels and human-written documents and queries, and (2) a dataset that facilitates evaluation in both monolingual retrieval in different languages (MoIR) and across languages (CLIR). Following standard practice (Lavrenko et al., 2002; Vulić and Moens, 2015), we create queries by concatenating the title and description field of each CLEF "topic" (Figure 3.3), and the title and text fields in documents (Figure 3.2).

[11]Many of the CLIR collections in CLEF, TREC, NTCIR, and FIRE have strict licensing constraints and are not publicly accessible (Galuščáková et al., 2021).

| | | Rel. Doc. | | Document Length | | Query Length | |
|---|---|---|---|---|---|---|---|
| | Corpus size | Average | Median | Whitespace | WordPiece | Whitespace | WordPiece |
| EN | 169.5K | 18.6 | 7.0 | 509.1 | 698.4 | 18.8 | 22.4 |
| DE | 294.8K | 32.6 | 24.0 | 283.5 | 488.9 | 17.2 | 28.4 |
| IT | 157.6K | 15.9 | 8.0 | 298.3 | 480.8 | 21.3 | 30.3 |
| FI | 55.3K | 10.7 | 5.0 | 256.0 | 646.7 | 12.8 | 34.8 |
| RU | 16.7K | 5.4 | 3.0 | 258.3 | 555.8 | 16.9 | 34.8 |
| TR | - | - | | - | - | 14.6 | 31.3 |
| SW | - | - | | - | - | 21.5 | 40.3 |
| SO | - | - | | - | - | 17.6 | 43.0 |
| KG | - | - | | - | - | 14.8 | 45.3 |
| UG | - | - | | - | - | 15.6 | 58.2 |

**Table 3.2:** CLEF 2003 dataset statistics of document collections and queries. We report the total number of documents (corpus size), the average/median number of relevant documents per query and the average/median number of tokens after whitespace and WordPiece tokenization (we use the pre-trained tokenizer of mBERT (Devlin et al., 2019)). Swahili and Somali queries are provided by (Bonab et al., 2019), and Kyrgyz and Uyghur are provided by us (see Chapter 8).

In Table 3.1 we list all languages used in this thesis and show their language resource categorization according to Joshi et al. (2020b).[12] Following Yong et al. (2023), we group languages belonging to resource category 4 and 5 into high-resource languages (top half). Deviating from their classification, we consider all other languages as low-resource languages (bottom half). In summary, our experiments involve twelve high-resource and five low-resource languages from a diverse set of seven languages families. Table 3.2 summarizes token frequency statistics of the CLEF 2003 dataset (Braschler, 2004). CLEF contains 60 queries, which are manually translated into document languages. The number of documents in CLEF range from 16.7K (RU) to 294.8K (DE). Notably, we find that WordPiece tokenization leads to disproportionately more tokens for Finnish and Russian documents and low-resource queries.[13] This reflects the fact that some languages are under-represented in the pre-training corpus on which the tokenizer was trained (cf. Chapter 2.3). It also indicates that these languages are less favored in IR because models need to allocate a larger token budget to encode the same queries.

For our experiments in Chapter 7, we require a large-scale in-domain training dataset to train our ranking models. While the CLEF datasets are sufficiently large and realistic for CLIR evaluation, they are too small to train supervised neural ranking models. We therefore resort to the mMARCO benchmark (Bonifacio et al., 2021), which consists of parallel (i.e., machine translated) queries and documents in thirteen different languages.

[12] See also https://microsoft.github.io/linguisticdiversity/

[13] In Appendix B, we show token-level distributions of queries and documents.

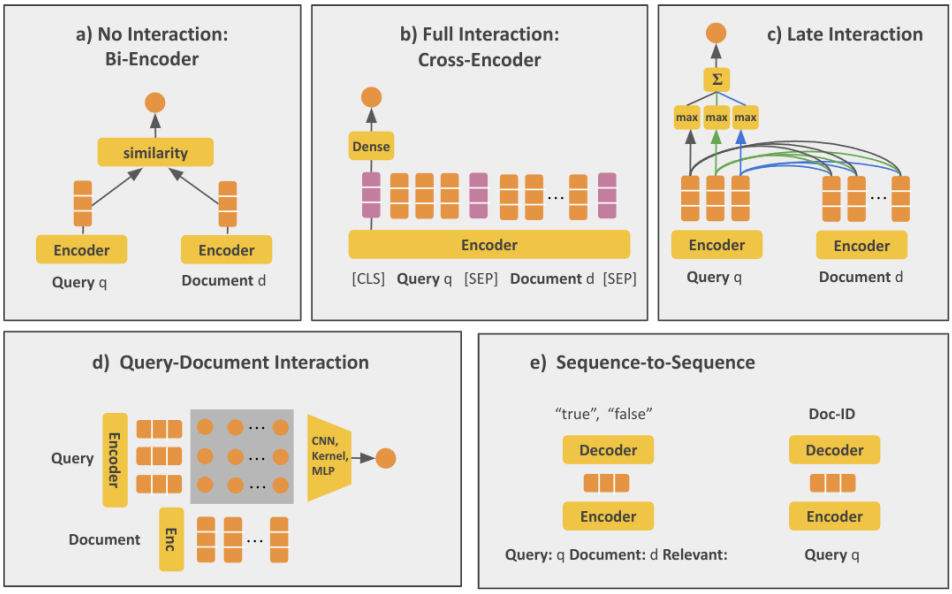


**Figure 3.4:** Overview of Neural IR paradigms: a) Bi-Encoder models encode queries and documents independently and compute relevance scores with vector similarity measures. b) Cross-Encoder models jointly represent query-document pairs and model relevance as a classification problem. Here, the encoder enables full interaction in all layers. c) Late interaction models pre-compute document token representations offline and query-document interactions on the last layer online. d) Models construct "image" representations from query-document token-level interactions and higher-level interactions with convolutional neural networks. e) Encoder-decoder models predict relevance based on token generation probabilities. This image adapted from (Khattab and Zaharia, 2020).

## 3.4 Neural Retrieval Paradigms

Neural retrieval models are based on semantic representations of query and document (sub)words. As shown in Figure 3.4, different approaches can be broadly classified by the way how interactions between query and document tokens are calculated and how relevance scores are predicted. Most retrieval models have in common that they are composed of (1) a token-level representation lookup, followed by (2) computing higher-level semantic representations from token-level interactions. In the following, adopt the classification of (Khattab and Zaharia, 2020) and review neural retrieval paradigms. Due to their relevance in the following chapters, we focus on bi-encoder and cross-encoder models. Our overview is limited to neural retrievers. We acknowledge the existence of other neural approaches such as query and document expansion models (Nogueira et al., 2019c,a; Formal et al., 2021; Gospodinov et al., 2023, *inter alia*) and refer the reader to (Guo et al., 2020; Lin et al., 2022) for a more comprehensive overview of existing methods.

### 3.4.1 No Interaction: Bi-Encoders

Bi-encoder models are also known as dual encoders (Karpukhin et al., 2020; Ni et al., 2022), dense retrieval models (Lin et al., 2022), or Siamese networks (Bromley et al., 1993; Kenter et al., 2016) when two encoders share the same weights. In this approach, queries $q$ and documents $d$ are projected (i.e., embedded) independently of each other into fixed-sized low-dimensional vector representations $\overrightarrow{q}, \overrightarrow{d} \in \mathbb{R}^d$. The relevance score for a given a document is computed with a vector space similarity measure $s^{BE} = sim(\overrightarrow{q}, \overrightarrow{d})$ such as cosine similarity (see Figure 3.4 a)). Bi-encoders compute semantic query/document representations either (1) with a non-parametric aggregation function such as average pooling (Le and Mikolov, 2014; Kenter et al., 2016; Galke et al., 2017, *inter alia*),[14] or (2) with a trained encoder model that computes *intra-token interactions* (Reimers and Gurevych, 2019; Karpukhin et al., 2020; Xiong et al., 2021; Qu et al., 2021). In Transformer-based encoders, these interactions are computed with attention mechanism and summarized in `[CLS]`-embeddings (see Section 2.3). We now briefly discuss two popular implementations of Transformer-based bi-encoders.

Karpukhin et al. (2020) present the dense passage retrieval (DPR) model, which is trained with a contrastive training objective to learn `[CLS]`-embeddings to minimize (maximize) the distance between queries and relevant (non-relevant) documents. For this, the authors use different question answering (QA) datasets and train DPR on random in-batch negative examples (i.e., non-relevant documents), hard negatives mined with BM25 (negative examples with a high lexical overlap), and positive passages (relevant documents). Later, Thakur et al. (2021) showed that DPR performs well when tested on in-domain data and substantially worse than BM25 when it is applied on out-of-domain datasets (zero-shot domain transfer).[15] Another popular dense retriever is Contriever (Izacard et al., 2021). Contrary to DPR, it is trained in a self-supervised way without any human-labeled data. To obtain positive query-document training pairs, the authors apply two random cropping strategies. First, following the Inverse Cloze Task (Lee et al., 2019), the authors extract random spans as queries and use their complement (i.e., text outside the span) as relevant documents. To include positive examples where queries and document can have overlapping tokens (i.e., exact lexical matches), the authors additionally sample contiguous overlapping text segments. Negative examples are obtained by mixing positive pairs from different examples. Izacard et al. (2021) train Contriever on web data and Wikipedia and show that it outperforms both DPR and BM25. Both DPR and Contriever have multilingual variants, dubbed mContriever (Izacard et al., 2021) and mDPR (Zhang et al., 2021, 2023c).

Bi-encoders do not compute inter-token interactions between query and document tokens. Encoding all semantic information in a fixed-sized vector of limited capacity is a bottleneck when we want to encode long documents such as news articles covering multiple topics (Tran et al., 2024). Instead of only considering

[14] We refer the reader to (Mitra and Craswell, 2017) for a more thorough review.

[15] This is in line with our results on zero-shot cross-lingual transfer for CLIR (see Chapter 6).

the first $n$ document tokens (Karpukhin et al., 2020), one can also (1) encode local parts of documents extracted with a sliding window (Hofstätter et al., 2020b) or (2) increase the representational capacity by encoding documents with multiple embeddings, each capturing different local aspects (Zhang et al., 2022b; Kong et al., 2022). For example, Zhang et al. (2022b) replace the `[CLS]` token by multiple `[Viewer]` tokens. The embeddings of these tokens are optimized to encode different aspects of the input text. Here, the similarity between a query and a document is measured as the similarity between the query and a document's most similar `[Viewer]` token. To avoid different `[Viewer]` tokens to collapse to the same representation during training, the authors use a uniformity loss to incentivize dissimilarity between the most similar viewer token (i.e., the embedding closest to the query embedding) and other viewer tokens.[16]

Despite their limited "capacity" bi-encoders are still a popular approach because document embeddings can be pre-computed and stored offline. Online retrieval can be run in a very efficient way thanks to fast approximate nearest neighbor libraries (Johnson et al., 2019). Because of this, bi-encoders are especially used as first-stage retrievers (Nogueira et al., 2019b), also known as prerankers. Multilingual bi-encoders are effective prerankers in CLIR, because they allow for efficient retrieval in setups where queries and documents are expected to have little to no lexical overlap.

### 3.4.2 Full Interaction: Cross-Encoders

Nogueira and Cho (2019) are the first to adopt BERT (Devlin et al., 2019) for IR. Their model, dubbed MonoBERT, frames predicting the relevance of a document as a binary classification task and can therefore be considered an instance of pointwise learning-to-rank (Liu, 2009, p.33). MonoBERT represents query-document pairs by concatenating their tokens into a single input sequence $Concat(q, d) =$ `[CLS] q [SEP] d [SEP]`. It uses the contextualized embedding of the special classification token $E_{\texttt{[CLS]}}$ at the output layer to jointly encode query-document pairs with a single dense feature representation, which is then used to predict the document relevance in a final classification layer.

$$s^{\text{CE}}(q, d) = \text{softmax}(E_{\texttt{[CLS]}} \cdot W + b) \tag{3.8}$$

The forward pass of MonoBERT's Transformer layers enables *full interaction* because all query tokens can attend all other query tokens as well as all document tokens, and vice versa. According to Lin et al. (2022), models that (1) follow the input representation template of concatenating queries and documents, as described above, and (2) combine Transformers with a classification head are called

[16] Li et al. (2023b) distinguish between single-vector retrieval and multi-vector retrieval. Here, we consider multi-view models as bi-encoder models, as there is no token-level interaction between queries and documents involved during the computation of relevance signals.

*cross-encoders* (CE) (MacAvaney et al., 2019; Nogueira and Cho, 2019). Cross-encoders are inherently limited by the maximum input sequence length the underlying BERT-based encoder model can encode. To mitigate this limitation, cross-encoder variants such as BERT-MaxP and BERT-SumP (Dai and Callan, 2019), and BIRCH (Yilmaz et al., 2019) compute passage-level or sentence-level relevance scores, which are then aggregated into document-level relevance scores. PARADE (Li et al., 2023a) takes a different approach and instead aggregates passage-level `[CLS]` representations with pooling mechanisms (e.g., max pooling or average pooling) or with model-based aggregation. In Section 5.5.3, we investigate relevance score aggregation for cross-lingual retrieval.

While cross-encoders are more expressive than bi-encoders they are also much slower due to the quadratic complexity of the attention mechanism (cf. Section 2.3). To balance retrieval speed and retrieval effectiveness, Nogueira et al. (2019b) propose to use MonoBERT in a multi-stage system as a slower but more effective reranker that is used to refine (i.e., rerank) the ranking of the top-$k$ documents returned by an efficient first-stage retriever.[17] Other approaches to improve the retrieval speed include knowledge distillation from large cross-encoders into smaller cross-encoders (Chen et al., 2021) or into bi-encoder models (see Section 3.4.4), and early stopping in lower layers (Xin et al., 2020).

### 3.4.3 Late Interaction: ColBERT

Late interaction models (Khattab and Zaharia, 2020; Gao et al., 2021a; Qian et al., 2022, *inter alia*) strike a balance between the efficiency of bi-encoders and expressiveness of cross-encoders. These models pre-compute and store document token representations at indexing time. At search time, rather than computing full interaction in all layers, they compute query-document token interactions only on representations extracted from the output layer.

The first late interaction model, dubbed ColBERT, was proposed by Khattab and Zaharia (2020). ColBERT uses a BERT-based encoder to first contextualize query and document tokens independently of each other. Here, input sequences are prepended by special tokens `[Q]` and `[D]`. Queries shorter than a pre-defined number of tokens are additionally padded with `[mask]` tokens, which function as a soft version of query augmentation (Khattab and Zaharia, 2020). To reduce the index size, the authors project query and document tokens embeddings into smaller vectors. ColBERT uses a non-parametric function to compute inter-token interactions and relevance scores (MaxSim). More precisely, MaxSim implements *late interaction* and identifies for each query token its most similar document token according to the cosine similarity between their respective embeddings. In other words, the relevance score is computed as the sum of "best match"-similarities across all query tokens. Formally, for a sequence of length-normalized (L2 vector norm) and contextualized query tokens $Q = [Q_1, Q_2, ..., Q_N]$ and document to-

[17] We revisit multi-stage ranking in Chapter 6.

kens $D = [D_1, D_2, ..., D_M]$, a document's relevance score $s^{\text{ColBERT}}(q, d)$ is computed as the sum of maximum similarities:

$$s^{\text{ColBERT}}(q, d) = \sum_{i=1}^{N} \max_{j=1}^{M} Q_i \cdot D_j^T \tag{3.9}$$

Khattab and Zaharia (2020) show that ColBERT can be used directly as a reranker and, importantly, that the MaxSim operator can also be used to directly identify for each query token its best matching token in the entire corpus. This allows ColBERT to be also deployed as a first stage ranker. Similar to bi-encoders, MaxSim can be computed efficiently on the entire corpus by using fast approximate nearest neighbor search libraries such as FAISS (Johnson et al., 2017). Late interaction can also be implemented with attention models (Gao et al., 2020). However, the ability to utilize FAISS makes ColBERT a more efficient choice.

Several studies focus on (1) reducing ColBERTs index size, (2) reducing its query latency and (3) improving the effectiveness of late interaction (Gao et al., 2021a; Hofstätter et al., 2022; Santhanam et al., 2022). For example, ColBERTer (Hofstätter et al., 2022) learns transformations from subword token embeddings to word-level embeddings to reduce the index size. COIL (Gao et al., 2021a) computes relevance scores as the sum of (i) interactions between exact lexical matches between query and document tokens and (ii) the dot product between their respective `[CLS]`-embeddings. Between computing late interaction on all query-document tokens on one hand and only on exact lexical matches on the other hand, the CITADEL model (Li et al., 2023b) frames computing interactions between query and document tokens as a dynamic lexical routing problem. Here, a routing model predicts alignments that encode which query-document token pairs interact. In a different work, Qian et al. (2022) propose to relax the constraint that each query token can only be matched to a single document token. Their model, dubbed ALIGNER, constructs interaction matrices from pairwise dot products, and controls which tokens get aligned with a learned sparse alignment matrix. Finally, ColBERT-PRF combines ColBERT's token-level embeddings and k-means clustering for pseudo-relevance feedback (Wang et al., 2021b).

### 3.4.4 Other Approaches

**Query-Document Interaction Models.** Early neural ranking models rely on (pre-trained) word embeddings and compute input representations from local pair-wise interactions between query and document tokens (Pang et al., 2016; Guo et al., 2016; Hui et al., 2017, 2018; Xiong et al., 2017). For example, MatchPyramid (Pang et al., 2016) and PACCR (Hui et al., 2017, 2018) model relevance prediction analogous to image classification. These models construct "images" where each "pixel" corresponds to, e.g., the dot product similarity between a query-document word embedding pair. Higher-level interactions such as matching phrases are captured with convolutional neural networks (CNN) (LeCun and Bengio, 1995). Different 2D CNN kernels capture different n-gram similarity patterns. MatchPyramid

and PACCR aggregate local matches into feature vectors that are used to compute relevance scores. Later, MacAvaney et al. (2019) show that word embeddings can be replaced with contextualized BERT embeddings.

**Generation-based IR.** Raffel et al. (2020) propose a unified transfer learning paradigm that casts different types of NLP task into sequence-to-sequence (seq2seq) tasks. The authors train a single encoder-decoder model, dubbed "Text-to-Text Transfer Transformer" (T5), on the union of training instances from a diverse set of NLP tasks. Nogueira et al. (2020) adopt this approach for IR. Their model, dubbed MonoT5, is similar to MonoBERT in the sense that both models are pointwise learning-to-rank models (Liu, 2009, p.33). Nogueira et al. (2020) use the template "Query: {q} Document: {d} Relevant:" to create training instances. The authors initialize MonoT5 with a T5 checkpoint and train the model to predict the next token to be "true" and "false" for relevant and non-relevant documents. To obtain a document's relevance score $s^{\text{monoT5}}$, the authors compute the softmax function only over the vocabulary tokens "true" and "false" and use the probability assigned to the former. A shortcoming of MonoT5 is that it can only be used for re-ranking because each document needs to be scored individually.

Tay et al. (2022) introduced with the *Differentiable Search Index (DSI)* model a novel and fully end-to-end retrieval paradigm. DSI exploits the fact that large language models can memorize factual knowledge in their model parameters (Bansal et al., 2022; Mallen et al., 2022; Carlini et al., 2023). The idea of DSI is to "store" (i.e. memorize) a corpus during indexing and decode document IDs from query text at retrieval time. *Indexing* is cast as a seq2seq task, where a DSI model learns to associate the content of a document to its document ID. That is, the indexing task is to predict document IDs from their content. To represent document IDs, Tay et al. (2022) investigate (1) atomic identifiers, (2) tokenizable IDs and (3) semantic IDs obtained from hierarchical k-means clustering. *Retrieval* is formulated as another seq2seq task. Here, the model learns to match queries to (relevant) document IDs. At test time, DSI ranks documents by sorting softmax logits of (atomic) document IDs or with beam-search decoding (tokenizable) document IDs. Follow-up work focuses on improving the representation of document IDs (Wang et al., 2022b), effectively generating query-document pairs for training DSI models (Zhuang et al., 2022; Bevilacqua et al., 2022) and different self-supervised pre-training tasks (Chen et al., 2022a). In summary, DSI is different from the classic "index-retrieve-then-rank" paradigm (Chen et al., 2022a) and learns the entire pipeline in an end-to-end fashion (Metzler et al., 2021).

**Hybrid Approaches.** Different paradigms excel at different aspects of retrieval. Prior work (Thakur et al., 2021; Chen et al., 2022b) shows that dense retrieval models trained on general domain data struggle when applied on out-of-domain data, a setting to which lexical models are robust and perform competitively. Sparse-dense hybrid approaches address this challenge, e.g., by linearly interpolating relevance

scores from sparse and dense rankers (Wang et al., 2021a; Luan et al., 2021), fusing rankings produced by sparse and dense rankers (Cormack et al., 2009; Chen et al., 2022b), or by guiding the training of dense rankers to explicitly capture signals that BM25 fails to capture (Gao et al., 2021c). On the architecture side, Zhang et al. (2023a) propose a hybrid between cross-encoders and late interaction models and show improvements on out-of-domain generalization while maintaining constant in-domain performance. In their model, the final relevance score is computed as the sum of MonoBERT's relevance score and an additional late interaction relevance score. In another hybrid approach, Zhang et al. (2022a) use a cross-encoder ranker and bi-encoder retriever in an "Adversarial Retriever-Ranker" (AR2) setup to gradually improve the ranker by training on increasingly more difficult negative examples provided by the retriever.

Finally, several studies use knowledge distillation (KD) (Hinton et al., 2015) to train a student bi-encoder model to imitate a slower but more expressive teacher model (Hofstätter et al., 2020a; Lin et al., 2021c; Izacard and Grave, 2021). For example, TCT-ColBERT (Lin et al., 2021c) uses ColBERT as a teacher model to distil its late interaction features into bi-encoder student models. TRMD-ColBERT (Choi et al., 2021) follow a two ranker multi-teacher KD approach. In the context of open-domain question answering (see Section 3.1.3), Izacard and Grave (2021) propose to inform the retriever with knowledge encoded in the model that is used to extract answers (i.e., the reader model). In this work, the authors distil cross-attention scores from a sequence-to-sequence model to a bi-encoder.

## 3.5 Conclusion

In this chapter, we provide an overview of the Cranfield evaluation paradigm and historical IR test collections (Section 3.1). In Section 3.2, we discussed the main challenges that arise in CLIR due to the lexical gap and lack of resources. We introduced our evaluation protocol (Section 3.3) and described datasets, metrics and baselines which we use throughout the rest of this thesis. Finally, we reviewed different neural retrieval paradigms and retrieval model architectures (Section 3.4). In the following chapters, we investigate the bi-encoder paradigm based on cross-lingual word embeddings (Chapter 4) and multilingual sentence encoders (Chapter 5). We then focus on resource-lean transfer of multilingual zero-shot cross-encoders in Chapters 6 to 8.

# Part II

# Resource-Lean Transfer of Bi-Encoders

# Chapter 4

# Cross-Lingual Retrieval with Static Word Embeddings

[1]In this chapter, we propose a fully unsupervised framework for ad-hoc cross-lingual information retrieval (CLIR), requiring no cross-lingual supervision and no CLIR task supervision at all. The framework leverages shared cross-lingual word embedding spaces (CLWE) in which terms, queries, and documents can be represented in a cross-lingually aligned semantic embedding space. We specifically experiment with CLWE induction methods introduced in Section 2.2. In Section 4.2, we introduce BoW-Agg, which is a CLIR model that follows the bi-encoder paradigm under the bag-of-words assumption. BoW-Agg encodes queries and documents by aggregating their constituent CLWEs. In the second model, we use CLWEs to perform a term-by-term query translation (TbT-QT) using cross-lingual nearest neighbors. In Section 4.4.1, we first conduct pilot experiments to validate the effectiveness of our CLIR models. In Section 4.4.2, we then provide a comprehensive comparative evaluation of projection-based, i.e. resource-lean CLWE induction models on word-level, sentence-level and document-level cross-lingual retrieval. Our findings show that resource-lean CLWE-based CLIR models perform well and outperform a lexical baseline. Compared to our resource-intensive CLIR baseline, where we use machine translation to translate queries, CLWE-based CLIR models fall behind.

[1]This chapter is adapted from: (1) **Robert Litschko**, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. 2018. Unsupervised crosslingual information retrieval using monolingual data only. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, pages 1253–1256, (2) **Robert Litschko**, Goran Glavaš, Ivan Vulić, and Laura Dietz. 2019. Evaluating resource-lean cross-lingual embedding models in unsupervised retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1109–1112.

## 4.1 Introduction

Traditional symbolic information retrieval models such as tf-idf (Sparck Jones, 1972) rely on lexical signals to match queries with documents based on overlapping terms. As discussed in Section 3.2, whenever there is a lack of word overlap between queries and documents, i.e. a *lexical gap* (Berger et al., 2000), those models fail to accurately estimate relevance. This can occur for many reasons, including cultural language variation (soccer vs. football) or language variation due to paraphrasing and the use of synonyms. To mitigate this problem, researchers have developed query (document) expansion models that augment queries (documents) with additional keywords (Lee et al., 2008; Nogueira et al., 2019a,c). Moving away from symbolic retrieval methods and towards semantic representation-based models allows retrieval to learn distributional word representations or, as they are more commonly referred to, word embeddings that map words to a semantic vector space where words with similar meanings have similar representations (Chapter 2).

In Cross-lingual Information Retrieval (CLIR) the *lexical gap between languages* is caused by mismatching vocabularies (Levow et al., 2005; Nie, 2010), shared words between languages are mostly limited to numerals and named entities. Researchers have soon broadened the work on word vector representations to cross-lingual word embeddings (CLWEs; see also Section 2.2). CLWE models typically operate on the word-level and induce a single *shared cross-lingual vector space*, so that representations of word translations are additionally aligned between languages. In this way, CLWEs provide a way of knowledge transfer between languages, thereby facilitating cross-lingual NLP (Klementiev et al., 2012; Hermann and Blunsom, 2014; Guo et al., 2015; Zhang et al., 2016; Heyman et al., 2017a, *inter alia*) and IR applications (Vulić and Moens, 2015) in a straightforward fashion. In CLIR, a shared cross-lingual space can (1) serve as an aligned input space for end-to-end neural ranking models (Guo et al., 2016; Mitra et al., 2017; MacAvaney et al., 2020b, *inter alia*), or (2) be used to construct semantic representations of queries and documents from embedding spaces for unsupervised cross-lingual retrieval, which is the focus of this chapter.

**Contributions.** Our key contributions and findings are summarized as follows:

**(1)** We present two CLIR models based on cross-lingual word embeddings. Our first model, dubbed BoW-Agg, follows the bi-encoder paradigm (Section 3.4) and encodes queries and documents independently into their respective embeddings by aggregating their constituent CLWEs.

**(2)** Our second model uses CLWEs to find nearest cross-lingual word neighbors for individual query terms. Those are then used in a term-by-term query translation (TbT-QT) approach to translate the query into the document language, which is then followed by a query likelihood retrieval model (Section 3.3).

**(3)** We benchmark both models on word-, sentence- and document-level cross-

lingual retrieval. Our results show that CLWEs are an effective and resource-lean way to bridge the cross-lingual language gap, as they require no direct task-level supervision and only limited cross-lingual supervision (bilingual dictionaries of 5k word pairs).

**Resource-Lean Transfer.** Referring to our taxonomy introduced in Chapter 1 (Figure 1.4), transferring CLIR models using only CLWEs fall under *unsupervised CLIR* branch since they require no (CL)IR training data (i.e., direct task supervision). To obtain CLIR models in different languages, we require (1) monolingual language data in the target languages and (2) minimal cross-lingual supervision in the form of bilingual dictionaries. Since bilingual dictionaries can be obtained in an unsupervised way (Section 2.2.3), our approaches are *fully unsupervised* and therefore suitable for scaling CLIR to a large number of languages.

**Non-Parametric Bi-Encoders Based on CLWEs.** As mentioned in Section 3.5, the notion of bi-encoders is typically used for parametric models (Karpukhin et al., 2020; Thakur et al., 2021; Izacard et al., 2021). Lin et al. (2022, p. 139) characterize bi-encoders along two criteria, they need to (1) represent queries and documents into a fixed-sized vectors and (2) use similarity measures to perform retrieval nearest neighbor search. We adopt this definition and refer to our models that aggregate token embeddings (Section 4.2.1 and Section 5.2) as *non-parametric versions of bi-encoders*.

## 4.2 Unsupervised Cross-lingual Retrieval Models

Our models rely on pre-trained CLWEs discussed in Section 2.2. With the induced cross-lingual spaces we can directly measure semantic similarity between words from the query and document language, but we still need to define how to represent queries and documents. In the following, we outline two models that use pre-trained cross-lingual embedding spaces for CLIR tasks.

### 4.2.1 Bag-of-Words Aggregation

In the first approach, dubbed BoW-Agg, we derive the cross-lingual embeddings of queries and documents by aggregating the cross-lingual embeddings of their constituent terms. Let $\overrightarrow{t}$ be the embedding of the term t, obtained from the cross-lingual embedding space and let $d = \{t_1, t_2, ..., t_{N_d}\}$ be a document from the collection consisting of $N_d$ terms. The embedding of the document d in the shared space can then be computed as:

$$\overrightarrow{d} = \overrightarrow{t_1} \circ \overrightarrow{t_2} \circ \cdots \circ \overrightarrow{t_{N_d}} \tag{4.1}$$

where $\circ$ is a semantic composition operator: it aggregates constituent term embeddings into a document embedding.[1] We opt for vector addition as composition for two reasons: 1) word embedding spaces exhibit linear linguistic regularities and 2) addition displays robust performance in compositional and IR tasks. A representation of the query vector $\overrightarrow{q}$ is then computed as the sum of embeddings of its constituent terms, disregarding their word order:

$$\overrightarrow{q} = \sum_{i=1}^{N_q} = \overrightarrow{t_i^q} \tag{4.2}$$

To obtain document representations, we compare two aggregation functions. First, we experiment with a simple non-weighted addition akin to how we compute query embeddings (BoW-Agg-Add). We also experiment with weighted addition where each term's embedding is weighted with the term's inverse document frequency (BoW-Agg-IDF):

$$\overrightarrow{d} = \sum_{i=1}^{N_d} = idf(t_i^d) \cdot \overrightarrow{t_i^d} \tag{4.3}$$

BoW-Agg-IDF relies on the common assumption that not all terms equally contribute to the document meaning: it emphasizes vectors of more document specific terms.[2] Finally, we compute the relevance score simply as the cosine similarity between query and document embeddings in the shared cross-lingual space:

$$rel_{Agg}(q,d) = \frac{\overrightarrow{q} \cdot \overrightarrow{d}}{||\overrightarrow{q}|| \cdot ||\overrightarrow{d}||} \tag{4.4}$$

Notably, Galke et al. (2017) and Zhang et al. (2018) also investigate aggregating monolingual word embeddings for IR. In BoW-Agg, however, we *first align monolingual embedding spaces* and then aggregate CLWEs for unsupervised CLIR.

### 4.2.2 Term-by-Term Query Translation

Our second CLIR model, Term-by-Term Query Translation (TbT-QT), exploits the cross-lingual word embedding space in a different manner: it performs a term-by-term translation of the query into the language of the document collection relying solely on the shared cross-lingual space. Each source language query term

[1] There is a large number of options for the composition operator, ranging from unsupervised operations like addition and element-wise multiplication (Mitchell and Lapata, 2008) to complex parametrized (e.g., tensor-based) composition functions (Milajevs and et al, 2014). We discard the parametrized composition functions because they require parameter optimization through supervision, and we are interested in *fully unsupervised CLIR*.

[2] Note that with both variants of BoW-Agg, we effectively ignore both query and document terms that are not represented in the cross-lingual embedding space.

$t^q$ is replaced by the target language term $tr(t^q)$, that is, its cross-lingual nearest neighbor in the embedding space. The cosine similarity is used for computing cross-lingual semantic similarities of terms. In other words, the query $q' = \{tr(t_1^q), tr(t_2^q), \ldots, tr(t_{N_q}^q), \}$ in $L_T$.

By effectively transforming a CLIR task into a monolingual IR task, we can apply any of the traditional IR ranking functions designed for sparse text representations. We opt for the ubiquitous query likelihood model (Ponte and Croft, 1998), which we introduce in Section 3.3. We additionally apply smoothing on the unigram language model (LM-UN) of individual documents with the unigram language model of the entire collection, using the Dirichlet smoothing scheme (Zhai and Lafferty, 2004):

$$rel_{TbT}(q', d) = \Pi_{i=1}^{N_{q'}} \lambda \cdot P(t_i^{q'}|d) + (1 - \lambda) \cdot P(t_i^{q'}|D) \tag{4.5}$$

$P(t_i^{q'}|d)$ is the maximum likelihood estimate (MLE) of term $t_i^{q'}$ appearing in document $d$, $P(t_i^{q'}|D)$ is the MLE of term's probability based on the target collection $D$, and $\lambda = N_d/(N_d + \mu)$ determines the ratio between the contributions of the local and global language model, with $N_d$ being the document length and $\mu$ the parameter of Dirichlet smoothing. TbT-QT is conceptually similar to query expansion methods proposed in (Roy et al., 2016, 2018), where monolingual embedding spaces are used to expand queries with k-nearest neighbors extracted from query terms. TbT-QT and BoW-Agg, when combined with unsupervised cross-lingual word embeddings, represent a fully unsupervised CLIR framework.

## 4.3 Experimental Setup

Our experimental design is two-fold. In a pilot study, we first validate the effectiveness of our CLWE-based models on a smaller set of languages (Section 4.4.1, and then provide a large-scale evaluation of resource-lean CLWE models on word-level, sentence-level and document-level retrieval (Section 4.4.2).

**CLIR Datasets and Language Pairs.** First, we benchmark BoW-Agg and TbT-QT on ad-hoc document-level retrieval on a reduced set of three language pairs of the CLEF 2001–2003 benchmarks (Braschler, 2002, 2003, 2004): EN→NL, EN→IT, EN→FI.[3] We then expand our evaluation and also include word-level and sentence-level cross-lingual retrieval. We evaluate our models on six languages of varying language proximity - English (EN), German (DE), Italian (IT), Finnish (FI), Dutch (NL) and Russian (RU). Specifically, we experiment with the following nine language pairs in CLEF 2003: EN→{DE, FI, IT, RU}, DE→{FI, IT, RU}, and FI→{IT, RU}. For sentence-level CLIR evaluation, we resort to the parallel Europarl corpus (Koehn, 2005). Since Europarl does not contain Russian translations, we evaluate sentence-level CLIR on the remaining six language pairs. For

[3]Finnish was included to CLEF evaluation only in 2002 and 2003.

each language pair we randomly sample 1K "queries" (i.e., source language sentences) and 100K "documents" (i.e., target language sentences). Given a sentence in the source language, an ideal CLIR model would rank its mate sentence (i.e., its translation) in the target language on top. That is, in this setting there is only one relevant "document" per "query".

**CLWE Training Data.** We use pre-trained 300-dimensional FASTTEXT vectors (Bojanowski et al., 2017b)[4] for CLWE models that build on top of monolingual word embeddings. For our pilot experiments (Table 4.1) we trained BWESG embeddings on full document-aligned Wikipedias[5] using SGNS with suggested parameters from prior work (Vulić and Moens, 2016): 15 negative samples, a global decreasing learning rate of 0.025 and a window size of 16. PROC embeddings of Smith et al. (2017) are trained with 10K translation pairs obtained from Google Translate. The training setup for MUSE follows closely the default setup of Lample et al. (2018): we refer the reader to the original paper and the model implementation accessible online for more information and technical details.[6] For our main experiments (Table 4.2, Table 4.3 and Table 4.4) we obtained dictionaries for supervised CLE models by translating 7K most frequent English words to the other four languages via Google Translate. For each language pair, we split the dictionaries into 5K pairs for training[7] and 2K pairs for word-level CLIR evaluation on bilingual lexicon induction (BLI) (Irvine and Callison-Burch, 2017).

**Models and Baselines.** We evaluate different CLIR models, obtained by combining models for inducing cross-lingual word vector spaces discussed in Section 2.2 - CCA, PROC, PROC-B, RCSLS, VECMAP, MUSE, ICP, BWESG - with each of the two ranking models BoW-Agg and TbT-QT. For the latter we compute the relevance score $rel_{TbT}$ with a Dirichlet smoothing parameter value of $\mu = 1000$ (Zhai and Lafferty, 2004). In our pilot experiments we additionally evaluate an ensemble ranker that combines the two ranking functions: BoW-Agg-IDF and TbT-QT. If $r_1$ is the rank of document $d$ for query $q$ according to the TbT-QT model and $r_2$ is the rank produced by BoW-Agg-IDF, the ensemble ranker ranks the documents in increasing order of the scores $\lambda \cdot r_1 + (1 - \lambda) \cdot r_2$. This approach is similar to Reciprocal Rank Fusion proposed by Cormack et al. (2009). We evaluate ensembles with values $\lambda = 0.5$ and $\lambda = 0.7$, i.e., with more weight allocated to the more powerful TbT-QT model (see Table 4.1). We compute the results of CLWE-based models to two baselines: (1) a monolingual unigram language model (LM-UNI;

[4] https://fasttext.cc/docs/en/pretrained-vectors.html

[5] http://linguatools.org/tools/corpora/wikipedia-comparable-corpora/

[6] https://github.com/facebookresearch/MUSE

[7] We use all 5K pairs to train all supervised models except Proc-B, for which we use training dictionaries of only 1K word translation pairs. This is because we want to evaluate whether the bootstrapping procedure can compensate for less bilingual supervision.

| | | EN→NL | | | EN→ IT | | | EN→ FI | | |
|---|---|---|---|---|---|---|---|---|---|---|
| CLWE | Model | 2001 | 2002 | 2003 | 2001 | 2002 | 2003 | 2002 | 2003 | AVG |
| – | LM-UNI | .122 | .204 | .145 | .087 | .161 | .128 | .096 | .138 | .135 |
| BWESG | BoW-Agg-Add | .111 | .138 | .137 | .087 | .115 | .147 | .027 | .085 | .106 |
| | BoW-Agg-IDF | .144 | .202 | .188 | .127 | .157 | .188 | .083 | .125 | .152 |
| | TbT-QT | **.296** | **.309** | .344 | **.292** | **.306** | .342 | **.182** | .167 | **.280** |
| | Ensemble ($\lambda = 0.5$) | .236 | .296 | .316 | .255 | .260 | .314 | .169 | .187 | .254 |
| | Ensemble ($\lambda = 0.7$) | .255 | .309 | .326 | **.272** | .278 | .333 | **.173** | .205 | .269 |
| PROC | BoW-Agg-Add | .149 | .168 | .203 | .138 | .155 | .236 | .078 | .217 | .168 |
| | BoW-Agg-IDF | .185 | .196 | .243 | .169 | .166 | .248 | .086 | .204 | .187 |
| | TbT-QT | .241 | .268 | .314 | .234 | .286 | .328 | .140 | .182 | .249 |
| | Ensemble ($\lambda = 0.5$) | .258 | .296 | .327 | .232 | .271 | .317 | .136 | .259 | .262 |
| | Ensemble ($\lambda = 0.7$) | .257 | .300 | .331 | .240 | **.281** | .328 | .149 | .261 | .268 |
| MUSE | BoW-Agg-Add | .150 | .181 | .218 | .140 | .166 | .249 | .103 | **.259** | .183 |
| | BoW-Agg-IDF | .198 | .224 | .268 | .178 | .193 | .276 | .118 | .246 | .213 |
| | TbT-QT | .268 | .298 | **.359** | .244 | .272 | **.354** | .147 | .239 | .273 |
| | Ensemble ($\lambda = 0.5$) | .278 | .318 | .354 | .241 | .269 | **.356** | .157 | **.319** | .286 |
| | Ensemble ($\lambda = 0.7$) | **.279** | **.323** | **.361** | .244 | .275 | .354 | .163 | .312 | **.289** |

**Table 4.1:** CLIR performance on all three test language pairs for all models in comparison (MAP scores). Best (ensemble) models are highlighted in bold.

i.e., without query translation) as a sanity check baseline;[8] (2) a much stronger baseline (MT-IR) translates the query to the collection language using a full-blown MT model and then performs monolingual retrieval using LM-UNI. In contrast to CLWE-based CLIR, our MT-IR baseline is more resource-demanding as it requires large sentence-aligned corpora.

## 4.4 Results and Discussion

In Section 4.4.1 we first validate the effectiveness of BoW-Agg-{Add, IDF} and TbT-QT in isolation and as ensembles on three language pairs. We then experiment with seven resource-lean CLWE induction methods (i.e. mapping based methods relying on bilingual dictionaries) on nine language pairs in Section 4.4.2.

### 4.4.1 Unsupervised Retrieval using Monolingual Data Only

Table 4.1 shows the performance of all models on all test collections, reported in terms of the standard mean average precision (MAP) measure (Section 3.3).

[8]Relying on lexical overlap between the query and documents, LM-UNI is bound to perform poorly in CLIR where the query language differs from the collection language.

**CLIR based on Unsupervised vs. Supervised CLWEs.** First, apart from BoW-Agg-Add (with BWESG CLWEs), all models outperform the lexical baseline LM-UNI. This demonstrates the effectiveness of our CLIR models in bridging the language gap across languages. The weighted variant of BoW-Agg (BoW-Agg-IDF) outperforms the simpler non-weighted summation model (BoW-Agg-Add) across the board. These results suggest that the common IR assumption about document-specific terms being more important than the terms occurring collection-wide is also valid for constructing dense document representations by summing word embeddings. BoW-Agg models based on PROC embeddings (the bilingual signal is word translation pairs) outperform models based on BWESG (requiring document-aligned data) on average. This is an encouraging finding, as word translations pairs are easier to obtain than document-aligned comparable corpora. Most importantly, the unsupervised MUSE + TbT-QT CLIR model displays peak performance close to BWESG, which requires comparable data and is thus more resource-intensive than MUSE (*fully unsupervised*). We find this to be a very important result: it shows that we can perform robust CLIR without any cross-lingual information, that is, by relying purely on monolingual data.

**Ensemble Models.** Ensembles generally outperform the best-performing individual CLIR models, and for some test collections (e.g., EN→NL 2002, EN→FI 2003) by a wide margin. Between the two interpolation factors, $\lambda = 0.5$ and $\lambda = 0.7$, we find that the latter yields consistently stronger results than the former (except for EN→FI 2003). This is not surprising, since the single models TbT-QT also outperforms BoW-Agg-IDF. Notably, our ensemble method is a specific instance of a broader class of rank fusion methods (Kurland and Culpepper, 2018). The individual effectiveness of TbT-QT and BoW-Agg-IDF and the observation that our ensembles improve upon both models indicate that they encode complementary relevance signals. This is intuitive as the former matches queries against documents in the lexical space while the latter uses semantic representations.

**Language Similarity and Aggregation.** The results in Table 4.1 imply that the proximity of CLIR languages plays a role only to a certain extent. Most models do exhibit lower performance for EN→FI than for the other two language pairs: this is expected since Finnish is lexically and typologically more distant from English than Italian and Dutch. However, even though NL is linguistically closer to EN than IT,[9] we find mixed results. Overall, the results for TbT-QT and BoW-Agg are higher when the document language is NL. A closer inspection reveals that the results for EN→IT are higher in six out of nine cases on the 2003 portion of CLEF, whereas on the remaining two portions, we find EN→NL consistently yields better results (except for PROC TbT-QT on the 2002 portion of CLEF).

[9] Both NL and EN are Germanic languages, while Italian is a Romanic language (see Section 3.3).

| | DE→X | | | EN→X | | | | FI→X | | |
|---|---|---|---|---|---|---|---|---|---|---|
| CLWE Model | FI | IT | RU | DE | FI | IT | RU | IT | RU | AVG |
| CCA | .353 | .506 | .411 | .542 | .383 | .624 | .454 | .353 | .340 | .441 |
| PROC | .359 | .510 | .425 | .544 | .396 | .625 | .464 | .355 | .342 | .447 |
| PROC-B | .354 | .507 | .392 | .521 | .360 | .605 | .419 | .328 | .315 | .422 |
| RCSLS | **.395** | **.529** | **.458** | **.580** | **.438** | **.652** | **.510** | **.388** | **.376** | **.481** |
| VECMAP | **.302** | .493 | **.322** | **.521** | **.292** | .600 | **.323** | **.355** | **.312** | **.391** |
| MUSE | .000 | **.496** | .272 | .520 | .000 | **.608** | .000 | .000 | .001 | .211 |
| ICP | .251 | .447 | .245 | .486 | .262 | .577 | .259 | .263 | .231 | .336 |

**Table 4.2:** BLI performance of different CLWE models.

### 4.4.2 Resource-Lean Cross-Lingual Embedding Models

We now compare the CLIR performance of seven different resource-lean CLWE methods on nine different languages. This excludes BWSG since it relies on comparable corpora, while other CLWE methods are mapping-based and use bilingual dictionaries.

**Word Translation Results.** We examine how word translation performance of CLWE models relates to their CLIR performance in Table 4.2. We first intrinsically evaluate BLI performance on 2K test dictionaries, in terms of mean reciprocal rank (MRR). Not surprisingly, the RCSLS model with a BLI-tailored objective exhibits the best word translation performance. Simple projection models – CCA and PROC – also exhibit solid performance and the bootstrapping-based model PROC-B, trained using only 1K pairs, does not lag behind by much. Unsupervised CLWE models, among which VECMAP (Artetxe et al., 2018) performs best, despite recent claims (Lample et al., 2018; Artetxe et al., 2018), do not match the performance of their supervised competitors.

**CLIR Results.** In Table 4.3 we show CLIR results at the document level (CLEF dataset; MAP), and in Table 4.4 we summarize sentence-level CLIR performance (Europarl dataset; MRR) of CLWE-based CLIR models. The scores in the upper half of both tables correspond to the embedding aggregation model (BoW-Agg-IDF), and the scores in the lower half are obtained with the term-by-term query translation model (TbT-QT). In both CLIR evaluations, we find that TbT-QT and BoW-Agg perform comparably. Interestingly, on document-level CLIR, on six out of nine language pairs, the best results are achieved with TbT-QT, whereas in sentence-level CLIR, in five out of six language pairs, the best-performaning model is BoW-Agg-IDF. This supports the intuition that semantic matching with CLWEs is robust towards lexical variations, but less effective for long and topically diverse documents. Conversely, TbT-QT fully relies on lexical matches, but it does not suffer (as much) from topical diversity. Both TbT-QT and BoW-Agg outperform the unigram (LM-UNI) model, validating the results in our pilot experiments.

| | | DE→X | | | EN→X | | | | FI→X | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | CLWE | FI | IT | RU | DE | FI | IT | RU | IT | RU | AVG |
| LM-UNI | – | .113 | .137 | .001 | .139 | .138 | .128 | .001 | .130 | .001 | .088 |
| MT-IR | – | .338 | .431 | .238 | .384 | .282 | .428 | .265 | .406 | .261 | .337 |
| BoW-Agg-IDF | CCA | .251 | .210 | .158 | .249 | .193 | .243 | .151 | .145 | .146 | .194 |
| | PROC | .255 | .212 | .152 | .261 | .200 | .240 | .152 | .149 | .146 | .196 |
| | PROC-B | **.295** | .230 | .155 | .288 | **.259** | .265 | .166 | .151 | .136 | **.216** |
| | RCSLS | .196 | .189 | .122 | .237 | .127 | .211 | .133 | .130 | .113 | .162 |
| | ICP | .252 | .170 | .167 | .230 | .230 | .231 | .119 | .117 | .124 | .182 |
| | MUSE | .001 | .210 | **.195** | .280 | .000 | .272 | .002 | .002 | .001 | .107 |
| | VECMAP | .240 | .129 | .162 | .200 | .150 | .201 | .104 | .096 | .109 | .154 |
| TbT-QT | CCA | .159 | .243 | .133 | **.290** | .150 | .335 | .141 | .212 | .170 | .204 |
| | PROC | .160 | .242 | .098 | .283 | .147 | .325 | **.179** | **.215** | .183 | .204 |
| | PROC-B | .207 | **.267** | .099 | .286 | .154 | **.368** | .138 | .189 | .151 | .206 |
| | RCSLS | .119 | .186 | .146 | .262 | .117 | .261 | .156 | .185 | .086 | .169 |
| | ICP | .141 | .215 | .105 | .265 | .174 | .304 | .144 | .138 | .144 | .181 |
| | MUSE | .000 | .247 | .120 | .267 | .000 | .338 | .002 | .004 | .000 | .109 |
| | VECMAP | .278 | .241 | .099 | .278 | .149 | .259 | .145 | .194 | **.209** | .206 |

**Table 4.3:** Document-level CLIR results (CLEF).

Compared to the resource-intensive MT-IR baseline, CLWE-based models underperform in document and sentence retrieval. Sometimes the gap narrows down to a few MAP points (e.g., DE→RU and EN→FI on CLEF). Notably, the gaps are larger in sentence-level CLIR. This is expected since MT models are trained on parallel data such as Europarl.

Comparing different CLWE models, we observe that these CLIR results do not follow the trends observed in the BLI task. For example, the best-performing CLWE model on BLI, RCSLS, yields only mediocre CLIR results. This implies that overfitting CLWE models to word translation performance may hurt performance in downstream tasks such as CLIR. Furthermore, the PROC-B model, trained using only 1K word pairs, exhibits better CLIR performance than other supervised models (CCA, PROC, and RCSLS), trained on 5K word pairs. In sentence-level CLIR evaluation, the unsupervised VECMAP outperforms other embedding models on most languages. This is consistent with its strong BLI performance, where it performs best among unsupervised CLWE models, and expected, since both BLI and sentence-level CLIR are similar tasks (i.e., retrieval of translation pairs). The performance drops of MUSE between our pilot experiments and main experiments confirm its lack of robustness reported in (Glavaš et al., 2019).

Overall, we conclude that MT is a better option for languages where we have training data available, whereas the resource-lean CLWE models offer an effective and resource-lean solution for in scenarios where we have no parallel data to train MT models. As such, they are a viable alternative for transferring CLIR models to a large number of languages.

| Model | CLWE | DE→FI | DE→IT | EN→DE | EN→FI | EN→IT | FI→IT | AVG |
|---|---|---|---|---|---|---|---|---|
| LM-UNI | - | .047 | .071 | .073 | .043 | .078 | .036 | .058 |
| MT-IR | - | .543 | .693 | .726 | .661 | .804 | .698 | .688 |
| BoW-Agg-IDF | CCA | .132 | .309 | .390 | .135 | .492 | .122 | .263 |
| | PROC | .131 | .309 | .396 | .135 | .496 | .124 | .265 |
| | PROC-B | .162 | .341 | **.414** | .143 | .521 | .137 | .286 |
| | RCSLS | .121 | .301 | .350 | .117 | .438 | .136 | .244 |
| | ICP | .087 | .158 | .243 | .074 | .300 | .054 | .153 |
| | MUSE | .001 | .336 | .404 | .000 | .499 | .000 | .207 |
| | VECMAP | **.227** | **.360** | .306 | **.156** | .470 | **.204** | **.287** |
| TbT-QT | CCA | .104 | .336 | .363 | .113 | .537 | .171 | .270 |
| | PROC | .100 | .336 | .364 | .110 | .529 | .172 | .269 |
| | PROC-B | .112 | .344 | .358 | .105 | **.550** | .158 | .271 |
| | RCSLS | .091 | .313 | .321 | .097 | .471 | .160 | .242 |
| | ICP | .121 | .312 | .342 | .094 | .508 | .135 | .252 |
| | MUSE | .009 | .344 | .352 | .009 | .533 | .010 | .210 |
| | VECMAP | .138 | .334 | .362 | .106 | .528 | .182 | .275 |

**Table 4.4:** Sentence-level CLIR results (Europarl).

## 4.5 Conclusion

In this chapter, we presented a fully unsupervised CLIR framework that leverages unsupervised cross-lingual word embeddings induced solely on the basis of monolingual corpora. We show that our models are able to retrieve relevant content cross-lingually without any bilingual data at all, i.e., they exhibit competitive performance on standard CLEF CLIR evaluation data for all three test language pairs (pilot study). We also present a comprehensive evaluation study on the effectiveness of resource-lean models for inducing cross-lingual embedding spaces for cross-lingual retrieval. We show that the word translation (BLI) performance, on which resource-lean CWLE models are commonly evaluated, is a poor predictor for CLIR performance of the model. Our results also reveal that MT-based CLIR models outperform CLWE-based CLIR models by a large margin. However, their adoption is limited by the availability of large-scale parallel training data. In resource-lean scenarios, CLWE-based CLIR models are a viable solution, as demonstrated by their ability to outperform lexical baselines.

Importantly, compared to pre-trained language models (PLM) (Devlin et al., 2019; Conneau and Lample, 2019) and large language models (LLM) (OpenAI, 2023; Touvron et al., 2023), CLWEs have distinct advantages: (i) they require substantially less compute power and are therefore more sustainable, (ii) they do not suffer from the "curse of multilinguality" (Conneau et al., 2020), since each language has its own parameter budget, and (iii) models such as TbT-QT operate on the symbolic space and are therefore easier to interpret. We make our code and resources available at: https://github.com/rlitschk/UnsupCLIR

# Chapter 5

# Cross-Lingual Retrieval with Contextual Embeddings

[1]In Section 2.3.3, we discussed multilingual pre-trained language models (mPLM) which encode text as contextual embeddings in a shared embedding space. In this chapter, we investigate how effective multilingual spaces induced by mPLMs are for cross-lingual IR and, compared to static cross-lingual word embeddings (CLWE), to what extent contextualization impacts CLIR in a resource-lean scenario. In Section 5.2, we propose three mPLM-based CLIR models with different types of contextualization (*none*, *static* and *in-place contextualization*) and compare them against multilingual text encoders specialized for sentence-level similarity, i.e. multilingual sentence encoders (see Section 5.3.1). Our results on sentence- and document-level CLIR reveal that mPLMs fall behind static CLWEs and that the best results are achieved with a multilingual sentence encoder. We show that further gains can be obtained from localized relevance matching where queries are matched against document segments, which allow models to capture relevance signals beyond the supported maximum sequence length of mPLMs. Finally, in Section 5.5.5, we show that following a few-shot learning approach consistently improves the results of the best-performing multilingual sentence encoder.

## 5.1 Introduction

Cross-lingual information retrieval (CLIR) systems respond to queries in a source language by retrieving relevant documents in another, target language. Their success is typically hindered by data scarcity: they operate in challenging low-resource

[1]Adapted from: (1) **Robert Litschko**, Ivan Vulic, Simone Paolo Ponzetto, and Goran Glavaš. 2021. Evaluating multilingual text encoders for unsupervised cross-lingual retrieval. In *Advances in Information Retrieval: 43rd European Conference on IR Research (ECIR)*, pages 342–358, Lucca, Italy (Online); (2) **Robert Litschko**, Ivan Vulic, Simone Paolo Ponzetto, and Goran Glavaš. 2022. On cross-lingual retrieval with multilingual text encoders. *Information Retrieval Journal 25.2*, pages 149–183.

settings without sufficient labeled training data, i.e., human relevance judgments, to build reliable in-domain supervised models (e.g., neural matching models for pairwise retrieval (Yu and Allan, 2020; Jiang et al., 2020a)). In Chapter 4, we showed that language transfer by means of cross-lingual embedding spaces (CLWEs) can yield strong performance in a range of unsupervised ad-hoc CLIR setups. However, this approach, by limitations of static CLWEs, cannot capture and handle polysemy in the underlying text representations, and captures only "static" word-level semantics. *Contextual representations*, obtained from multilingual pre-trained language models (mPLM), alleviate this issue (Liu et al., 2020) because they encode occurrences of the same word differently depending on its context (Section 2.3).

Multilingual text encoders have already found applications in document-level CLIR. Jiang et al. (2020a) use mBERT as a matching model by feeding pairs of English queries and foreign language documents. MacAvaney et al. (2020b) use mBERT in a zero-shot setting, where they train a retrieval model on top of mBERT on English relevance data and apply it on a different language. However, prior work has not investigated unsupervised CLIR setups, and a systematic comparative study focused on the suitability of the multilingual text encoders for diverse ad-hoc CLIR tasks and language pairs is still lacking. In the current chapter, we address this research gap and study whether these general-purpose multilingual text encoders can be used directly for ad-hoc CLIR without any additional supervision (i.e., cross-lingual relevance judgments). To this end, we investigate whether they can outperform unsupervised CLIR approaches based on static CLWEs, and how they perform depending on the (properties of the) language pair at hand. We compare three classes of models: CLIR models based on representations extracted from mPLMs, multilingual sentence encoders (Section 2.3.3), and few-shot CLIR where we further fine-tune models on limited in-domain supervision.

We first study *unsupervised mPLM-backed CLIR models* and propose different "encoding variants" with varying degrees of contextualization (Section 5.2). The first variant completely ignores contextualization and extracts static word embeddings from mPLMs by encoding words in isolation (ISO). The second variant, AOC, aggregates for each word multiple contextual representations. The third variant, SEMB, creates in-place embeddings and directly encodes queries and documents with mPLMs. During retrieval, we follow the BoW-Agg approach introduced in Chapter 4. There exist many pre-trained models that can be used out of the box for cross-lingual fine-tuning. Here, we investigate two popular models, mBERT (Devlin et al., 2019) and XLM (Conneau and Lample, 2019), on CLEF document-level retrieval and Europarl sentence-level retrieval.

Moreover, with the rising interest in zero-shot or few-shot learning, i.e., training a multilingual encoder on a resource-rich language and applying it on a resource-scarce language with little to no extra training, much research has focused on developing *general-purpose multilingual sentence encoders* (Artetxe and Schwenk, 2019a; Yang et al., 2020b; Feng et al., 2022; Reimers and Gurevych, 2020). Consequently, besides mBERT and XLM, we additionally benchmark multilingual sentence encoders in our unsupervised CLIR setup. We focus on four popular off-

the-shelf sentence encoders, discussed in Section 5.3.1, to gauge their effectiveness in document-level and sentence-level CLIR. Those encoders are naturally designed for cross-lingual sentence retrieval and unable to encode long documents. We therefore experiment with a localized relevance matching approach in Section 5.5.3, where we follow a sliding window approach to represent each document with multiple embeddings corresponding to sentences or chunks.

Finally, we measure the performance gains obtained from further fine-tuning multilingual sentence encoders on little in-domain data under a *few-shot learning* setting. This simulates a scenario where limited annotation budget is available.

**Contributions.** Our key contributions and findings are summarized as follows:

**(1)** We empirically validate that, without any task-specific fine-tuning, multilingual encoders such as mBERT and XLM fail to outperform CLIR approaches based on static CLWEs (Sections 5.5.1 to 5.5.2). Their performance also crucially depends on how one encodes semantic information with the models (e.g., treating them as sentence/document encoders directly versus averaging over constituent words and/or subwords).

**(2)** We show that multilingual sentence encoders, i.e. mPLMs fine-tuned on labeled data from sentence pair tasks like natural language inference or semantic text similarity as well as using parallel sentences, substantially outperform general-purpose models (mBERT and XLM) in sentence-level CLIR (Section 5.5.2); further, they can be leveraged for localized relevance matching and in such a pooling setup improve the performance on unsupervised document-level CLIR (Section 5.5.3).

**(3)** In-domain fine-tuning of the best-performing unsupervised Transformer model (Reimers and Gurevych, 2020) (i.e., zero-shot language transfer, no domain transfer) – yields considerable gains over the original unsupervised ranker (Section 5.5.5). This renders fine-tuning with little in-domain data more beneficial than transferring models trained on large-scale out-of-domain datasets.

**Resource-Lean Transfer.** Our CLIR models based only on mPLMs (ISO, AOC, SEMB) do not use any direct task supervision (i.e., IR training data) and therefore fall into the class of *unsupervised CLIR methods* (cf. Chapter 1, Figure 1.4). We also benchmark multilingual sentence encoders, which are trained on sentence-similarity tasks (Section 5.3.1). From a retrieval perspective, those models are *transferred from related tasks*. Here, the task supervision is limited in a qualitative sense, i.e. sentence similarity models do not capture the same features as retrieval models. The task differences between semantic matching and relevance matching are also discussed in (Guo et al., 2016; Rao et al., 2019a). We finally quantify the performance difference of multilingual sentence encoders compared to their variants fine-tuned in a *few-shot setting*. Here, the supervision is limited in quantity, i.e. we assume availability of relevance annotations of a few queries.

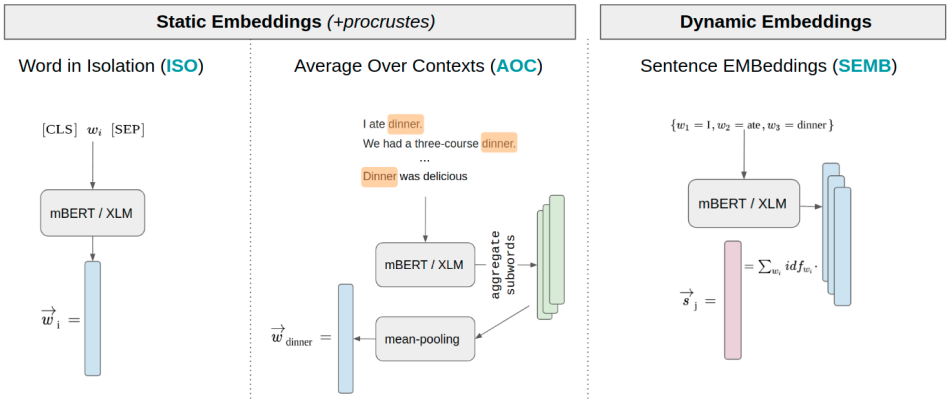


**Figure 5.1:** CLIR Models based on Multilingual Transformers. **Left:** Induce a static embedding space by encoding each vocabulary term in isolation; then refine the bilingual space for a specific language pair using the standard Procrustes projection. **Middle:** Aggregate different contextual representations of the same vocabulary term to induce static embedding space; then refine the bilingual space for a specific language pair using the standard Procrustes projection. **Right:** Direct encoding of a query-document pair with the multilingual encoder.

## 5.2 Unsupervised Cross-lingual Retrieval Models

Massively multilingual pre-trained neural language models such as mBERT and XLM(-R) can be used as a dynamic embedding layer to produce contextualized word representations, since they share a common input space on the subword level (e.g. word-pieces, byte-pair-encodings) across all languages (Section 2.3.3). Let us assume that a term (i.e., a word-level token) is tokenized into a sequence of $K$ subword tokens ($K \geqslant 1$; for simplicity, we assume that the subwords are word-pieces ($wp$)): $t_i = \{wp_{i,k}\}_{k=1}^{K}$. The multilingual encoder then produces contextualized subword embeddings for the term's $K$ constituent subwords $\overrightarrow{wp_{i,k}}$, $k = 1, \ldots, K$, and we can aggregate these subword embeddings to obtain the representation of the term $t_i$: $\overrightarrow{t_i} = \psi\left(\{\overrightarrow{wp_{i,k}}\}_{k=1}^{K}\right)$, where the function $\psi()$ is the aggregation function over the $K$ constituent subword embeddings. Once these term embeddings $\overrightarrow{t_i}$ are obtained, we follow the BoW-Agg-IDF approach to obtain query and document embeddings. We refer the reader to Section 4.2.1 for details. We now illustrate three encoding variants to obtaining word and sentence representations from pre-trained Transformers (Figure 5.1) and describe them in more detail in what follows.

### 5.2.1 Encoding Words in Isolation

In this approach, we obtain static word embeddings from multilingual language models. We first use mBERT and XLM in two different ways to induce static word embedding spaces for all languages. In a simpler variant, we feed terms

into the encoders *in isolation* (**ISO**), that is, without providing any surrounding context for the terms. This effectively constructs a static word embedding table similar to what is done with the CLWEs in Section 4.2.1, and allows the CLIR model to operate at a non-contextual word level. An empirical CLIR comparison between ISO and CLIR operating on traditionally induced CLWEs (Chapter 4) then effectively quantifies how well multilingual encoders (mBERT and XLM) capture word-level representations (Vulić et al., 2020). We specifically feed each term $t_i$ tokenized into its subwords $\{wp_{i,k}\}_{k=1}^{K}$ together with the special tokens into the encoder `[CLS]` $wp_{i,1} \dots wp_{i,K}$ `[SEP]` to obtain the terms' subword embeddings $\{\overrightarrow{wp_{i,k}}\}_{k=1}^{K}$. We then extract the contextualized embedding of $t$ via *mean-pooling*, i.e., by averaging of their constituent subword embeddings:

$$\psi(\{\overrightarrow{wp_{i,k}}\}_{k=1}^{K}) = \frac{1}{K}\sum_{k=1}^{K}\overrightarrow{wp_{i,k}} \tag{5.1}$$

As a result, we obtain for each term a $d$-dimensional term embedding where $d$ corresponds to the encoder models' hidden size. Following this approach, we encode all $|V|$ vocabulary terms of a given language $L_1$ and concatenate their term embeddings to form the embedding matrix $\mathbf{X}_{L_1} \in \mathbb{R}^{d \times |V|}$.

In a preliminary analysis we evaluated the static ISO (and AOC embeddings, see below) induced for different languages with multilingual encoders, on the bilingual lexicon induction (BLI) task (Glavaš et al., 2019). We observed poor BLI performance, suggesting that further projection-based alignment of respective monolingual embedding spaces is warranted. In other words, the obtained static embeddings, despite being induced with multilingual encoders, did not appear to be lexically well-aligned across languages, which consistent with (Cao et al., 2020). To strengthen the token-level alignment, we adopted the standard Procrustes method (Smith et al., 2017; Artetxe et al., 2018) for learning an orthogonal linear projection from the embedding (sub)space of one language to the embedding space of the other language (Glavaš et al., 2019). More precisely, for a given query language $L_1$ and document language $L_2$, we first encode their vocabularies into respective embedding matrices $\mathbf{X}_{L_1}$ and $\mathbf{X}_{L_2}$. Next, we project $L_1$ onto $L_2$ by using a learned projection matrix $\mathbf{W}$, which we obtain from the Procrustes method explained in Section 2.2.1. During retrieval we use query embeddings $\mathbf{X}_{L_1}\mathbf{W}$ and document embeddings $\mathbf{X}_{L_2}$ to encode them following the bag of word embedding approach (BoW-Agg-IDF; see Section 4.2.1).

### 5.2.2 Average-over-Context Embeddings

In the second, more elaborate variant we do leverage the contexts in which the terms appear, constructing *average-over-contexts* embeddings (**AOC**). For each term $t$ we collect a set of sentences $s_i \in \mathcal{S}_t$ in which the term $t$ occurs. We use the full set of Wikipedia sentences $\mathcal{S}$ to sample sets of contexts $\mathcal{S}_t$ for each vocabulary term $t$. For a given sentence $s_i$ let $j$ denote the position of $t$'s first

occurrence. We then transform $s_i$ with mBERT or XLM as the encoder, $enc(s_i)$, and, similar to ISO embeddings, extract the contextualized embedding of $t$ via *mean-pooling*, $\psi\left(\{\overrightarrow{wp_{j,k}}\}_{k=1}^{K}\right) = 1/K \cdot \sum_{k=1}^{K} \overrightarrow{wp_{j,k}}$. For each vocabulary term, we obtain $N_t = min(|\mathcal{S}_t|, \tau)$ contextualized vectors, with $|\mathcal{S}_t|$ as the number of Wikipedia sentences containing $t$ and $\tau$ as the maximal number of sentence samples for a term. The final static embedding of $t$ is then simply the average over the $N_t$ contextualized vectors:

$$\overrightarrow{t} = \frac{1}{N_t} \sum_{s_i \in \mathcal{S}_t}^{N_t} \psi\left(enc(s_i)_{j:j+K}\right) \tag{5.2}$$

We denote with $j{:}j{+}K$ the subsequence in sentence $s_i$ corresponding to the first occurrence of term $t$ consisting of $K$ subwords. Similar to ISO embeddings, we first form static embedding tables by encoding and concatenating vocabulary term embeddings, and then map, for each language pair, the query language embedding space to the document language.

### 5.2.3 Multilingual LMs as Sentence Embedders

In both AOC and ISO, we exploit the multilingual (contextual) encoders to obtain the static embeddings for word types (i.e., terms): we can then leverage these static word embeddings obtained from contextualized encoders in exactly the same ad-hoc CLIR setup (Section 4.2.1) in which CLWEs had previously been evaluated (Chapter 4). In an arguably more straightforward approach, we also use pre-trained multilingual Transformers (i.e., mBERT or XLM) to directly semantically encode the whole input text (**SEMB**). To this end, we encode the input text by averaging the contextualized representations of all terms in the text (we again compute the weighted average, where the terms' IDF scores are used as weights, see Section 4.2.1). For SEMB, we take the contextualized representation of each term $t_i$ to be the contextualized representation of its first subword token, i.e., $\overrightarrow{t_i} = \psi\left(\{\overrightarrow{wp_{i,k}}\}_{k=1}^{K}\right) = \overrightarrow{wp_{i,1}}$.[2]

## 5.3 Cross-Lingual Transfer with Limited Supervision

We now discuss two resource-lean transfer methods for CLIR. In Section 5.3.1, we first discuss multilingual sentence encoders, i.e. we transfer models from related tasks (limited supervision) to CLIR. We then discuss our few-shot CLIR approach where we specialize sentence encoders on in-domain data (Section 5.3.2).

[2] In our preliminary experiments taking the vector of the first term's subword consistently outperformed averaging vectors of all its constituent subwords.

### 5.3.1 Specialized Multilingual Sentence Encoders

Off-the-shelf multilingual Transformers (mBERT and XLM) have been shown to yield sub-par performance in unsupervised text similarity tasks; therefore, in order to be successful in semantic text (sentences or paragraph) comparisons, they first need to be fine-tuned on text matching (typically sentence matching) datasets (Reimers and Gurevych, 2020; Cao et al., 2020; Zhao et al., 2020b). Such encoders *specialized for semantic similarity* are supposed to encode sentence meaning more accurately, supporting tasks that require unsupervised (ad-hoc) semantic text matching. In contrast to off-the-shelf mBERT and XLM, which contextualize (sub)word representations, these models directly produce a semantic embedding of the input text. We provide a brief overview of the models included in our comparative evaluation.

**Language Agnostic SEntence Representations (LASER).** Artetxe and Schwenk (2019a) adopt a standard sequence-to-sequence architecture typical for neural machine translation (MT). It is trained on 223M parallel sentences covering 93 languages. The encoder is a multi-layered bidirectional LSTM and the decoder is a single-layer unidirectional LSTM. The 1024-dimensional sentence embedding is produced by max-pooling over the outputs of the encoder's last layer. The decoder then takes the sentence embedding as additional input at each decoding step. The decoder-to-encoder attention and language identifiers on the encoder side are deliberately omitted, so that all relevant information gets 'crammed' into the fixed-sized sentence embedding produced by the encoder. In our experiments, we directly use the output of the encoder to represent both queries and documents.

**Language-agnostic BERT Sentence Embeddings (LaBSE)** (Feng et al., 2022) is another neural dual-encoder framework, also trained with parallel data. Unlike LASER and m-USE, where the encoders are trained from scratch on parallel data, LaBSE starts its training from a pre-trained mBERT instance (i.e., a 12-layer Transformer network pre-trained on the concatenated corpora of 100+ languages). In addition to the multi-task training objective of m-USE, LaBSE additionally uses standard self-supervised objectives used in pre-training of mBERT and XLM: masked and translation language modeling (MLM and TLM, see Section 2.3.3). For further details, we refer the reader to the original work.

**Knowledge Distillation (DISTIL)** (Reimers and Gurevych, 2020) is a teacher-student framework for injecting the knowledge obtained through specialization for semantic similarity from a specialized monolingual Transformer (e.g., BERT) into a non-specialized multilingual Transformer (e.g., mBERT). It first specializes for semantic similarity a monolingual (English) teacher encoder $M$ using the available semantic sentence-matching datasets for supervision. In the second, *knowledge distillation* step a pre-trained multilingual student encoder $\widehat{M}$ is trained to mimic the output of the teacher model. For a given batch of sentence translation pairs

$\mathcal{B} = \{(s_j, t_j)\}$, the teacher-student distillation training minimizes the following loss:

$$\mathcal{J}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \left[ \left( M(s_j) - \widehat{M}(s_j) \right)^2 + \left( M(s_j) - \widehat{M}(t_j) \right)^2 \right] \tag{5.3}$$

The teacher model $M$ is Sentence-BERT (Reimers and Gurevych, 2019), BERT specialized for embedding sentence meaning on semantic text similarity (Cer et al., 2017) and natural language inference (Williams et al., 2018) datasets. The teacher network only encodes English sentences $s_i$. The student model $\widehat{M}$ is then trained to produce for both $s_j$ and $t_j$ the same representation that $M$ produces for $s_j$. Minimizing the distance between $M(s_j)$ and $\widehat{M}(s_j)$ encourages the student to learn semantic knowledge encoded by the teacher model, and minimizing the distance between $M(s_j)$ and $\widehat{M}(t_j)$ aligns it across language boundaries. We benchmark different DISTIL models in our CLIR experiments, with the student $\widehat{M}$ initialized with different multilingual Transformers.

**Multilingual Universal Sentence Encoder (m-USE)** is a general-purpose sentence embedding model for transfer learning and semantic text retrieval tasks (Yang et al., 2020b). It relies on a standard dual-encoder neural framework (Chidambaram et al., 2019; Yang et al., 2019a) with shared weights, trained in a multi-task setting with an additional translation bridging task. For more details, we refer the reader to the original work. There are two pre-trained m-USE instances available – we opt for the 3-layer Transformer encoder with average-pooling.

### 5.3.2 In-Domain Contrastive Fine-Tuning

**Contrastive Metric-Based Learning.** In most ad-hoc retrieval setups, one at best has a handful of relevance judgments for the test collection of interest. One approach in such low-supervision settings is to use the few available relevance judgments to reshape the representation space of (multilingual) text encoders. In this so called bi-encoder paradigm, the objective is to bring representations of queries, produced independently by the pre-trained encoder, closer to the representations of their relevant documents (produced again independently by the same encoder) than to the representations of irrelevant documents. In Section 4.1, we defined non-parametric bi-encoders as models which encode queries and documents with a parameter-free aggregating function such as mean-pooling. Here, we assume limited task supervision to train parametric bi-encoders (henceforth, bi-encoders). The objectives of contrastive metric-based learning push the instances that stand in a particular relation (e.g., query and *relevant* document) closer together according to a predefined similarity or distance metric (e.g., cosine similarity) than corresponding pairs that do not stand in the relation of interest (e.g., the same query and some irrelevant document). It is precisely the approach used for obtaining multilingual encoders specialized for sentence similarity tasks covered in Section 5.3.1 (Reimers and Gurevych, 2019; Feng et al., 2022; Yang et al., 2020b).

We propose to use contrastive metric-based learning to fine-tune the representation space for the concrete ad-hoc retrieval task, using a limited amount of relevance judgments available for the target collection. To this end, we employ a popular contrastive learning objective referred to as Multiple Negative Ranking Loss (MNRL) (Thakur et al., 2020). Given a query vector $q_i$, a relevant document $d_i^+$ and a set of in-batch negatives $\{d_{i,j}^-\}_{j=1}^m$ we fine-tune the parameters of a pre-trained multilingual encoder by minimizing MNRL, given as:

$$\mathcal{L}\left(q_i, d_i^+, \{d_{i,j}^-\}_{j=1}^m\right) = -\log \frac{e^{\lambda \cdot \text{sim}(q_i, d_i^+)}}{e^{\lambda \cdot \text{sim}(q_i, d_i^+)} + \sum_{j=1}^m e^{\lambda \cdot \text{sim}(q_i, d_{i,j}^-)}} \tag{5.4}$$

Each document, the relevant $d_j^+$ and each of the irrelevant $d_{i,j}^-$, receives a score that reflects their similarity to the query $q_i$: for this, we rely on cosine similarity, i.e. $\text{sim}(q_i, d_j) = \cos(q_i, d_j)$. Document scores, scaled with a temperature factor $\lambda$, are then converted into a probability distribution with a softmax function. The loss is then, intuitively, the negative log likelihood of the relevant document $d_j^+$. In Section 5.5.5, we fine-tune in this manner the best-performing multilingual encoder on document-level CLIR (Section 5.5.1).

## 5.4 Experimental Setup

**Evaluation Data.** We follow the experimental setup outlined in Section 4.3 and compare the models from Section 5.2 on language pairs comprising five languages: English (EN), German (DE), Italian (IT), Finnish (FI) and Russian (RU). Specifically, for document-level retrieval we run experiments for the following nine language pairs: EN-{FI, DE, IT, RU}, DE-{FI, IT, RU}, FI-{IT, RU}. Following our previous experiments, we use the 2003 portion of the CLEF benchmark (Braschler, 2004), see Section 3.3 for details. For sentence-level retrieval, we use the same experimental setup as in Section 4.3. That is, for each language pair we sample from Europarl (Koehn, 2005) 1K source language sentences as queries and 100K target language sentences as the "document collection".[3]

**Baseline Models.** To establish whether multilingual encoders outperform CLWEs in a fair comparison, we compare their performance against the strongest CLWE-based CLIR model from Section 4.4, dubbed Proc-B. Proc-B induces a bilingual CLWE space from pre-trained monolingual FASTTEXT embeddings[4] using the linear projection computed as the solution of the Procrustes problem given the dictionary of word-translation pairs. Compared to simple Procrustes mapping, Proc-B iteratively (1) augments the word translation dictionary by finding mutual nearest

[3] Russian is not included in Europarl and we therefore exclude it from sentence-level experiments. Further, since some multilingual encoders have not seen Finnish data in pre-training, we additionally report the results over a subset of language pairs that do not involve Finnish.

[4] https://fasttext.cc/docs/en/pretrained-vectors.html

neighbors and (2) induces a new projection matrix using the augmented dictionary. The final bilingual CLWE space is then plugged into the BoW-Agg-IDF model from Section 2.2.

Our document-level retrieval SEMB models do not get to see the whole document but only the first 128 word-piece tokens. For a more direct comparison, we therefore additionally evaluate the Proc-B baseline (PROC-$B_{LEN}$) which is exposed to exactly the same amount of document text as the multilingual XLM encoder (i.e., the leading document text corresponding to first 128 word-piece tokens). Finally, we compare CLIR models based on multilingual Transformers to a baseline relying on machine translation baseline (MT-IR). In MT-IR, 1) we translate the query to the document language using Google Translate and then 2) perform monolingual retrieval using a standard Query Likelihood Model (Ponte and Croft, 1998) with Dirichlet smoothing (Zhai and Lafferty, 2004).

**Model Details.** For all multilingual encoders we experiment with different input sequence lengths: 64, 128, 256 subword tokens. For AOC we collect (at most) $\tau = 60$ contexts for each vocabulary term: for a term not present at all in Wikipedia, we fall back to the ISO embedding of that term. We also investigate the impact of $\tau$ in Section 5.5.4. In all cases (SEMB, ISO, AOC), we surround the input with the special sequence start and end tokens of respective pre-trained models: `[CLS]` and `[SEP]` for BERT-based models and $\langle s \rangle$ and $\langle / s \rangle$ for XLM-based models. For vanilla multilingual encoders (mBERT and XLM) and all three variants (SEMB, ISO, AOC), we independently evaluate representations from different Transformer layers (cf. Section 5.5.4). For comparability, for ISO and AOC – methods that effectively induce static word embeddings using multilingual contextual encoders – we opt for exactly the same term vocabularies used by the Proc-B baseline, namely the top 100K most frequent terms from respective monolingual fastText vocabularies. We additionally experiment with different instances of the DISTIL model: $\text{DISTIL}_{\text{XLM-R}}$ initializes the student model with the pre-trained XLM-R Transformer (Conneau et al., 2020); $\text{DISTIL}_{\text{USE}}$ instantiates the student as the pre-trained m-USE model (Yang et al., 2020b); whereas $\text{DISTIL}_{\text{DistilmBERT}}$ distils the knowledge from the Sentence-BERT teacher into a multilingual version of DistilBERT (Sanh et al., 2019), a 6-layer Transformer pre-distilled from mBERT.[5] For SEMB models we scale embeddings of special tokens (sequence start and end tokens, e.g., `[CLS]` and `[SEP]` for mBERT) with the mean IDF value of input terms.

## 5.5 Results and Discussion

We first discuss our main results of document- and sentence-level retrieval, followed by localized relevance matching, further analysis and few-shot CLIR.

[5] Working with mBERT directly instead of its distilled version led to similar scores, while increasing running times.

| | EN→X | | | | DE→X | | | FI→X | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | FI | IT | RU | DE | FI | IT | RU | IT | RU | AVG | w/o FI |
| *Baselines* | | | | | | | | | | | |
| MT-IR | .276 | **.428** | .383 | **.263** | **.332** | **.431** | .238 | **.406** | .261 | **.335** | **.349** |
| PROC-B | .258 | .265 | .166 | .288 | .294 | .230 | .155 | .151 | .136 | .216 | .227 |
| PROC-B$_{\text{LEN}}$ | .165 | .232 | .176 | .194 | .207 | .186 | .192 | .126 | .154 | .181 | .196 |
| *Models based on multilingual Transformers* | | | | | | | | | | | |
| SEMB$_{\text{XLM}}$ | .199* | .187* | .183 | .126* | .156* | .166* | .228 | .186* | .139 | .174 | .178 |
| SEMB$_{\text{mBERT}}$ | .145* | .146* | .167 | .107* | .151* | .116* | .149* | .117 | .128* | .136 | .137 |
| AOC$_{\text{XLM}}$ | .168 | .261 | .208 | .206* | .183 | .190 | .162 | .123 | .099 | .178 | .206 |
| AOC$_{\text{mBERT}}$ | .172* | .209* | .167 | .193* | .131* | .143* | .143 | .104 | .132 | .155 | .171 |
| ISO$_{\text{XLM}}$ | .058* | .159* | .050* | .096* | .026* | .077* | .035* | .050* | .055* | .067 | .083 |
| ISO$_{\text{mBERT}}$ | .075* | .209 | .096* | .157* | .061* | .107* | .025* | .051* | .014* | .088 | .119 |
| *Similarity-specialized sentence encoders (with parallel data supervision)* | | | | | | | | | | | |
| DISTIL$_{\text{FILTER}}$ | .291 | .261 | .278 | .255 | .272 | .217 | .237 | .221 | .270 | .256 | .250 |
| DISTIL$_{\text{XLM-R}}$ | .216 | .190* | .179 | .114* | .237 | .181 | .173 | .166 | .138 | .177 | .167 |
| DISTIL$_{\text{USE}}$ | .141* | .346* | .182 | .258 | .139* | .324* | .179 | .104 | .111 | .198 | .258 |
| DISTIL$_{\text{DistilmBERT}}$ | **.294** | .290* | **.313** | .247* | .300 | .267* | **.284** | .221* | **.302*** | .280 | .280 |
| LaBSE | .180* | .175* | .128 | .059* | .178* | .160* | .113* | .126 | .149 | .141 | .127 |
| LASER | .142 | .134* | .076 | .046* | .163* | .140* | .065* | .144 | .107 | .113 | .094 |
| m-USE | .109* | .328* | .214 | .230* | .107* | .294* | .204 | .073 | .090 | .183 | .254 |

**Table 5.1:** Document-level CLIR results (Mean Average Precision). **Bold**: best model for each language-pair. *: difference in performance w.r.t. PROC-B significant at $p < 0.05$, computed via paired two-tailed t-test with Bonferroni correction.

### 5.5.1 Document-Level CLIR Results

We show the performance (MAP) of multilingual encoders on document-level CLIR tasks in Table 5.1. The first main finding is that none of the self-supervised models (mBERT and XLM in ISO, AOC, and SEMB variants) outperforms the CLWE baseline PROC-B. However, the full PROC-B baseline has, unlike mBERT and XLM variants, been exposed to the full content of the documents. A fairer comparison, against PROC-B$_{\text{LEN}}$, which has also been exposed to the same amount of text,[6] reveals that SEMB and AOC variants come reasonably close, albeit still do not outperform PROC-B$_{\text{LEN}}$. This suggests that the document-level retrieval could benefit from encoders able to encode longer portions of text, e.g., (Beltagy et al., 2020; Zaheer et al., 2020). For document-level CLIR, however, these models would first have to be ported to multilingual setups. Scaling embeddings by their *idf* (PROC-B) effectively filters out high-frequent terms such as stopwords. We therefore experiment with explicit a priori stopword filtering in DISTIL$_{\text{DistilmBERT}}$,

[6] Specifically, we tokenize queries/documents with the BPE tokenizer of XLM and then detokenize the first 128 subword tokens back into word tokens.

dubbed $\text{DISTIL}_{\text{FILTER}}$. Results show that performance deteriorates which indicates that stopwords provide important contextualization information. While SEMB and AOC variants exhibit similar performance, ISO variants perform much worse. The direct comparison between ISO and AOC demonstrates the importance of contextual information and seemingly limited usability of off-the-shelf multilingual encoders as word encoders, if no context is available, and if they are not further specialized to encode word-level information (Liu et al., 2021).

Similarity-specialized multilingual encoders, which rely on pre-training with parallel data, yield mixed results. Three models, $\text{DISTIL}_{\text{DistilmBERT}}$, $\text{DISTIL}_{\text{USE}}$ and m-USE, generally outperform the PROC-B baseline.[7] LASER is the only encoder trained on parallel data that does not beat the PROC-B baseline. We believe this is because (a) LASER's recurrent encoder provides text embeddings of lower quality than Transformer-based encoders of m-USE and DISTIL variants and (b) it has not been subjected to any self-supervised pre-training like DISTIL models. Even the best-performing CLIR model based on a multilingual encoder ($\text{DISTIL}_{\text{DistilmBERT}}$) overall falls behind the MT-based baseline (MT-IR). However, it is very important to note that the performance of MT-IR critically depends on the quality of MT for the concrete language pair: for language pairs with weaker MT (e.g., FI-RU, EN-FI, FI-RU, DE-RU), $\text{DISTIL}_{\text{DistilmBERT}}$ can substantially outperform MT-IR (e.g., 9 MAP points for FI-RU and DE-RU). In contrast, the gap in favor of MT-IR is, as expected, largest for the pairs of large typologically similar languages, for which also the most reliable MT systems exist: EN-IT, EN-DE. In other words, the feasibility and robustness of a strong MT-IR CLIR model seems to diminish with more distant language pairs and lower-resource languages.

The variation in results with similarity-specialized sentence encoders indicates that: (a) despite their seemingly similar high-level architectures typically based on dual-encoder networks (Cer et al., 2018), it is important to carefully choose a sentence encoder in document-level retrieval, and (b) there is an inherent mismatch between the granularity of information encoded by the current state-of-the-art text representation models and the document-level CLIR task.

### 5.5.2 Sentence-Level Cross-Lingual Retrieval

We show the sentence-level CLIR performance in Table 5.2. Unlike in document-level CLIR, self-supervised SEMB variants here manage to outperform PROC-B. The better relative SEMB performance than in document-level retrieval is somewhat expected: sentences are much shorter than documents (i.e., typically shorter than the maximal sequence length of 128 word pieces). All purely self-supervised mBERT and XLM variants, however, perform worse than the MT-IR baseline.

Multilingual sentence encoders specialized with parallel data excel in sentence-level CLIR, all of them substantially outperforming the competitive MT-IR baseline. This, however, does not come as much of a surprise, because these models

[7] As expected, m-USE and $\text{DISTIL}_{\text{USE}}$ perform poorly on language pairs involving Finnish, as they have not been trained on any Finnish data.

| | EN→FI | EN→IT | EN→DE | DE→FI | DE→IT | FI→IT | AVG | w/o FI |
|---|---|---|---|---|---|---|---|---|
| *Baselines* | | | | | | | | |
| MT-IR | .659 | .803 | .725 | .541 | .694 | .698 | .687 | .740 |
| Proc-B | .143 | .523 | .415 | .162 | .342 | .137 | .287 | .427 |
| *Models based on multilingual Transformers* | | | | | | | | |
| $\text{SEMB}_{\text{XLM}}$ | .309* | .677* | .465 | .391* | .495* | .346* | .447 | .545 |
| $\text{SEMB}_{\text{mBERT}}$ | .199* | .570 | .355 | .231* | .481* | .353* | .365 | .469 |
| $\text{AOC}_{\text{XLM}}$ | .099 | .527 | .274* | .102* | .282 | .070* | .226 | .361 |
| $\text{AOC}_{\text{mBERT}}$ | .095* | .433* | .274* | .088* | .230* | .059* | .197 | .312 |
| $\text{ISO}_{\text{XLM}}$ | .016* | .178* | .053* | .006* | .017* | .002* | .045 | .082 |
| $\text{ISO}_{\text{mBERT}}$ | .010* | .141* | .087* | .005* | .017* | .000* | .043 | .082 |
| Similarity-specialized sentence encoders (with parallel data supervision) | | | | | | | | |
| $\text{DISTIL}_{\text{XLM-R}}$ | .935* | .944* | .943* | .911* | .919* | .914* | .928 | .935 |
| $\text{DISTIL}_{\text{USE}}$ | .084* | .960* | .952* | .137 | .920* | .072* | .521 | .944 |
| $\text{DISTIL}_{\text{DistilmBERT}}$ | .847* | .901* | .901* | .811* | .842* | .793* | .849 | .882 |
| LaBSE | .971* | .972* | .964* | .948* | .954* | .951* | .960 | .963 |
| LASER | **.974*** | **.976*** | **.969*** | **.967*** | **.965*** | **.961*** | **.969** | **.970** |
| m-USE | .079* | .951* | .929* | .086* | .886* | .039* | .495 | .922 |

**Table 5.2:** Sentence-level CLIR results (MAP). **Bold**: best model for each language-pair. *: difference in performance with respect to Proc-B, significant at $p < 0.05$, computed via paired two-tailed t-test with Bonferroni correction.

(a) have been trained using parallel data (i.e., sentence translations), and (b) have been optimized exactly on the sentence similarity task. In other words, in the context of the cross-lingual sentence-level task, these models are effectively supervised models. The effect of supervision is most strongly pronounced for LASER, which was, being also trained on parallel data from Europarl, effectively subjected to in-domain training. We note that at the same time LASER was the weakest model from this group on average in the document-level CLIR task.

The fact that similarity-specialized multilingual encoders perform much better in sentence-level than in document-level CLIR suggests viability of a different approach to document-level retrieval: instead of obtaining a single encoding for the document, one may (independently) encode its sentences (or larger windows of content) and (independently) measure their semantic correspondence to the query. We investigate this *localized relevance matching* approach to document-level CLIR with similarity-specialized multilingual encoders in the next section.

### 5.5.3 Localized Relevance Matching

Contrary to most NLP tasks, in ad-hoc document retrieval we face the challenge of semantically representing long documents. According to Robertson and Walker (1994), documents can be viewed either as a concatenation of topically heteroge-

neous short sub-documents (*"Scope Hypothesis"*) or as a more verbose version of a short document on the same topic (*"Verbosity Hypothesis"*). Under both hypotheses, the source of relevance of the document for the query is localized, i.e., there should exist (at least one) segment (relatively short w.r.t. the length of the whole document) that is the source of relevance of the document for the query. Furthermore, a query may represent an information need on a specific aspect of a topic that is simply not discussed at the beginning, but rather somewhere later in the document: the maximum input sequence length imposed by neural text encoders directly limits the retrieval effectiveness in such cases. Even if we assume that we can encode the complete document with our multilingual encoders, these document representations would likely become semantically less precise (i.e., fuzzier) as they would aggregate contextualized representations of many more tokens; in Section 5.5.4 we validate this empirically and show that simply increasing the maximum sequence length of multilingual encoders does not improve their cross-lingual retrieval performance.

Recent work proposed pre-training procedures for encoding long documents (Dai et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020). These models have been pre-trained only for English. Pre-training their multilingual counterparts, however, would require extremely large and massively multilingual corpora and computational resources of the scale that we do not have at our disposal. In the following, we instead experiment with two resource-lean alternatives: we represent documents either as (1) sets of overlapping text *segments* obtained by running a sliding window over the document or (2) a collection of document *sentences*, which we then encode independently similar to (Akkalyoncu Yilmaz et al., 2019). For a single document, we now need to store multiple semantic representations (embeddings), one for each text segment or sentence. While these approaches clearly increase the index size as well as the retrieval latency (as the query representation needs to be compared against embeddings of all document segments or sentences), sufficiently fast ad-hoc retrieval for most use cases can still be achieved with highly efficient approximate search libraries such as FAISS (Johnson et al., 2017). In the context of monolingual reranking with cross-encoders (Chapter 6), related work shows that aggregating relevance scores (Dai and Callan, 2019; Yilmaz et al., 2019) or representations (Li et al., 2023a) obtained from different passages and sentences improves retrieval results. Representing documents as multiple segments or sentences allows for fine-grained local matching against the query: a setting in which sentence-specialized multilingual encoders are supposed to excel, see Table 5.2.

**Localized Relevance Matching: Segments.** In this approach, we slide a window of size 128 word tokens over the document with a stride of 42 tokens, creating multiple overlapping 128-word segments from the input document. Each segment is then encoded separately, leveraging the encoders from Section 5.2. We then score for relevance each segment by comparing its respective embedding with the query embedding. We then compute the final relevance score by averaging the

| | | EN→X | | | | DE→X | | | FI→X | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | k | FI | IT | RU | DE | FI | IT | RU | IT | RU | AVG | Δ AVG |
| PROC-B | 1 | .242 | .253 | .182 | .286 | .280 | .217 | .158 | .147 | .166 | .215 | −0.86 |
| | 2 | .241 | .244 | .153 | .287 | .282 | .207 | .116 | .147 | .115 | .199 | −2.40 |
| | 3 | .234 | .235 | .150 | .277 | .269 | .194 | .113 | .153 | .109 | .193 | −3.04 |
| | 4 | .228 | .217 | .135 | .255 | .276 | .171 | .105 | .167 | .098 | .184 | −3.95 |
| $\text{DISTIL}_{\text{DmBERT}}$ | 1 | .330 | .327 | .248 | .365 | .324 | .293 | **.244** | .268 | .236 | .293 | +1.32 |
| | 2 | **.349** | .315 | **.269** | .382 | **.347** | .287 | .216 | **.272** | **.226** | **.296** | +1.61 |
| | 3 | .323 | .291 | .261 | .353 | .335 | .268 | .226 | .248 | .208 | .279 | −0.03 |
| | 4 | .299 | .263 | .207 | .330 | .316 | .236 | .189 | .217 | .181 | .249 | −3.10 |
| $\text{DISTIL}_{\text{XLM-R}}$ | 1 | .284 | .218 | .160 | .233 | .267 | .195 | .162 | .181 | .156 | .206 | +2.92 |
| | 2 | .279 | .208 | .164 | .253 | .264 | .194 | .179 | .187 | .157 | .209 | +3.25 |
| | 3 | .264 | .191 | .141 | .228 | .253 | .188 | .145 | .171 | .157 | .193 | +1.60 |
| | 4 | .236 | .169 | .105 | .203 | .237 | .167 | .114 | .153 | .113 | .166 | −1.07 |
| $\text{DISTIL}_{\text{USE}}$ | 1 | .149 | .355 | .202 | .363 | .138 | .332 | .199 | .074 | .118 | .214 | +1.64 |
| | 2 | .162 | **.377** | .192 | .416 | .136 | **.344** | .197 | .081 | .095 | .222 | +2.42 |
| | 3 | .150 | .344 | .180 | .391 | .137 | .319 | .181 | .079 | .091 | .208 | +1.00 |
| | 4 | .135 | .313 | .163 | .364 | .128 | .280 | .158 | .064 | .086 | .188 | −1.03 |
| LaBSE | 1 | .221 | .108 | .124 | .141 | .198 | .093 | .077 | .063 | .143 | .130 | −1.32 |
| | 2 | .212 | .118 | .102 | .189 | .199 | .103 | .060 | .085 | .083 | .128 | −1.13 |
| | 3 | .198 | .104 | .080 | .153 | .190 | .089 | .052 | .076 | .066 | .112 | −2.90 |
| | 4 | .186 | .088 | .065 | .128 | .176 | .075 | .036 | .069 | .049 | .097 | −4.42 |
| mUSE | 1 | .073 | .345 | .215 | .361 | .082 | .331 | .210 | .053 | .084 | .195 | +1.19 |
| | 2 | .102 | .370 | .213 | **.404** | .085 | **.344** | .209 | .056 | .085 | .208 | +2.46 |
| | 3 | .083 | .333 | .198 | .376 | .074 | .296 | .186 | .053 | .082 | .187 | +0.38 |
| | 4 | .075 | .291 | .178 | .348 | .067 | .257 | .178 | .047 | .077 | .169 | −1.43 |
| LASER | 1 | .135 | .058 | .049 | .075 | .155 | .054 | .070 | .082 | .061 | .082 | +1.40 |
| | 2 | .150 | .069 | .071 | .099 | .161 | .055 | .060 | .088 | .062 | .091 | +2.26 |
| | 3 | .136 | .054 | .053 | .074 | .142 | .044 | .052 | .072 | .049 | .075 | +0.71 |
| | 4 | .113 | .037 | .038 | .057 | .118 | .032 | .045 | .052 | .038 | .059 | −0.91 |

**Table 5.3:** Document-level CLIR results for *localized relevance matching* against document *segments* (overlapping 128-token segments). Document relevance is the average of relevance scores of $k$ highest-scoring segments. Results (for 9 language pairs from CLEF) shown for the PROC-B baseline and all similarity-specialized encoders. Δ AVG denotes relative performance increases/decreases w.r.t. the respective base performances from Table 5.1.

relevance scores of the top-$k$ highest-scoring segments.

Table 5.3 displays the results of all multilingual encoders in our comparison, for $k \in \{1, 2, 3, 4\}$.[8] For most encoders (with the exception of LaBSE and the PROC-B baseline) we observe gains from segment-based localized relevance matching: we observe the largest average gain of 3.25 MAP points for $\text{DISTIL}_{\text{XLM-R}}$ (from 0.177 for document encoding to 0.209 for segment-based localized relevance matching).

[8]For $k = 1$, the relevance of the document is exactly the score of the highest scoring segment.

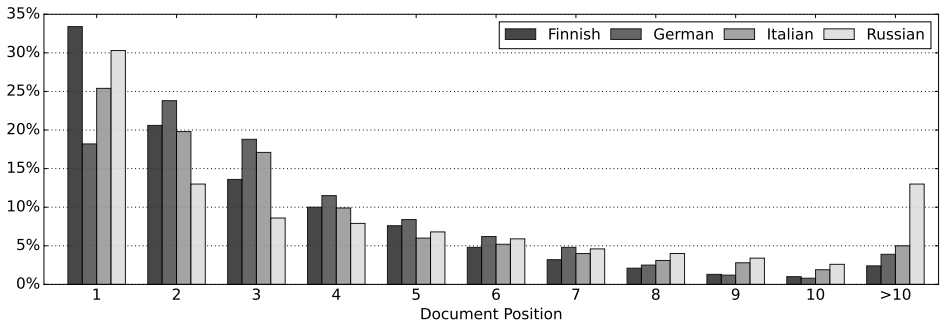


**Figure 5.2:** Comparison of within-document positions of top-ranked segments in segment-based localized relevance matching for different *collection languages*. Proportions aggregated across all multilingual CLIR models from Table 5.3.

Most importantly, we observe gains for our best-performing multilingual encoder $\text{DISTIL}_{\text{DmBERT}}$: localized relevance matching (for $k = 2$) pushes its performance by 1.6 MAP points (the base performance of 0.28 is shown in Table 5.1). We suspect that applying BoW-Agg-IDF in PROC-B (see Equation 4.3 in Section 4.2.1) has a similar (albeit query-independent) soft filtering effect to localized relevance matching and that this is why localized relevance matching does not yield any gains for this competitive baseline.

For all five multilingual encoders for which we observe gains from localized relevance matching, these gains are the largest for $k = 2$, i.e., when we average the relevance scores of the two highest-scoring segments. In 63.7% of the cases, the two highest-scoring segments are mutually consecutive, overlapping segments: we speculate that in those cases it is the span of text in which they overlap that contains the signal that makes the document relevant for the query. Matching queries with the most similar segment embedding effectively filters out the rest of the document. Our results suggest that improvements are pretty consistent across language pairs: we only fail to observe gains when Russian is the language of the target document collection. Localized relevance matching can in principle decrease the performance if segmentation produces (many) false positives (i.e., irrelevant segments with high semantic similarity with the query). We suspect this to more often be the case for Russian than for other languages. We further investigate this by comparing positions of high-scoring segments across document collection languages. We look at the distributions of document positions among the top-ranked 100 segments (gathered from all collection documents): the distributions of top-ranked segment results per positions in respective documents (i.e., 1 indicates the first segment of the document, 2 the second, etc.) are shown for each of the four collection languages (aggregated across all multilingual encoders from Table 5.3) in Figure 5.2. The distributions of positions of high-scoring segments confirms our suspicion that something is different for Russian compared to other languages: we observe a much larger presence of high-scoring segments that appear later in

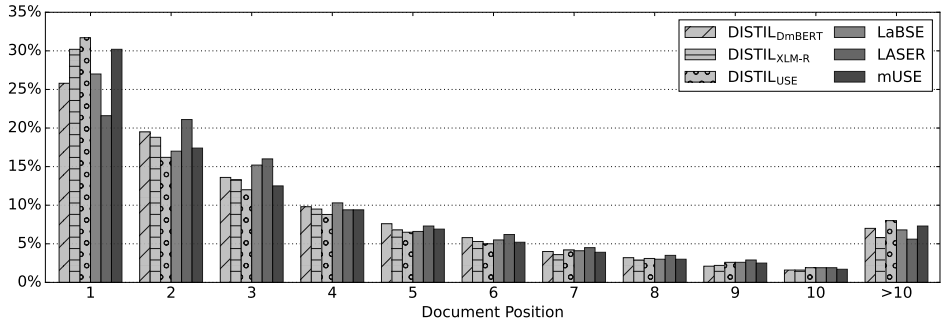


**Figure 5.3:** Comparison of within-document positions of top-ranked segments in segment-based localized relevance matching for different *multilingual text encoders*. Proportions aggregated across all multilingual CLIR models from Table 5.3.

the documents, i.e., at positions larger than 10 (>10): while there is between 2% and 5% of such "late" high-scoring segments in Italian, German, and Finnish collections, in the Russian collection there is 13% of such segments. Our manual inspection confirmed that these late segments are indeed most often false positives (i.e., irrelevant for the query, yet with representations highly similar to those of the queries): this presumably causes the lower performance on *-RU benchmarks.

Figure 5.3 compares the individual multilingual encoders along the same dimension: document positions of the segments they rank the highest. Unlike for collection languages, we do not observe major differences across multilingual encoders – for all of them, the top-ranked segments seem to have similar within-document position distributions, with "early" segments (positions 1 and 2) having the highest relative participation at the top of the ranking. In general, the analysis of positions of high-scoring segments empirically validates the intuition that the most relevant content is often localized at the beginning of the target documents within the newswire CLEF corpora, which in turn reflects the writing style of the news domain.

**Localized Relevance Matching: Sentences.** The selection of the segmentation strategy can have a profound effect on the effectiveness of localized relevance matching. Instead of (overlapping 128-token) segments, one could, for example, measure the relevance of each document sentence for the query and (max-)pool the sentence relevance scores. Sentence-level segmentation and relevance pooling is particularly interesting when considering multilingual encoders that have been specialized precisely for sentence-level semantics (i.e., produce accurate sentence-level representations; see Section 5.3.1). In Table 5.4 we show the results of sentence-level localized relevance matching for all multilingual encoders. Unlike with segment-based localized relevance matching (see Table 5.3), here we see improvements for all multilingual encoders: what is more important, improvements

| | | EN→X | | | | DE→X | | | FI→X | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | k | FI | IT | RU | DE | FI | IT | RU | IT | RU | AVG | Δ AVG |
| Proc-B | 1 | .219 | .207 | .136 | .191 | .235 | .203 | .138 | .089 | .126 | .171 | −5.16 |
| | 2 | .216 | .273 | .158 | .238 | .267 | .247 | .176 | .142 | .122 | .204 | −1.90 |
| | 3 | .229 | .267 | .165 | .245 | .284 | .231 | .168 | .153 | .120 | .207 | −1.61 |
| | 4 | .231 | .247 | .173 | .235 | .286 | .215 | .166 | .150 | .120 | .202 | −2.07 |
| $\text{DISTIL}_{\text{DmBERT}}$ | 1 | **.381** | .288 | .249 | .332 | .338 | .248 | .234 | .234 | .234 | .282 | +0.24 |
| | 2 | .371 | .313 | **.303** | .399 | .343 | .285 | .286 | .246 | **.280** | **.314** | +3.44 |
| | 3 | .360 | .308 | .288 | .407 | **.359** | .274 | **.288** | .247 | .279 | .312 | +3.26 |
| | 4 | .345 | .298 | .264 | .382 | .352 | .262 | .263 | **.248** | .271 | .298 | +1.87 |
| $\text{DISTIL}_{\text{XLM-R}}$ | 1 | .323 | .220 | .144 | .239 | .316 | .215 | .148 | .200 | .149 | .217 | +4.00 |
| | 2 | .339 | .250 | .199 | .306 | .305 | .246 | .200 | .229 | .196 | .252 | +7.51 |
| | 3 | .328 | .260 | .205 | .311 | .318 | .237 | .209 | .222 | .208 | .255 | +7.81 |
| | 4 | .311 | .263 | .188 | .298 | .319 | .225 | .178 | .220 | .179 | .242 | +6.52 |
| $\text{DISTIL}_{\text{USE}}$ | 1 | .131 | .270 | .181 | .332 | .121 | .244 | .200 | .070 | .054 | .178 | −2.01 |
| | 2 | .139 | .331 | .226 | .408 | .134 | .321 | .240 | .076 | .132 | .223 | +2.50 |
| | 3 | .131 | .329 | .220 | **.433** | .129 | .334 | .235 | .074 | .129 | .224 | +2.56 |
| | 4 | .134 | .340 | .212 | .428 | .122 | .329 | .225 | .068 | .124 | .220 | +2.21 |
| LaBSE | 1 | .188 | .182 | .126 | .167 | .185 | .147 | .101 | .112 | .112 | .147 | +0.57 |
| | 2 | .225 | .197 | .182 | .213 | .227 | .180 | .108 | .138 | .139 | .179 | +3.77 |
| | 3 | .245 | .186 | .157 | .234 | .255 | .163 | .089 | .136 | .110 | .175 | +3.39 |
| | 4 | .249 | .192 | .117 | .235 | .248 | .139 | .077 | .145 | .106 | .167 | +2.65 |
| mUSE | 1 | .123 | .270 | .147 | .317 | .112 | .256 | .124 | .070 | .034 | .161 | −2.17 |
| | 2 | .139 | .368 | .212 | .395 | .127 | .334 | .187 | .079 | .069 | .212 | +2.92 |
| | 3 | .142 | **.369** | .230 | .428 | .122 | **.341** | .189 | .083 | .077 | .220 | +3.72 |
| | 4 | .138 | .357 | .220 | .429 | .116 | .331 | .172 | .081 | .086 | .214 | +3.13 |
| LASER | 1 | .207 | .130 | .096 | .147 | .206 | .123 | .107 | .141 | .112 | .141 | +7.30 |
| | 2 | .175 | .172 | .127 | .184 | .206 | .138 | .133 | .165 | .129 | .159 | +9.07 |
| | 3 | .191 | .177 | .153 | .185 | .197 | .141 | .154 | .172 | .136 | .167 | +9.94 |
| | 4 | .175 | .172 | .133 | .179 | .184 | .131 | .125 | .166 | .123 | .154 | +8.60 |

**Table 5.4:** Document-level CLIR results for *localized relevance matching* against document *sentences*. Document relevance is the average of relevance scores of $k$ highest-scoring sentences. Results (for 9 language pairs from CLEF) shown for the Proc-B baseline and all multilingual encoders specialized for encoding sentence-level semantics. Δ AVG denotes relative performance increases/decreases w.r.t. the respective base performances from Table 5.1.

over the baseline performance of the same encoders (see Table 5.1) are substantially larger than for segment-based localized relevance matching (e.g., 10 and 3.8 MAP-point improvements from sentence matching for LASER and LaBSE, respectively, compared to 2-point improvement for LASER and an 1-point MAP drop for LaBSE from segment matching). Sentence-level matching with the best-performing base multilingual encoder $\text{DISTIL}_{\text{DmBERT}}$ and pooling over two highest-ranking sentences (i.e., $k = 2$) yields the best unsupervised CLIR score that we observed overall (31.4 MAP points). For all encoders, averaging the scores of $k = 2$

| | | Segmentation | | Sentence Splitting | |
|---|---|---|---|---|---|
| | #Documents | #Segments | Factor | #Sentences | Factor |
| DE | 294,809 | 1,281,993 | 4.35 | 5,385,103 | 18.27 |
| IT | 157,558 | 749,855 | 4.76 | 2,225,069 | 14.12 |
| FI | 55,344 | 224,390 | 4.05 | 1,286,702 | 23.25 |
| RU | 16,715 | 72,102 | 4.31 | 289,740 | 17.33 |

**Table 5.5:** Increase in computational complexity (i.e., decrease in retrieval efficiency) due to localized relevance matching via segments and sentences.

or $k = 3$ highest-scoring sentences gives better results than considering only the single best sentence (i.e., $k = 1$) – this would indicate that the content relevant to a given query is still not overly localized within documents (i.e., not confined to a single document sentence).

Finally, it is important to note that the gains in retrieval effectiveness (i.e., MAP gains) obtained with localized relevance matching (segment-level and sentence-level) come at the expense of reduced retrieval efficiency (i.e., increased retrieval time): the query representation now needs to be compared with each of the segment or sentence representations, instead of with only one aggregate representation for the whole document. The slowdown factor is proportional to the average number of segments/sentences per document in the document collection. Table 5.5 summarizes the approximate slowdown factors (i.e., average numbers of segments and sentences) for CLEF document collections in different languages.

### 5.5.4 Further Analysis

We now further investigate three aspects that may impact CLIR performance of models based on multilingual encoders: (1) layer(s) from which we take vector representations, (2) number of contexts used in AOC variants, and (3) sequence length in document-level CLIR.

**Layer Selection.** All multilingual encoders have multiple layers and one may in principle choose to take (sub)word representations for CLIR at the output of any of them. Figure 5.4 shows the impact of taking subword representations after each layer for self-supervised mBERT and XLM variants. We find that the optimal layer differs across the encoding strategies (AOC, ISO, and SEMB; cf. Section 5.2) and tasks (document-level vs. sentence-level CLIR). ISO, where we feed the terms into encoders without any context, seems to do best if we take the representations from lowest layers. This makes intuitive sense, as the parameters of higher Transformer layers encode compositional rather than lexical semantics (Ethayarajh, 2019; Rogers et al., 2020b). For AOC and SEMB, where both models obtain representations by contextualizing (sub)words in a sentence, we get the

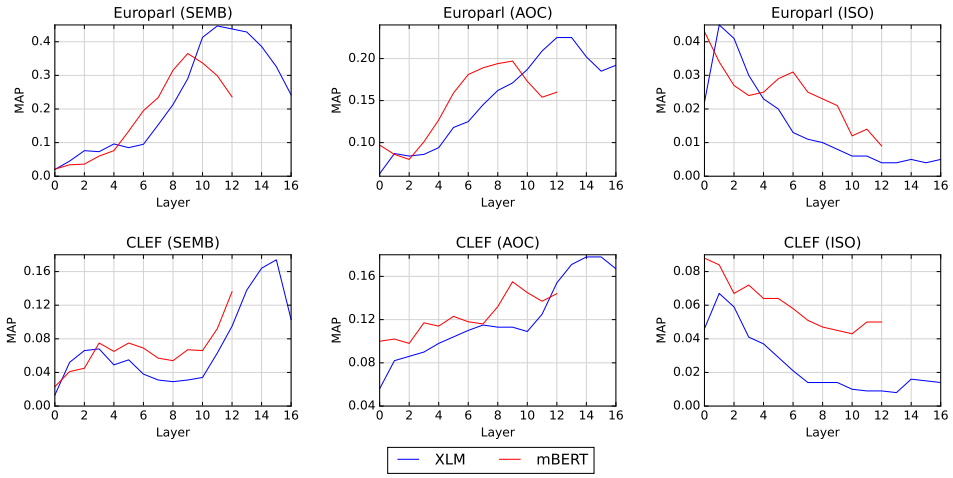


**Figure 5.4:** CLIR performance of mBERT and XLM as a function of the Transformer layer from which we obtain the representations. Results (averaged over all language pairs) shown for all three encoding strategies (SEMB, AOC, ISO).

best performance for higher layers – the optimal layers for document-level retrieval (L9/L12 for mBERT, and L15 for XLM) seem to be higher than for sentence-level retrieval (L9 for mBERT and L11/L12 for XLM). These results are consistent with related work done on monolingual news document retrieval (Fan et al., 2021b; MacAvaney et al., 2019) and retrieval of short documents (Fan et al., 2021b).

**Number of Contexts in AOC.** We construct AOC term embeddings by averaging contextualized representations of the same term obtained from different Wikipedia contexts. This raises an obvious question of a sufficient number of contexts needed for a reliable (static) term embedding. Figure 5.5 shows the AOC results depending on the number of contexts used to induce the term vectors (cf. $\tau$ in Section 5.2). The AOC performance seems to plateau rather early – at around 30 and 40 contexts for mBERT and XLM, respectively. Encoding more than 60 contexts (as we do in our main experiments) would therefore bring only negligible performance gains.

**Input Sequence Length.** Multilingual encoders have a limited input length and they, unlike CLIR models operating on static embeddings (Proc-B, as well as our AOC and ISO variants), effectively truncate long documents. This limitation was, in part, also the motivation for localized relevance matching approaches in the previous section. In our main experiments we truncated the documents to first 128 word pieces. Now we quantify (Table 5.6) if and to which extent this has a detrimental effect on document-level CLIR performance. Somewhat counterintuitively, encoding a longer chunk of documents (256 word pieces) yields a minor performance deterioration (compared to the length of 128) for *all* multilingual encoders. We suspect that this is a combination of two effects: (1) it is more difficult to se-

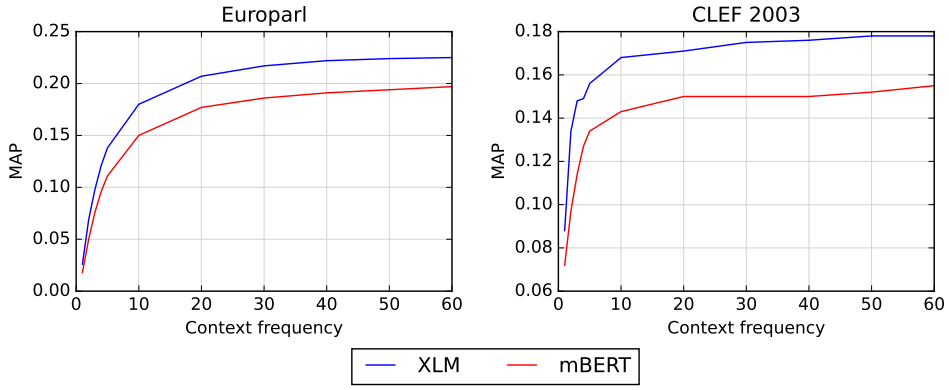


**Figure 5.5:** CLIR performance of AOC variants (mBERT and XLM) w.r.t. the number of contexts used to obtain the term embeddings.

| Length | $\text{SEMB}_{\text{mBERT}}$ | $\text{SEMB}_{\text{XLM}}$ | $\text{DIST}_{\text{use}}$ | $\text{DIST}_{\text{XLM-R}}$ | $\text{DIST}_{\text{DmBERT}}$ | mUSE | LaBSE | LASER |
|---|---|---|---|---|---|---|---|---|
| 64 | .104 | .128 | .235 | .167 | .237 | .254 | .127 | .089 |
| 128 | .137 | .178 | .258 | .162 | .280 | .247 | .125 | .068 |
| 256 | .117 | .158 | .230 | .146 | .250 | .197 | .096 | .027 |

**Table 5.6:** Document-level unsupervised CLIR results w.r.t. the input text length. Scores averaged over all language pairs not involving Finnish.

mantically accurately encode a longer portion of text, which leads to semantically less precise embeddings of 256-token sequences; and (2) for documents in which the query-relevant content is not within the first 128 tokens, that content might often also appear beyond the first 256 tokens, rendering the increase in input length inconsequential to the recognition of such documents as relevant. These results, combined with gains obtained from localized relevance matching in the previous section render localized matching (i.e., document relevance pooled from segment- or sentence-level relevance scores) as a more promising strategy for retrieving long documents than attempts to increase the input length of multilingual Transformers. Our findings from localized relevance matching seem to indicate that the relevance signal is highly localized: in such a setting, aggregating representations of very many tokens (i.e., across the whole document), e.g., with long-input Transformers (Beltagy et al., 2020; Zaheer et al., 2020), is prone to produce semantically fuzzier (i.e., less precise) representations, from which it is harder to judge the document relevance for the query.

### 5.5.5 Few-shot CLIR Results

We now consider a common scenario in which a limited annotation budget exists. That is, we study the performance under a limited amount of "in-domain" relevance judgments that can be leveraged for fine-tuning of text encoders (as op-

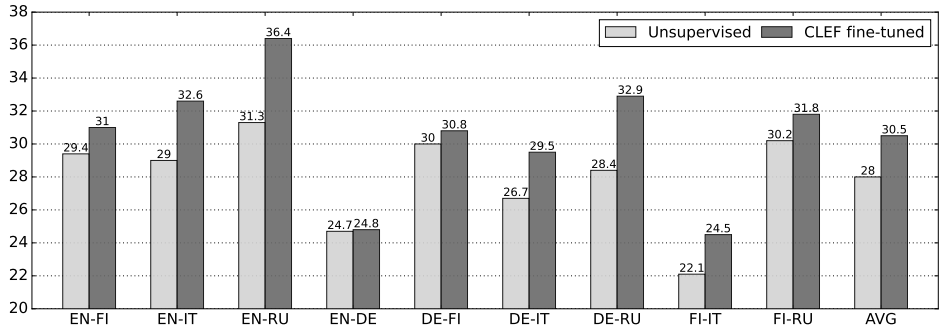


**Figure 5.6:** The effects of "in-domain" fine-tuning: comparison of CLIR performance with $\text{DISTIL}_{\text{DmBERT}}$ on the CLEF CLIR collections: (a) without any fine-tuning (i.e., an unsupervised CLIR approach; see Section 5.5.1) and (b) after in-domain fine-tuning on English CLEF data via contrastive metric-based learning (see Section 5.3.2): here we have only zero-shot language transfer, but no domain transfer (as was the case with L2R models from the previous section).

posed to a large amount of "out-of-domain" training data sufficient to train supervised ranking models, which is the focus of Chapter 6). To this end, we use the relevance judgments in the English portion of the CLEF collection to fine-tune our best-performing multilingual encoder ($\text{DISTIL}_{\text{DmBERT}}$), using the contrastive metric-based learning objective (Section 5.5.1) to refine the representation space of the encoder. We carry out fine-tuning and evaluation in a 10-fold cross-validation setup (i.e., we carry out fine-tuning 10 different times, each time training on different nine-tenths of the queries and evaluating on the remaining one-tenth) in order to prevent any information leakage between languages: in the CLEF collection, queries in languages other than English are simply translations of the English queries. This resulted (in each fold) with a fine-tuning training set consisting of merely 800-900 positive instances (in English). We trained in batches of 16 positive instances and for each of them created all possible in-batch negatives[9] for the Multiple Negative Ranking Loss objective (see Section 5.3.2). With cross-validation in place, for each language pair, we obtain predictions for all queries without any information leakage, which makes the results of contrastive fine-tuning fully comparable with all previous results.

The CLIR results with the contrastively fine-tuned $\text{DISTIL}_{\text{DmBERT}}$ results are shown in Figure 5.6. On average, few-shot fine-tuning improves CLIR results by 2.5 MAP. The largest improvement is obtained for EN-RU (+5.1 MAP points) and the smallest improvement is obtained for EN-DE (+0.1 MAP points). Unlike re-ranking with full-blown pointwise learning-to-rank models (L2R; see Chapter 6), contrastive in-domain reshaping of the representation space of the multi-

[9]This means at most 15 in-batch negatives created from the other query-document pairs in the batch; there is less than 15 negatives only if there are other positive instances for the same query in the batch.

lingual encoder yields performance gains for all language pairs (2.5 MAP points on average). It is important to emphasize that – because contrastive metric-based fine-tuning only updates the parameters of the original multilingual Transformer ($\text{DISTIL}_{\text{DmBERT}}$) and introduces no additional parameters (i.e., no classification head on top of the encoder, as in the case of L2R models) – we can, in exactly the same manner as with the base model before fine-tuning, fully rank the entire document collection for a given query, instead of restricting ourselves to re-ranking the top results of a first-stage ranker.

## 5.6 Conclusion

Pre-trained multilingual encoders have been shown to be widely useful in natural language understanding (NLU) tasks; their utility as general-purpose text encoders in unsupervised settings, such as the ad-hoc cross-lingual IR, has been less investigated. In this chapter, we systematically validated the suitability of a wide spectrum of multilingual encoders for document- and sentence-level CLIR across diverse languages. We first profiled the popular self-supervised multilingual encoders (mBERT and XLM) as well as the multilingual encoders specialized for semantic text matching on semantic similarity datasets and parallel data as text encoders for unsupervised CLIR. Our empirical results show that self-supervised multilingual encoders (mBERT and XLM), without exposure to task supervision, generally fail to outperform CLIR models based on static cross-lingual word embeddings (CLWEs). Semantically specialized multilingual sentence encoders, on the other hand, do outperform CLWEs; the gains, however, are pronounced only in sentence retrieval, while being much more modest in document retrieval. Acknowledging that sentence-specialized multilingual encoders are not designed for encoding long documents, we proposed to exploit their strength – precise semantic encoding of short texts – in document retrieval too, by means of localized relevance matching, where we compare the query with individual document segments or sentences and max-pool the relevance scores; we showed that such localized relevance matching with sentence-specialized multilingual encoders yields substantial document-level CLIR gains. While multilingual text encoders excel in so many seemingly more complex language understanding tasks, our work renders ad-hoc CLIR in general and document-level CLIR in particular a serious challenge for these models. Furthermore, we investigated alternative supervised approach, based on contrastive metric-based learning and few-shot learning, designed for fine-tuning the representation space of a multilingual encoder when only a limited amount of "in-domain" relevance judgments is available. We show that such small-scale in-domain fine-tuning of multilingual encoders yields consistent improvements over their unsupervised counterparts. We make our code and resources available at: https://github.com/rlitschk/EncoderCLIR

# Part III

# Resource-Lean Transfer of Cross-Encoders

# Chapter 6

# Zero-shot Language and Domain Transfer of Rerankers

[1]In *Part II Resource-Lean Transfer of Bi-Encoders*, we focused on static and contextual cross-lingual word representations and their effectiveness on cross-lingual information retrieval (CLIR), following the bi-encoder paradigm. We investigated fully unsupervised CLIR methods and resource-lean setups, which rely on minimal cross-lingual supervision in the form of bilingual dictionaries (Chapters 4 and 5). We also investigated few-shot CLIR where we assumed access to some *in-domain* training data, i.e. CLIR task supervision (Section 5.5.5).

In *Part III Resource-Lean Transfer of Cross-Encoders* (Chapters 6 to 8), we now focus on the common setup where we have access to large-scale monolingual training data (i.e. IR task supervision) in English and investigate zero-shot cross-lingual transfer (ZS-XLT) of cross-encoder reranking models. In this chapter, following the pointwise learning-to-rank approach (Liu, 2009), we leverage existing rankers based on multilingual text encoders and evaluate their performance when applied on CLIR. We specifically compare two ranking models trained on different English *out-of-domain* training data. That is, we investigate their transfer performance where the languages and domains are different between the training and test dataset. We further quantify the performance gap between zero-shot transfer into a cross-lingual setup (ZS-XLT into CLIR) and zero-shot transfer into a monolingual information retrieval setup (ZS-XLT into MoIR). Our results show that models trained on English MoIR data transfer substantially better into MoIR than into CLIR. We refer to the underlying phenomenon as "monolingual overfitting" and propose a way to regularize it in the next chapter (Chapter 7).

[1]This chapter is adapted from: (1) **Robert Litschko**, Ivan Vulic, Simone Paolo Ponzetto, and Goran Glavaš. 2022. On cross-lingual retrieval with multilingual text encoders. *Information Retrieval Journal 25.2*, pages 149–183; (2) **Robert Litschko**, Ivan Vulic, and Goran Glavaš. 2022. Parameter-efficient neural reranking for cross-lingual and multilingual retrieval. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pages 1071–1082, Gyeongju, Republic of Korea.

## 6.1 Introduction

In this chapter, the focus is on multilingual cross-encoder rerankers applied in a multi-stage retrieval framework (Nogueira et al., 2019b). Multi-stage retrieval breaks the retrieval task down into first-stage retrieval, in which a fast preranker is applied to obtain an initial ranking, and second-stage reranking, where the top-$k$ documents are reranked with a more powerful but slower reranker. We further focus on the ZS-XLT paradigm (Liang et al., 2020; Hu et al., 2020; Ponti et al., 2020), which is commonly adopted in natural language processing (NLP) tasks to transfer models from high-resource to low-resource languages. However, characteristic to CLIR and different from most NLP tasks is that in CLIR models are transferred into a setup where the input is written in two different languages. More precisely, consider a ZS-XLT scenario where a model is transferred into the query-document pair $L_1$-$L_2$. We distinguish between ZS-XLT into CLIR where $L_1 \neq L_2$ and ZS-XLT into a monolingual information retrieval task (MoIR) where $L_1$ and $L_2$ are the same language (and different from the training language). We emphasize the distinction between CLIR and MoIR as two different tasks. The standard approach of training zero-shot models on English data results into MoIR ranking models (MacAvaney et al., 2019; Shi and Lin, 2019; Shi et al., 2020; Jiang et al., 2020b). At test time, these models can then either be applied on a MoIR task, which we refer to as *same task transfer* (MoIR→MoIR), or on a CLIR task, i.e. *cross-task transfer* (MoIR→CLIR). We find large performance differences between the two transfer setups (Section 6.4.2) and argue that MoIR is a suboptimal source task, because rerankers "overfit" to features that do not transfer well to CLIR.

ZS-XLT is enabled by multilingual pre-trained language models (mPLM) which encode different languages in a shared multilingual representation space (see Section 2.3.3). In the context of retrieval, MacAvaney et al. (2019) focus on ZS-XLT for MoIR and train a cross-encoder based on mBERT (Devlin et al., 2019) on the English news retrieval dataset TREC Robust04 (Voorhees, 2004). The authors evaluate their model on in-domain data in three target languages and show that it generally outperforms BM25 with substantial gains. Similar findings were made by Shi et al. (2020) who trained an mBERT-based reranker on out-of-domain data. We instead consider the more challenging setup and additionally evaluate the cross-task transfer performance of rerankers (MoIR→CLIR). We further investigate to what extent *domain differences* between the training and test set distributions affect the transfer performance. In this regard, our work is most similar to (Shi and Lin, 2019), who also perform ZS-XLT into MoIR and CLIR, and also consider domain differences between the training data and test data. Their ranking model, a cross-encoder based on mBERT, is trained on English tweets and applied on news retrieval datasets in different languages. Their results show that, despite the train-test domain differences, their model consistently outperforms BM25 in MoIR and BM25 combined with query translation in CLIR. However, all target language pairs used in their CLIR evaluation involve English on the query-side or document-side. This might lead to overly optimistic estimates of the transfer performance because

(1) English represents the largest share in mBERT's pre-training corpus (Conneau et al., 2020) and (2) each target language pair partially overlaps with the training language. In our experiments we evaluate models on nine cross-lingual language pairs where five pairs do not include English. Lastly, Jiang et al. (2020b) use parallel data to derive weak supervision for CLIR. This is different from our work, since we only assume availability of monolingual retrieval supervision.

As mentioned above, in addition to comparing task differences, i.e. zero-shot transfer into CLIR vs. MoIR, we also study *domain mismatches* between training and test data. Domain effects have been extensively studied in NLP tasks (Plank and van Noord, 2011; Müller et al., 2020; Glavaš et al., 2020, *inter alia*) and retrieval tasks (Akkalyoncu Yilmaz et al., 2019; Albalak et al., 2023), where they have been shown to impair the performance in transfer learning setups (Ruder et al., 2019). In Section 5.5.5, we showed that fine-tuning models on in-domain data (few-shot CLIR) consistently leads to improved retrieval results. In this chapter, we compare the performance of two off-the-shelf reranking models trained on the out-of-domain MS MARCO dataset (Nguyen et al., 2016) and in-domain TREC Robust04 dataset (Voorhees, 2004). We evaluate both models on the CLEF 2003 news retrieval dataset (Braschler, 2004) (see Section 3.3).

**Contributions.** Our key contributions are summarized as follows:

**(1)** We show that, on average, supervised neural rerankers (based on multilingual Transformers such as mBERT) trained on English relevance judgments from *different collections* (i.e., zero-shot language and domain transfer) do not surpass the best performing unsupervised CLIR approach based on multilingual sentence encoders (Section 6.4.1).

**(2)** We show that fine-tuning supervised CLIR models based on multilingual Transformers on monolingual (English) data leads to a type of "overfitting" to monolingual retrieval (Section 6.4.2): such models transfer much better to monolingual retrieval tasks in unseen target languages (*same task transfer*) than to cross-lingual retrieval tasks (*cross-task transfer*).

**Resource-Lean Transfer.** In this chapter, we study zero-shot cross-lingual transfer of cross-encoder rerankers. To train reranking models, we only require access to *monolingual supervision*, which allows for a *resource-lean* transfer from high-resource languages (e.g., English) to target language pairs for which we have no training data available (cf. Chapter 1, Figure 1.4). Arguably, monolingual task supervision is easier to obtain than cross-lingual supervision because it (1) poses fewer requirements on human annotators and (2) can be automated by discretizing lexical relevance scores into labels (Sun and Duh, 2020).

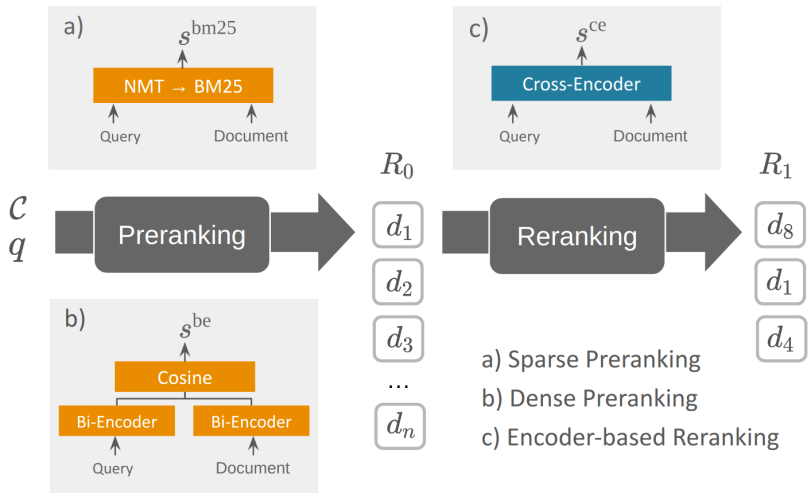


**Figure 6.1:** Overview of the multi-stage ranking approach to ad-hoc retrieval. **Stage 1 - Preranking:** We rank the document collection $\mathcal{C}$ by (a) running *sparse* BM25 retrieval on translated queries, or (b) according to the cosine similarity between *dense* query and document representations yielding an initial ranking $R_0$. **Stage 2 - Reranking:** We refine $R_0$ by reranking the top-k documents according to relevance scores predicted by a Cross-Encoder, yielding the refined ranking $R_1$.

## 6.2 Methodology

In the Chapters 6 to 8, we rely on the multi-stage ranking paradigm (Nogueira et al., 2019b) to evaluate cross-encoders (Nogueira and Cho, 2019) for zero-shot cross-lingual transfer for CLIR. We briefly review the basic components of multi-stage ranking systems and discuss how we adopt this paradigm for CLIR. As discussed in Section 3.4, pre-trained Transformers like BERT (Devlin et al., 2019) are often used as *Cross-Encoder (CE)* scoring models: the Transformer encodes a query-document concatenation fed as input to the model, and the encoding is then fed to a dense layer that predicts the relevance score (MacAvaney et al., 2020b; Jiang et al., 2020a; Nogueira et al., 2019b). Due to the quadratic complexity of the multi-head attention mechanism, computing scores for all query-documents pairs with Cross-Encoders is too slow for practical IR applications: they are thus primarily used as rerankers in a multi-stage ranking approach (MacAvaney et al., 2020b; Geigle et al., 2022). Figure 6.1 illustrates our multi-stage CLIR workflow.[2]

**First-Stage Retrieval.** Preranking, based on a fast and efficient ranking method, is applied to every document from the document collection in order to provide a good initial ranking, targeting high recall. Let $q^{l1}$ be a query in language $l_1$ and

[2]In this chapter, we use multilingual bi-encoders (CLIR) and BM25 (MoIR) as first-stage rankers. In Chapter 8 we additionally investigate BM25 together with neural machine translation (NMT).

$\mathcal{C}^{l2} = \{d_i\}_{i=1}^{n}$ be a document collection containing $n$ documents in language $l_2$. Associating and ranking documents w.r.t. relevance scores $s_i$ we obtain an initial ranking

$$R_0 = [(d_1, s_1), (d_2, s_2) \dots (d_n, s_n)], \tag{6.1}$$

where $s_1 > s_2 > \dots s_n$. We transfer our rerankers based on multilingual pre-trained languages models – and trained on English relevance judgments – to (i) CLIR tasks as well as to (ii) monolingual IR tasks in target languages, termed MoIR. In MoIR, we opt for a lexical preranker and score documents with $s^{\text{bm25}} = \text{BM25}(q, d)$. A widely used approach in CLIR is to machine translate the query (Bonifacio et al., 2021; Lawrie et al., 2022): this process effectively translates CLIR into a noisy variant of MoIR. We study first stage retrievers based on query translation in the next chapter. In this chapter, we experiment with a representation-based approach based on pre-trained multilingual *Bi-Encoders* (*BE*): here, we embed the query and documents independently, and then use the cosine similarity between their embeddings $s^{be} = cos(BE(q), BE(d))$. In the preranking stage, unlike later in reranking, we use the encoders merely as general-purpose text encoders, without any additional retrieval-specific training. To compare our results to previous bi-encoder experiments (Chapter 5) we use the rankings produced by the first-stage retrievers (i.e., multilingual sentence encoders) from Section 5.3.1.

**Second-Stage Reranking.** This stage refines the initial ranking obtained via preranking. It relies on a $CE$ model which captures fine-grained (but more costly to model and run) semantic interactions between queries and documents. The ranking is then:

$$R_1 = [(d_1, s_1^{ce}), (d_2, s_2^{ce}) \dots (d_k, s_k^{ce})] \tag{6.2}$$

To this end, we rely on multilingual CEs to compute the binary relevance score $s^{ce}$ on the concatenation of query and document pairs:

$$s^{ce} = CE(\texttt{[CLS]}\, q\, \texttt{[SEP]}\, d_i\, \texttt{[SEP]}) \tag{6.3}$$

Reranking the top $k$ documents with CEs yields the final ranking $R_1$, common choices for $k$ include $k = 100$ (MacAvaney et al., 2019; Craswell et al., 2020b; Naseri et al., 2021) and $k = 1,000$ (Nguyen et al., 2016; Khattab and Zaharia, 2020). In this chapter, we experiment with two existing rerankers. The first model[3] was trained on the large-scale MS MARCO passage retrieval dataset (Nguyen et al., 2016), consisting of approx. 400M tuples, each consisting of a query, a relevant passage and a non-relevant passage. Transferring rankers trained on MS MARCO to various ad-hoc IR settings (i.e., domains) has been shown successful (Li et al., 2023a; MacAvaney et al., 2020a; Craswell et al., 2021b). Here, we investigate the

[3] https://huggingface.co/amberoad/bert-multilingual-passage-reranking-msmarco

performance of this supervised ranker trained on MS MARCO in simultaneous domain and language transfer. The second multilingual pointwise ranker (MacAvaney et al., 2019) is trained on TREC Robust04 dataset (Voorhees, 2004). Although this dataset is substantially smaller than MS MARCO (528K documents and 311K relevance judgments), by covering newswire documents it is domain-wise closer to our target CLEF test collection.

## 6.3 Experimental Setup

**Evaluation Datasets and Prerankers.** We adopt our standard experimental setup for document-level retrieval introduced in Section 3.3 and use the CLEF 2003 benchmark (Braschler, 2004). That is, for MoIR we evaluate our rerankers on English (EN), Finnish (FI), German (DE), Italian (IT) and Russian (RU). Here, we experiment with two prerankers. Our first preranker, query likelihood model (QLM), is a lexical model described in Section 3.3. Our second preranker uses fastText embeddings (Bojanowski et al., 2017a), which are described in Section 2.1, and follows the BoW-Agg-IDF approach explained in Section 4.2.

For CLIR, we reuse the same language pairs as in the previous chapter: EN-{FI, IT, RU, DE}, DE-{FI,IT,RU} and FI-{IT, RU}. For first-stage retrieval, we reuse the rankings produced by cross-lingual bi-encoder models: DISTIL (Reimers and Gurevych, 2020), mUSE (Yang et al., 2020b), LaBSE (Feng et al., 2022), and LASER (Artetxe and Schwenk, 2019a), we refer the reader to Section 5.2 for details. For completeness, we also include our BoW-Agg-IDF model based on cross-lingual word embeddings (Proc-B) from Chapter 4 as a preranker.

**Reranking Model Details.** Transferring (re-)rankers across domains and/or languages is a promising method when in-language and in-domain fine-tuning data is scarce (MacAvaney et al., 2019). As mentioned above, for second-stage reranking, we experiment with two pointwise rankers, both based on mBERT, pre-trained on English relevance data. The first model is made available on HuggingFace (Wolf et al., 2020)[4] and was trained on the large-scale MS MARCO passage retrieval dataset (Nguyen et al., 2016). Transferring rankers trained on MS MARCO to various ad-hoc IR settings (i.e., domains) has been shown successful (Li et al., 2023a; MacAvaney et al., 2020a; Craswell et al., 2021b). Here, we investigate the performance of this supervised ranker trained on MS MARCO in simultaneous domain and language transfer. The second multilingual pointwise ranker (MacAvaney et al., 2020b) is trained on TREC 2004 Robust dataset (Voorhees, 2004). While this dataset is much smaller, it is also much closer to the target domain (i.e. in-domain transfer). Both models are used to rerank the top $k = 100$ pre-ranked documents, yielding the final ranking $R_1$.

[4] `amberoad/bert-multilingual-passage-reranking-msmarco`

## 6.4 Results and Discussion

In this section, we investigate ZS-XLT from two perspectives. We first evaluate simultaneous domain and language transfer of rerankers trained on large-scale English retrieval data (MoIR) (Section 6.4.1). In Section 6.4.2, we then compare the difference between transferring reranking models into monolingual (ZS-XLT into MoIR) versus cross-lingual settings (ZS-XLT into CLIR).

### 6.4.1 Domain and Language Transfer Effects

**Language Transfer Effects.** In Table 6.1 we summarize the results of domain and language transfer experiments with the two pointwise mBERT-based rerankers. For clarity, at the top of the table, we repeat the unsupervised CLIR results from Table 5.1, which we obtained with similarity-specialized multilingual encoders (i.e., without any re-ranking). Intuitively, re-ranking – both with the MS MARCO-trained model and TREC-trained model – brings the largest gains for the weakest unsupervised rankers: mUSE, LaBSE, and LASER (cf. Section 5.5.1). The gains are somewhat larger when transferring the model trained on MS MARCO. Comparing the overall average performance, re-ranking the results of the best-performing unsupervised ranker – $\text{DISTIL}_{\text{DmBERT}}$ – brings no performance gains; in fact, re-ranking with the TREC-trained model reduces the quality of the base ranking by 7 MAP points. A possible explanation for this could be that stronger prerankers also introduce more challenging false positives (Gao et al., 2021b).

Upon closer inspection, we find that the transfer performance of the better-performing MS MARCO re-ranker depends on the performance of the first-stage retriever and the target language pair. For example, under the best-performing first-stage retriever, $\text{DISTIL}_{\text{DmBERT}}$, the re-ranker improves the performance in five out of nine language pairs. Notably, three out of the remaining four language pairs ({EN, DE, FI}→RU and FI→IT) involve Russian. We hypothesize that this is because (1) training rerankers on MoIR data does not expose models to query-document pairs written in different scripts and (2) Russian, similar to Finnish, is typologically distant from the training language (English). For transfer setups involving the same script (i.e., Latin script), rerankers trained on MS MARCO consistently improve first-stage rankings from all prerankers.[5] These results suggest that, assuming a strong multilingual bi-encoder as the first-stage retriever, supervised re-ranking models do not transfer well to distant language pairs compared to languages from the same family. Unsurprisingly, we obtain better reranking results for language pairs involving English queries, suggesting that limiting CLIR evaluation to EN on the query-side or document-side (Shi and Lin, 2019) is not a reliable measure of the true ZS-XLT abilities of multilingual cross-encoders.

[5]The only exception to this is FI-IT with $\text{DISTIL}_{\text{DmBERT}}$ as a preranker.

| | EN→X | | | | DE→X | | | FI→X | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | FI | IT | RU | DE | FI | IT | RU | IT | RU | AVG | Δ AVG |
| *No re-ranking (reference)* | | | | | | | | | | | |
| Proc-B | .258 | .265 | .166 | .288 | .294 | .230 | .155 | .136 | .216 | .223 | – |
| $\text{DIST}_{\text{DmBERT}}$ | .294 | .290 | **.313** | .247 | .300 | .267 | **.284** | **.221** | **.302** | **.280** | – |
| $\text{DIST}_{\text{XLM-R}}$ | .219 | .191 | .149 | .148 | .215 | .179 | .142 | .167 | .125 | .170 | – |
| $\text{DIST}_{\text{USE}}$ | .141 | .346 | .182 | .258 | .139 | .324 | .179 | .104 | .111 | .198 | – |
| mUSE | .077 | .313 | .186 | .262 | .077 | .293 | .183 | .053 | .092 | .171 | – |
| LaBSE | .191 | .163 | .136 | .087 | .172 | .136 | .103 | .117 | .140 | .138 | – |
| LASER | .146 | .092 | .060 | .039 | .153 | .089 | .062 | .117 | .076 | .093 | – |
| *Re-ranker trained on MS MARCO* | | | | | | | | | | | |
| Proc-B | .327 | .330 | .191 | **.321** | .321 | .230 | .212 | .160 | .149 | .246 | +2.30 (8) |
| $\text{DIST}_{\text{DmBERT}}$ | **.340** | .335 | .219 | .288 | .339 | .284 | .245 | .217 | .160 | .270 | −1.02 (5) |
| $\text{DIST}_{\text{XLM-R}}$ | .310 | .252 | .137 | .232 | **.370** | .219 | .165 | .183 | .062 | .270 | +3.74 (7) |
| $\text{DIST}_{\text{USE}}$ | .215 | **.354** | .224 | .295 | .219 | **.310** | .236 | .133 | .075 | .229 | +3.10 (7) |
| mUSE | .170 | .348 | .235 | .314 | .162 | .301 | .253 | .110 | .093 | .220 | +4.95 (9) |
| LaBSE | .300 | .275 | .169 | .170 | .360 | .240 | .138 | .166 | .190 | .240 | +**6.60** (9) |
| LASER | .258 | .166 | .089 | .092 | .228 | .151 | .114 | .127 | .160 | .148 | +5.49 (9) |
| *Re-ranker trained on TREC Robust04* | | | | | | | | | | | |
| Proc-B | .290 | .292 | .141 | .310 | .278 | .214 | .148 | .108 | .103 | .209 | −1.40 (3) |
| $\text{DIST}_{\text{DmBERT}}$ | .284 | .283 | .153 | .274 | .252 | .246 | .130 | .147 | .119 | .210 | −7.00 (1) |
| $\text{DIST}_{\text{XLM-R}}$ | .270 | .227 | .093 | .242 | .226 | .200 | .079 | .129 | .069 | .170 | +0.00 (5) |
| $\text{DIST}_{\text{USE}}$ | .195 | .321 | .119 | .309 | .194 | .287 | .113 | .113 | .117 | .196 | −0.20 (5) |
| mUSE | .143 | .330 | .129 | .313 | .139 | .261 | .131 | .086 | .079 | .179 | +0.80 (4) |
| LaBSE | .275 | .234 | .086 | .158 | .245 | .180 | .076 | .115 | .077 | .161 | +2.30 (5) |
| LASER | .201 | .164 | .121 | .095 | .171 | .137 | .118 | .111 | .093 | .135 | +4.20 (9) |

**Table 6.1:** Document-level CLIR results on the CLEF collection obtained by language and domain transfer of supervised re-ranking models. For each query, we re-rank the top 100 results produced by the base multilingual ranker with two mBERT-based L2R models trained on English data: MS MARCO (Nguyen et al., 2016) (middle part of the table) and TREC Robust04 (bottom third of the table) (Voorhees, 2004; MacAvaney et al., 2020b). **Bold**: the best performance for each language pair and the average. Δ: performance difference compared to preranker and the number of language pairs for which reranking improved the results.

**Domain Transfer Effects.** We now analyze domain differences between the fine-tuning and test datasets. Recall that the first reranker, as shown in the middle part of Table 5.1, is trained on the MS MARCO dataset (Nguyen et al., 2016), which is based on web search data, and the second model (bottom half) is trained on the TREC Robust04 dataset (Voorhees, 2004), which is based on news retrieval data. Despite its lower domain similarity, we find that MS MARCO appears to be the better source task. This is somewhat counterintuitive and might be explained

| | EN | FI | DE | IT | RU | AVG | Δ AVG |
|---|---|---|---|---|---|---|---|
| *No re-ranking (reference)* | | | | | | | |
| QLM | .471 | .376 | .400 | .463 | .325 | .407 | – |
| FastText | .310 | .327 | .314 | .314 | .214 | .296 | – |
| *Re-ranker trained on MS MARCO* | | | | | | | |
| QLM | **.520** | .469 | **.424** | **.488** | **.359** | **.452** | +4.53 |
| FastText | .434 | .430 | .384 | .468 | **.359** | .415 | +**11.90** |
| *Re-ranker trained on TREC Robust04* | | | | | | | |
| QLM | .481 | **.520** | .420 | .454 | .303 | .436 | +1.98 |
| FastText | .375 | .462 | .367 | .429 | .299 | .386 | +8.76 |

**Table 6.2:** Cross-lingual zero-shot transfer for monolingual retrieval: results on the monolingual CLEF portions. Base rankers (top third of the table) – QLM with Dirichlet Smoothing and aggregation of static monolingual word embeddings (fastText) and re-ranking with pointwise mBERT-based models trained on English MS MARCO (middle third) and TREC Robust04 data (bottom third), respectively.

by the fact that it is trained on a much larger dataset. Notably, our results in Section 5.5.5 allow us to compare the difference between few in-domain data in the target language (few-shot CLIR) against large-scale training data in a different source language (zero-shot CLIR). While the former setup yields consistent performance gains on all target language pairs, we find that the latter performs worse. In this regard, our results are in line with Craswell et al. (2021b), who conclude that in-domain data, in addition to MS MARCO, is crucial to successfully transfer ranking models. However, it is important to note that in our comparison both approaches use different encoder models (mBERT vs. DmBERT) and retrieval paradigms (bi-encoder vs. cross-encoder). In future work, we plan to investigate whether few-shot CLIR models still outperform zero-shot CLIR models when both follow the same setting. In summary, our results suggest that, at least in ZS-XLT for CLIR, (1) the size of the training dataset plays a crucial role and (2) domain similarity alone is insufficient for a successful transfer. Next, we control for train-test task differences by transferring both rerankers to a monolingual target (MoIR→MoIR).

### 6.4.2 Same Task vs. Cross-Task Transfer

At first glance, our CLIR results for the mBERT-based pointwise L2R rankers (Section 6.4.1) – i.e., the fact that using them for re-ranking does not improve the performance of our best-performing unsupervised ranker ($\text{DISTIL}_{\text{DmBERT}}$) – seem at odds with their solid cross-lingual transfer results reported in previous work (MacAvaney et al., 2020b). It is, however, important to notice the fundamental difference between the two evaluation settings: what was previously evaluated (MacAvaney et al., 2020b) was the effectiveness of (zero-shot) *cross-lingual transfer* of a *monolingual retrieval* model, trained on English data and transferred to

a set of target languages. In other words, both in training and at inference time, the models deal with queries and documents written in the same language. Our work here, instead, focuses on a fundamentally different scenario of cross-lingual retrieval, where the language of the query is different from the language of document collection. We argue that, in a supervised setting, in which one trains on monolingual English data only, the latter (i.e., CLIR) represents a more difficult transfer setup.

To validate the above assumption, we additionally evaluate the two mBERT-based re-rankers from Section 6.4.1 trained on MS MARCO and TREC Robust04, respectively, on monolingual portions of the CLEF collection. We use them to re-rank two strong monolingual baselines: (1) Query Likelihood Model (QLM, based on unigrams) (Ponte and Croft, 1998) with Dirichlet smoothing (Zhai and Lafferty, 2004), which we also used for the machine-translation baseline (MT-IR) in our previous evaluation (see Section 5.4); and (2) a retrieval model based on aggregation of IDF-scaled static word embeddings (see Section 2.2).[6] For the latter, we used the monolingual fastText embeddings trained on Wikipedias of the respective languages,[7] with vocabularies limited to the 200K most frequent terms.

The results of mBERT-based re-rankers in cross-lingual transfer for monolingual retrieval are summarized in Table 6.2. We see that, unlike in CLIR (see Table 6.1), mBERT-based re-rankers do substantially and consistently improve the performance of the base retrieval models, even though the base performance of the monolingual baselines (QLM and fastText) is significantly above the best CLIR performance obtained with unsupervised rankers (see $\text{DISTIL}_{\text{DmBERT}}$ in Table 6.1; MAP: 0.280). This is in line with the findings from (MacAvaney et al., 2020b): multilingual encoders (e.g., mBERT) do seem to be a viable solution for (zero-shot) cross-lingual transfer of learning-to-rank models for monolingual retrieval (i.e., MoIR→MoIR). But why are they not as effective when transferred to CLIR settings (as shown in Section 6.4.1)? We hypothesize that monolingual English training on large-scale datasets leads to a sort of *"overfitting" to monolingual retrieval* (e.g., the model may implicitly learn to assign a lot of importance to exact term matches) – such (latent) features will, in principle, transfer reasonably well to other monolingual retrieval settings, regardless of the target language. However, CLIR instances are likely to generate out-of-training-distribution values for these latent features (e.g., if the model learned to value exact matches during training, at predict time in CLIR settings, it would need to recognize word-level translations between the two languages), confusing the pointwise classifier.

[6] This corresponds to the Proc-B baseline in CLIR evaluations; only here we use monolingual embeddings of the target language (instead of a bilingual word embedding space, as in CLIR).

[7] https://fasttext.cc/docs/en/pretrained-vectors.html

## 6.5 Conclusion

In this chapter, we investigated the effectiveness of supervised (re-)rankers, based on multilingual encoders, in zero-shot cross-lingual transfer for ad-hoc document-level CLIR evaluation setups. Summarizing the results from this chapter and our few-shot CLIR experiments in Section 5.5.5, it appears that – at least when it comes to zero-shot language transfer for cross-lingual document retrieval – specializing the representation space of a multilingual encoder with few(er) in-domain relevance judgments is more effective than employing a neural L2R ranker trained on large amounts of "out-of-domain" data.

We further show that, while rankers trained monolingually on large-scale English datasets can be successfully transferred to monolingual retrieval tasks in other languages, their transfer to CLIR setups, in which the query language differs from the language of the document collection, is much less successful. Our findings indicate that, during monolingual fine-tuning cross-encoders "overfit" to features that do not transfer well to CLIR tasks. In the next chapter, we systematically dissect this phenomenon, which we refer to as *"monolingual overfitting"*. We show that this type of overfitting can be effectively regularized by training cross-encoders on artificially code-switched data.

# Chapter 7

# Regularizing Monolingual Overfitting

[1]In this chapter, we build on our observation that the effectiveness of zero-shot rerankers, trained on monolingual data, diminishes when they are transferred into a setup where queries and documents are in different languages (cf. Section 6.4.2). Motivated by this, we propose to train ranking models on artificially code-switched data instead, which we generate by utilizing bilingual lexicons. We experiment with lexicons induced from (1) cross-lingual word embeddings and (2) parallel Wikipedia page titles. We use the multilingual MARCO dataset (mMARCO) (Bonifacio et al., 2021) to extensively evaluate reranking models on 36 language pairs spanning Monolingual IR (MoIR), Cross-lingual IR (CLIR), and Multilingual IR (MLIR). We choose mMARCO because, contrary to CLEF 2003 (Braschler, 2004) used in previous chapters, it consists of parallel queries and documents and also contains large training datasets. Our results show that code-switching can yield consistent and substantial gains of 5.1 MRR@10 in CLIR and 3.9 MRR@10 in MLIR, while maintaining stable performance in MoIR. Encouragingly, the gains are especially pronounced for distant languages (up to 2x absolute gain). We further show that our approach is robust towards the ratio of code-switched tokens and also extends to unseen languages. Our results demonstrate that training on code-switched data is a cheap and effective way of generalizing zero-shot rankers for cross-lingual and multilingual retrieval.

## 7.1 Introduction

Cross-lingual Information Retrieval (CLIR) is the task of retrieving relevant documents written in a language different from a query language. The large number of

[1]This Chapter is adapted from: **Robert Litschko**, Ekaterina Artemova, Barbara Plank. 2023. Boosting Zero-shot Cross-lingual Retrieval by Training on Artificially Code-Switched Data. In *Findings of the Association for Computational Linguistics* (Findings of ACL), pages 3096–3108, Toronto, Canada.

languages and limited amounts of training data pose a serious challenge for training ranking models. Previous work address this issue by using machine translation (MT), effectively casting CLIR into a noisy variant of monolingual retrieval (Li and Cheng, 2018; Shi et al., 2020, 2021; Moraes et al., 2021). MT systems are used to either train ranking models on translated training data (*translate train*), or by translating queries into the document language at retrieval time (*translate test*). However, CLIR approaches relying on MT systems are limited by their language coverage. Because training MT models is bounded by the availability of parallel data, it does not scale well to a large number of languages. In Section 8.4 we further show that using MT for IR is prone to propagation of unwanted translation artifacts such as topic shifts, repetition, hallucinations and lexical ambiguity (Artetxe et al., 2020; Li et al., 2022). In this chapter, we propose a *resource-lean MT alternative* to bridge the language gap and propose to use *artificially code-switched* data.

We focus on zero-shot cross-encoder (CE) models for reranking (MacAvaney et al., 2020b; Shi and Lin, 2019; Jiang et al., 2020b). Our study is motivated by the observation that the performance of CEs diminishes when they are transferred into CLIR as opposed to MoIR (see Section 6.4.1). In this chapter, we confirm our findings on the multilingual passage retrieval dataset, mMARCO (Bonifacio et al., 2021), and broaden our analysis to also include multilingual IR (MLIR). We provide an in-depth analysis of *"monolingual overfitting"* where the ranker learns features, such as exact keyword matches, which are useful in MoIR but do not transfer well to CLIR and MLIR due to the lack of lexical overlap (Section 6.4.2). Our work is in line with Roy et al. (2020), who show for bi-encoders that representations from zero-shot models are weakly aligned between languages, i.e., models prefer non-relevant documents in the same language over relevant documents in a different language. To address this problem, we propose to use code-switching as an inductive bias to regularize monolingual overfitting in CEs.

Generation of synthetic code-switched data has served as a way to augment data in cross-lingual setups in a number of NLP tasks (Singh et al., 2019; Einolghozati et al., 2021; Tan and Joty, 2021). They utilize substitution techniques ranging from simplistic re-writing in the target script (Gautam et al., 2021), looking up bilingual lexicons (Tan and Joty, 2021) to MT (Tarunesh et al., 2021). Previous work on improving zero-shot transfer for IR includes weak supervision (Shi et al., 2021), tuning the pivot language (Turc et al., 2021), multilingual query expansion (Blloshmi et al., 2021) and cross-lingual pre-training (Yang et al., 2020a; Yu et al., 2021a; Yang et al., 2022a; Lee et al., 2023). In this regard, code-switching is complementary to existing approaches. Our work is most similar to Shi et al. (2020), who use bilingual lexicons for full term-by-term translation to improve MoIR. Concurrent to our work, Huang et al. (2023) showed that code-switching improves the retrieval performance on low-resource languages, however, their focus lies on CLIR with English documents. As argued in Section 6.4.1, evaluating zero-shot rankers trained on English data on target language pairs that also involve English is not truly reflective of their zero-shot transfer capabilities and leads to overestimating retrieval performance. To the best of our knowledge, we are the first

to systematically investigate (1) artificial code-switching to train CEs and (2) the interaction between MoIR, CLIR and MLIR.

**Contributions.** Our key contributions and findings are summarized as follows:

**(1)** We validate our previous findings on ad-hoc document-level retrieval (Section 6.4.2) on a different CLIR task (answer passage retrieval) and show that training zero-shot rankers on monolingual data indeed leads to monolingual overfitting. That is, we empirically show that rankers trained on English data exhibit a better performance when transferred to a monolingual setup and worse performance when transferred to a cross-lingual setup (Section 7.4).

**(2)** We show that training on artificially code-switched data is an effective way to regularize monolingual overfitting and improve the transfer performance of zero-shot cross-lingual and multilingual rankers (Section 7.4). Our findings reveal that the gains are largest for difficult queries that have no lexical overlap with their relevant documents. The performance on queries that share a significant portion of tokens with their relevant documents is not negatively affected by our code switching.

**(3)** We demonstrate that our approach is robust towards different ratios of code-switched tokens and consistently improves the performance when models are transferred into a cross-lingual setup. Our results on MLIR further show that code-switching improves the effectiveness in generalizing to unseen languages.

**Resource-Lean Transfer.** We follow the same method as in the previous chapter and focus on zero-shot cross-lingual transfer of rerankers. This approach is *resource-lean* as it relies only *monolingual task supervision* and little cross-lingual supervision in the form of *bilingual dictionaries*, which can be obtained in a fully unsupervised fashion (see Section 2.2.3). Our goal is to use bilingual dictionaries to improve the generalization performance of CEs for CLIR and MLIR. Most importantly, and contrary to resource-intensive approaches, we do not rely on any large-scale parallel resources (see Taxonomy in Chapter 1, Figure 1.4).

## 7.2 Methodology

**Reranking with Cross-Encoders.** We follow the standard cross-encoder reranking approach (CE) proposed by Nogueira and Cho (2019), which formulates relevance prediction as a sequence pair (query-document pair) classification task. CEs are composed of an encoder model and a relevance prediction model. The encoder is a pre-trained language model (Devlin et al., 2019) that transforms the concatenated input `[CLS] Q [SEP] D [SEP]` into a joint query-document feature representation, from which the classification head predicts relevance. Finally, documents are reranked according to their predicted relevance. We argue that fine-tuning CEs on monolingual data biases the encoder towards encoding features that

**Algorithm 1** Multilingual Code Switching

**Input:** Language pool $\mathbf{L}$, dictionaries $\mathcal{D}_{\text{EN}\to\text{X}}$, transl. probability $p$, tokenized sequence $s$

**for** every position $i = 1 \dots |s|$ **do**
  $p_{cs} \sim U(0,1)$ ▷ Sample CS probability (uniform random).
  **if** $p_{cs} < p$ **then**
    $l_2 \sim U(\mathbf{L})$ ▷ Sample target language $l_2$ from $\mathbf{L}$.
    $s_i \leftarrow \mathcal{D}_{\text{EN}\to l_2}(s_i)$
  **end if**
**end for**
**return** perturbed sequence $s$

are only useful when the target setup is MoIR. To mitigate this bias, we propose to perturb the training data with code-switching, as described next.

**Artificial Code-Switching.** While previous work has studied code-switching (CS) as a natural phenomenon where speakers borrow words from other languages (e.g. anglicism) (Ganguly et al., 2016; Wang and Komlodi, 2018), we here refer to code-switching as a method to *artificially* modify monolingual training data. In the following we assume availability of English (EN–EN) training data. The goal is to improve the zero-shot transfer of ranking models into cross-lingual language pairs X–Y by training on code-switched data $\text{EN}_\text{X}$–$\text{EN}_\text{Y}$ instead, which we obtain by exploiting bilingual lexicons similar to Tan and Joty (2021). We now describe two CS approaches based on lexicons: one derived from word embeddings and one from Wikipedia page titles (see examples in Table 7.1).

**Code-Switching with Word Embeddings.** We rely on bilingual dictionaries $\mathcal{D}$ induced from cross-lingual word embeddings (Mikolov et al., 2013b; Heyman et al., 2017b) and compute for each EN term its nearest (cosine) cross-lingual neighbor. In order to generate $\text{EN}_\text{X}$–$\text{EN}_\text{Y}$ we then use $\mathcal{D}_{\text{EN}\to\text{X}}$ and $\mathcal{D}_{\text{EN}\to\text{Y}}$ to code-switch query and document terms from EN into the languages X and Y, each with probability $p$. This approach, dubbed Bilingual CS (**`BL-CS`**), allows a ranker to learn inter-lingual semantics between EN, X and Y. In our second approach, Multilingual CS (**`ML-CS`**), we additionally sample for each term a different target language into which it gets translated; we refer to the pool of available languages as seen languages (see Algorithm 1).

**Code-Switching with Wikipedia Titles.** Our third approach, **`Wiki-CS`**, follows (Lan et al., 2020; Fetahu et al., 2021) and uses bilingual lexicons derived from parallel Wikipedia page titles obtained from inter-language links. We first extract word $n$-grams from queries and documents with different sliding window of sizes $n \in \{1, 2, 3\}$. Longer $n$-gram are favored over shorter ones in order to account for multi-term expressions, which are commonly observed in named entities. In `Wiki-CS` we create a single multilingual dataset where queries and documents from different training instances are code-switched into different languages.

| Approach | Query | Document |
|---|---|---|
| `Zero-Shot` | What is an affinity credit card program? | Use your PayPal Plus credit card to deposit funds. If you have a PayPal Plus credit card, you are able to instantly transfer money from it to your account. This is a credit card offered by PayPal for which you must qualify. |
| `Fine-tuning` | Was ist ein Affinity-Kreditkartenprogramm? | Используйте свою кредитную карту PayPal Plus для внесения средств. Если у вас есть кредитная карта PayPal Plus, вы можете мгновенно переводить деньги с нее на свой счет. Это кредитная карта, предлагаемая PayPal, на которую вы должны претендовать. |
| `BL-CS` | Denn is einem affinity credit card programms? | Использовать your PayPal плюс кредита билет попытаться депозиты funds. если you have a PayPal плюс credit билет, скажите are able to instantly переход денег from it попытаться ваши account. This is a credit билет offered by paypal for причём you может qualify. |
| `ML-CS` | What is это affinità credit card program? | Use jouw PayPal Plus credit geheugenkaarten to depositi funds. إذا you хотя ein الائتمان aggiunta credit card, you are попытаться quindi sofort transfer geld from questo إلى deine account. Это является a кредита card offerto by paypal voor which you devono للتأهل |
| `Wiki-CS` | What is an affinity Kreditkarte program? | Use your PayPal Plus carta di credito to deposit funds. If you have a PayPal Plus carta di credito, you are able to instantly transfer denaro from it to your account. This is a carta di credito offered by PayPal for which you mosto qualify. |

**Table 7.1:** Different Code-Switching strategies on a single training instance for the target language pair DE–RU (Query ID: 711253, Document ID: 867890, label: 0). **`Zero-shot`:** Train a single zero-shot ranker on the original EN–EN MS MARCO instances (Bajaj et al., 2016). **`Fine-tuning`:** Fine-tune ranker directly on DE–RU, we use translations (Google Translate) provided by the mMARCO dataset Bonifacio et al. (2021). **Bilingual Code-Switching (`BL-CS`)**: Translate randomly selected EN query tokens into DE and randomly selected EN document tokens into RU, each token is translated with probability $p = 0.5$; **Multilingual Code-Switching (`ML-CS`)**: Same as `BL-CS` but additionally sample for each token its target language uniformly at random. **`Wiki-CS`**: Translate $n$-grams extracted with a sliding window. Tokens within a single query/document are code-switched with a single language; across training instances languages are randomly mixed. We use the following language pool of "seen languages": English, German, Russian, Italian, Dutch, Arabic.

## 7.3 Experimental Setup

**Models and Dictionaries.** We follow Bonifacio et al. (2021) and initialize rankers with the multilingual encoder mMiniLM provided by Reimers and Gurevych (2020) on HuggingFace (Wolf et al., 2020).[2] The maximum sequence length corresponds to 512 and we train our ranking models for a fixed number of 200,000 steps with a learning rate of 2e-5 and a batch size of 64. Following prior work by Reimers and Gurevych (2020), we extract negative samples from training triplets provided by MS MARCO (Bajaj et al., 2016) with a positive to negative ratio of 1:4. In the passage re-ranking task we rank for 6980 queries 1,000 passages respectively (qrels.dev.small). For `BL-CS` and `ML-CS` we use off-the-shelf multilingual MUSE embeddings[3] to induce bilingual lexicons (Lample et al., 2018), which have been aligned with initial seed dictionaries of 5k word translation pairs. We set the translation probability $p = 0.5$. For `Wiki-CS`, we use the lexicons provided by the linguatools project.[4]

**Baselines.** To compare whether training on CS'ed data $\text{EN}_\text{X}$–$\text{EN}_\text{Y}$ improves the transfer into CLIR setups, we include the zero-shot ranker trained on EN–EN as our main baseline (henceforth, `Zero-shot`). Our upper-bound reference, dubbed `Fine-tuning`, refers to ranking models that are directly trained on the target language pair X–Y, i.e. no zero-shot transfer. Following Roy et al. (2020), we adopt the *translate test* baseline and translate any test data into EN using our bilingual lexicons induced from word embeddings. On this data we evaluate both the `Zero-shot` baseline ($\texttt{Zero-shot}_{\text{Translate Test}}$) and our `ML-CS` model ($\texttt{ML-CS}_{\text{Translate Test}}$).

**Datasets and Evaluation.** We use the publicly available multilingual MARCO (mMARCO) dataset (Bonifacio et al., 2021), which includes fourteen different languages. We group those into six seen languages (EN, DE, RU, AR, NL, IT) and eight unseen languages (HI, ID, IT, JP, PT, ES, VT, FR) and construct a total of 36 language pairs.[5] Out of those, we construct setups where we have documents in different languages (EN–X), queries in different languages (X–EN), and both in different languages (X–X). Specifically, for each document ID (query ID) we sample the content from one of the available languages. For evaluation, we use the official evaluation metric MRR@10.[6] All models re-rank the top 1,000 passages provided for the passage re-ranking task. We report all results as averages over three random seeds.

[2] nreimers/mMiniLMv2-L6-H384-distilled-from-XLMR-Large
[3] https://github.com/facebookresearch/MUSE
[4] https://linguatools.org/wikipedia-parallel-titles
[5] Due to computational limitations we don't exhaustively evaluate on all possible language pairs.
[6] We use the implementation provided by the `ir-measures` package (MacAvaney et al., 2022).

| | EN | DE | RU | AR | NL | IT | AVG | $\Delta_{ZS}$ |
|---|---|---|---|---|---|---|---|---|
| `Zero-shot` | 35.0 | 25.9 | 23.8 | **23.9** | 27.2 | 26.9 | 25.5 | - |
| `Fine-tuning` | 35.0 | 30.3* | 28.5* | 27.2* | 30.8* | 30.9* | 29.5 | +4.0 |
| `Zero-shot`$_{\text{Translate Test}}$ | - | 22.5* | 18.2* | 17.7* | 24.7* | 23.3* | 21.3 | -4.2 |
| `ML-CS`$_{\text{Translate Test}}$ | - | 22.8* | 18.6* | 17.7* | 24.7* | 24.5* | 21.7 | -3.8 |
| `BL-CS` | - | **26.0** | **25.5** | 23.0 | **27.5** | **27.2** | **25.8** | **+0.3** |
| `ML-CS` | 34.0 | 25.9 | 24.7 | 21.3 | 27.2 | 26.9 | 25.2 | -0.3 |
| `Wiki-CS` | 33.8* | 25.6 | 24.1 | 20.5* | 27.0 | 25.5* | 24.5 | -1.0 |

**Table 7.2:** MoIR: Monolingual results on mMARCO languages and averaged over all languages (excluding EN) in terms of MRR@10. **Bold**: Best zero-shot performance for each language. $\Delta_{ZS}$: Absolute difference to `Zero-shot`. Results significantly different from `Zero-shot` are marked with * (paired t-test, Bonferroni correction, $p < 0.05$).[7]

## 7.4 Results and Discussion

We observe that code-switching improves cross-lingual and multilingual re-ranking, while not impeding monolingual setups, as shown next.

**Transfer into MoIR vs. CLIR.** We first quantify the performance drop when transferring models trained on EN–EN to MoIR as opposed to CLIR and MLIR. Our analysis is comparable to our experimental setup described in Section 6.4.2 where we refer to the former as same-task transfer (MoIR→MoIR) and to the latter as cross-task transfer (MoIR→CLIR). Comparing `Zero-shot` results between different settings we find that the average MoIR performance of 25.5 MRR@10 (Table 7.2) is substantially higher than CLIR with 15.7 MRR@10 (Table 7.3) and MLIR with 16.6 MRR@10 (Table 7.4). The transfer performance greatly varies with the language proximity, in CLIR the drop is larger for setups involving typologically distant languages (AR–IT, AR–RU), to a lesser extent the same observation holds for MoIR (AR–AR, RU–RU). This is consistent with previous findings made in other syntactic and semantic NLP tasks (He et al., 2019; Lauscher et al., 2020). It is also consistent with two key observations made in Section 6.4.2: (1) Cross-task transfer performs substantially worse than same-task transfer, and (2) the performance drops are largest when non-Latin scripts are involved. Furthermore, the performance gap to `Fine-tuning` on translated data is much smaller in MoIR (+4 MRR@10) than in CLIR (+11.1 MRR@10) and MLIR (+8.3 MRR@10). Our aim to is close this gap between zero-shot and full fine-tuning in a resource-lean way by training on code-switched queries and documents.

**Code-Switching Results.** Training on code-switched data consistently outperforms zero-shot models in CLIR and MLIR (Table 7.3 and Table 7.4). In AR–IT and

| | EN→X | | | | DE→X | | | AR→X | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | DE | IT | AR | RU | IT | NL | RU | IT | RU | AVG | $\Delta_{ZS}$ |
| `Zero-Shot` | 24.0 | 23.0 | 14.0 | 18.3 | 15.0 | 19.7 | 12.9 | 7.7 | 7.1 | 15.7 | - |
| `Fine-tuning` | 29.7* | 30.5* | 26.5* | 28.0* | 26.9* | 27.9* | 25.5* | 23.9* | 22.7* | 26.8 | +11.1 |
| $\texttt{ZS}_{\text{Translate Test}}$ | 22.8 | 23.2 | 16.4 | 17.0 | 15.8 | 17.5 | 11.8 | 9.8 | 8.7 | 15.9 | +0.2 |
| $\texttt{ML-CS}_{\text{Translate Test}}$ | 24.9 | 24.6 | 17.9* | 19.5 | 17.6 | 19.3* | 14.3 | 12.2* | 10.6* | 17.9 | +2.2 |
| `BL-CS` | **26.9*** | **27.3*** | 19.3* | 22.8* | **20.4*** | **22.8*** | 17.8* | **15.6*** | 14.1* | **20.8** | **+5.1** |
| `ML-CS` | 26.5* | 26.4* | 18.1* | 22.1* | 19.8* | **22.8*** | 17.8* | 15.3* | 14.2* | 20.3 | +4.6 |
| `Wiki-CS` | 26.2* | 26.4* | **19.4*** | **22.9*** | 19.4* | 22.4* | **18.3*** | 14.4* | 14.1* | 20.4 | +4.7 |

**Table 7.3:** CLIR: Cross-lingual results on mMARCO in terms of MRR@10.

| | Seen Languages | | | | | All Languages | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | X–EN | EN–X | X–X | $\text{AVG}_{\text{seen}}$ | $\Delta_{\text{seen}}$ | X–EN | EN–X | X–X | $\text{AVG}_{\text{all}}$ | $\Delta_{\text{all}}$ |
| `Zero-shot` | 19.0 | 23.5 | 16.3 | 19.6 | - | 16.5 | 20.8 | 12.9 | 16.6 | - |
| `Fine-tuning` | 24.8* | 26.4* | 21.1* | 24.1 | +4.5 | 26.5* | 26.5* | 21.9* | 25.0 | +8.3 |
| `ML-CS` | **24.2*** | 25.9* | **21.1*** | **23.7** | +4.1 | **21.6*** | 23.2* | 17.0* | 20.6 | +3.9 |
| `Wiki-CS` | 23.6* | **26.0*** | 20.6* | 23.4 | +3.8 | 21.3* | **23.8*** | **17.1*** | **20.7** | **+4.0** |

**Table 7.4:** MLIR: Multilingual results on mMARCO in terms of MRR@10. Left: Six seen languages for which we used bilingual lexicons to code-switch training data. Right: All fourteen languages included in mMARCO.

AR–RU we see improvements from 7.7 and 7.1 MRR@10 up to 15.6 and 14.1 MRR@10, rendering our approach particularly effective for distant languages. Encouragingly, Table 7.2 shows that the differences between both of our CS approaches (`BL-CS` and `ML-CS`) versus `Zero-shot` is not statistically significant, showing that gains can be obtained without impairing MoIR performance. Table 7.3 shows that specializing one zero-shot model for multiple CLIR language pairs (`ML-CS`, `Wiki-CS`) performs almost on par with specializing one model for each language pair (`BL-CS`). The results of `Wiki-CS` are slightly worse in MoIR and on par with `ML-CS` on MLIR and CLIR.

**Translate Test vs. Code-Switch Train.** In monolingual retrieval results presented in Table 7.2 both $\texttt{Zero-shot}_{\text{Translate Test}}$ and $\texttt{ML-CS}_{\text{Translate Test}}$ underperform compared to other approaches. This shows that zero-shot rankers work better on clean monolingual data in the target language than noisy monolingual data in English. In CLIR, where *Translate Test* bridges the language gap between X and Y, we observe slight improvements of +0.2 and +2.2 MRR@10 (Table 7.3). However, in both MoIR and CLIR *Translate Test* consistently falls behind code-switching at training time.

**Ablation: Translation Probability.** The translation probability $p$ allows us to control the ratio of code-switched tokens to original tokens, with $p = 0.0$ we

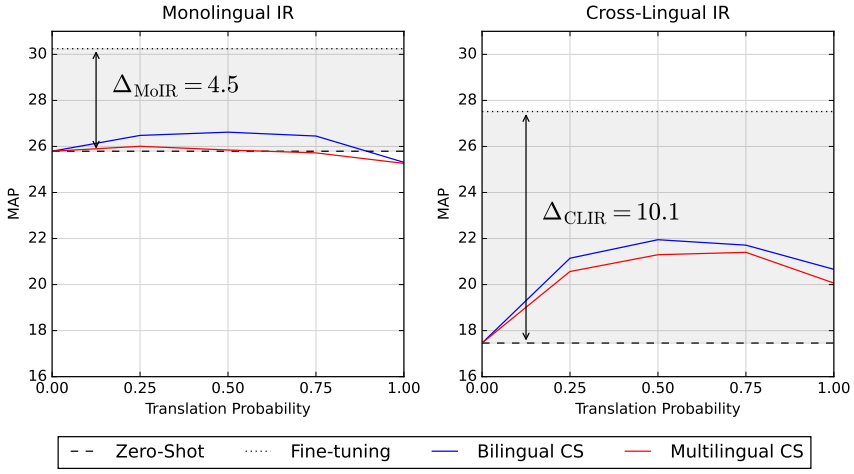


**Figure 7.1:** Retrieval performance in terms of mean average precision (MAP) for different translation probabilities, averaged across all language pairs (cf. Appendix C). $\Delta_{\text{MoIR}}$ and $\Delta_{\text{CLIR}}$ are computed on average results before rounding.

default back to the `Zero-shot` baseline, with $p = 1.0$ we attempt to code-switch every token. [8] Figure 7.1 (right) shows that code-switching a smaller portion of tokens is already beneficial for the zero-shot transfer into CLIR. The gains are robust towards different values for $p$. The best results are achieved with $p = 0.5$ and $p = 0.75$ for `BL-CS` and `ML-CS`, respectively. Figure 7.1 (left) shows that the absolute differences to `Zero-shot` are much smaller in MoIR.

**Monolingual Overfitting.** Exact matches between query and document keywords are strong relevance signals in MoIR but do not transfer well to CLIR and MLIR due to mismatching vocabularies. Training zero-shot rankers on monolingual data biases rankers towards learning features that cannot be exploited at test time. Code-Switching reduces this bias by replacing exact matches with translation pairs,[9] steering model training towards learning interlingual semantics instead. To investigate this, we group queries by their average token overlap with their relevant documents and evaluate each group separately on MLIR.[10] The results are shown in Table 7.5. Unsurprisingly, rankers work best when there is significant overlap between query and document tokens. However, the performance gains resulting from training on code-switched data (`ML-CS`) are most pronounced for queries

[8] Due to out-of-vocabulary tokens the percentage of translated tokens is slightly lower: 23% for $p = 0.25$, 45% for $p = 0.5$, 68% for $p = 0.75$ and 92% for $p = 1.0$. In Wiki CS 90% of queries and documents contain at least one translated n-gram, leading to 20% of translated tokens overall.

[9] We analyzed a sample of 1M positive training instances and found a total of 4,409,974 overlapping tokens before and 3,039,750 overlapping tokens after code-switching (`ML-CS`, $p = 0.5$), a reduction rate of ~31%.

[10] We use the model's SentencePiece tokenizer (Kudo and Richardson, 2018) and ignore the special tokens `<s>`, `</s>`, `<pad>`, `<unk>` and `<mask>`.

| | EN–X | X–EN | X–X |
|---|---|---|---|
| *No Code Switching* (Zero-Shot) | | | |
| No overlap | 12.2 | 11.0 | 7.4 |
| Some overlap | 29.7 | 22.4 | 19.9 |
| Significant overlap | 44.6 | 36.4 | 45.5 |
| All queries | 23.5 | 19.0 | 16.3 |
| *Multilingual Code Switching* (ML-CS) | | | |
| No overlap | 15.5 (+3.3) | 17.8 (+6.8) | 13.0 (+5.6) |
| Some overlap | 31.7 (+2.0) | 27.2 (+4.8) | 25.3 (+5.4) |
| Significant overlap | 44.7 (+0.2) | 37.8 (+1.4) | 45.1 (-0.5) |
| All queries | 25.9 (+2.4) | 24.2 (+5.3) | 21.1 (+4.8) |

**Table 7.5:** MLIR results on seen languages (MRR@10) broken down into queries that share no common tokens (no overlap), between one and three tokens (some overlap) and more than three tokens (significant overlap) with their relevant documents. Gains of ML-CS are shown in brackets. EN–X has 3,116 queries with no overlap, 3,095 with some overlap and 769 with significant overlap. X–EN has 3,147 queries with no overlap, 2,972 with some overlap and 861 with significant overlap. X–X has 3,671 queries with no overlap, 2,502 with some overlap and 807 with significant overlap.

| | Unseen QL | | Unseen DL | | |
|---|---|---|---|---|---|
| | FR–EN | ID–NL | EN–PT | DE–VT | IT–ZH |
| Zero-shot | 18.3 | 13.7 | 23.2 | 10.9 | 9.4 |
| Fine-tuning | 30.0* | 27.2* | 30.8* | 24.8* | 25.0* |
| ML-CS | **21.4*** | **18.3*** | 25.9* | **15.5*** | 14.8* |
| Wiki-CS | 21.0* | 17.2* | **26.2*** | 15.4* | **15.0*** |

| | Unseen Both | | | | | |
|---|---|---|---|---|---|---|
| | ES–FR | FR–PT | ID–VT | PT–ZH | AVG | $\Delta_{ZS}$ |
| Zero-shot | 19.0 | 18.7 | 11.8 | 9.6 | 15.0 | - |
| Fine-tuning | 29.0* | 29.0* | 25.8* | 25.4* | 27.4 | +12.2 |
| ML-CS | **22.7*** | **21.9*** | **16.4*** | 14.7* | **19.1** | **+4.1** |
| Wiki-CS | 21.9* | 20.5* | 15.3* | **14.8*** | 18.6 | +3.4 |

**Table 7.6:** CLIR results on unseen mMARCO languages in terms of MRR@10. Results include unseen query languages (QL), unseen document languages (DL) and unseen languages on both sides.

with some token overlap (up to +5.4 MRR@10) and no token overlap (up to +6.8 MRR@10). On the other hand, the gains are much lower for queries with more than three overlapping tokens and range from -0.5 to +1.4 MRR@10. This supports our hypothesis that code-switching indeed regularizes monolingual overfitting.

| | FR | ID | ES | PT | ZH | VT | AVG | $\Delta_{ZS}$ |
|---|---|---|---|---|---|---|---|---|
| BM25 | 15.5 | 14.9 | 15.8 | 15.2 | 11.6 | 13.6 | 14.3 | -12.0 |
| Zero-shot | **27.2** | **26.8** | **28.2** | **27.9** | **24.8** | 22.8 | 26.3 | - |
| Fine-tuning | 30.5* | 30.6* | 31.5* | 31.2* | 29.1* | 28.6* | 30.3 | +4.0 |
| ML-CS | 26.4 | 26.7 | 27.6 | 27.3 | 22.3 | **23.1*** | 25.6 | -0.7 |
| Wiki-CS | 25.8* | 25.5* | 27.1* | 26.5* | 22.2* | 21.8* | 24.8 | -1.8 |

**Table 7.7:** MoIR: Monolingual results on unseen mMARCO languages in terms of MRR@10.

**Multilingual Retrieval and Unseen Languages.** Here we compare how code-switching fares against Zero-shot on languages to which neither model has been exposed to at training time. Table 7.4 shows the gains remain virtually unchanged when moving from six seen (+4.1 MRR@10 / +3.8 MRR@10) to fourteen languages including eight unseen languages (+3.9 MRR@10 / +4.0 MRR@10). Results in Table 7.6 and Table 7.7 confirm that this holds for unseen languages on the query, document and both sides, suggesting that the best pivot language for zero-shot transfer (Turc et al., 2021) may not be monolingual but a code-switched language. On seen languages ML-CS is close to MT (Fine-tuning).

## 7.5 Conclusion

We propose a simple and effective method to improve zero-shot rankers: training on artificially code-switched data. We empirically test our approach on 36 language pairs, spanning monolingual, cross-lingual, and multilingual setups. Our method outperforms zero-shot models trained only monolingually and provides a resource-lean alternative to MT for CLIR. Importantly, training zero-shot rankers on code-switched training queries and documents yields largest gains for those queries that have no token overlap to their relevant documents while maintaining stable performance in other setups. These results suggest that our approach is effective in narrowing the gap between zero-shot reranking and full fine-tuning, i.e. regularizing monolingual overfitting. In MLIR our approach can match MT performance in some setups while relying only on bilingual dictionaries. To the best of our knowledge, this work is the first to propose artificial code-switched training data for cross-lingual and multilingual IR. We make our code and resources publicly available at: https://github.com/MaiNLP/CodeSwitchCLIR

# Chapter 8

# Parameter-Efficient Cross-Lingual Transfer

[1]In this chapter, we follow the same retrieval paradigm as in the previous two chapters, i.e., we transfer cross-encoder rerankers trained on English retrieval data to monolingual retrieval tasks other languages (MoIR) as well as cross-lingual retrieval tasks (CLIR) in a zero-shot fashion. As discussed in Section 2.3.3, zero-shot cross-lingual transfer (ZS-XLT) is facilitated by multilingual pre-trained language models (mPLM). We show that two *parameter-efficient* approaches for cross-lingual transfer, Sparse Fine-Tuning Masks (SFTMs) (Ansell et al., 2022) and Adapters (Pfeiffer et al., 2020), allow for a *more lightweight* and *more effective* zero-shot transfer compared to the standard approach solely based on mPLMs. We first train language adapters (or SFTMs) via Masked Language Modelling and then train retrieval (i.e., reranking) adapters (SFTMs) stacked on top, while keeping all other parameters frozen. This modular design allows us to compose rerankers at inference time by applying the ranking adapter (or SFTM) trained with source language data together with the language adapter (or SFTM) of a target language. We evaluate our models on the CLEF 2003 and HC4 benchmarks and, as another contribution, extend the former with queries in three new languages: Kyrgyz, Uyghur and Turkish. The proposed parameter-efficient methods for CLIR outperform the standard zero-shot transfer approach with full mPLM fine-tuning, while being more modular and reducing training times. The gains are particularly pronounced for low-resource languages, where our approaches also substantially outperform (i.e. improve the first-stage ranking of) the competitive machine translation-based rankers.

[1]This chapter is adapted from: **Robert Litschko**, Ivan Vulić, and Goran Glavaš. 2022. Parameter-efficient neural reranking for cross-lingual and multilingual retrieval. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pages 1071–1082, Gyeongju, Republic of Korea.

## 8.1 Introduction

Fine-tuning cross-encoders based on mPLMs on English data enables, in principle, downstream zero-shot cross-lingual transfer (ZS-XLT) to any language seen by the mPLM during pre-training (e.g., for mBERT, 104 languages). However, in language understanding tasks (Hu et al., 2020), massive performance drops have been observed when models are transferred to low-resource languages and languages that typologically distant from English (i.e., languages which are underrepresented during LM pre-training) (Lauscher et al., 2020). These findings are in line with our results presented in Section 6.4 and Section 7.4, where we show that ZS-XLT exhibits substantially worse performance when rerankers are transferred from English into the typologically distant languages Finnish and Russian.

Underrepresented languages suffer from the so-called *curse of multilinguality* (Conneau et al., 2020): sharing mPLM parameters (i.e., its fixed parameter budget/capacity) between an increasing number of languages at some point deteriorates the quality of text representations. Chang et al. (2024) find that low-resource languages initially benefit from multilingual data, however, only up to a certain point. In ZS-XLT, full fine-tuning on large-scale source language data (typically English) is prone to forgetting and interference effects (McCloskey and Cohen, 1989; Mirzadeh et al., 2020), which can also harm the quality of the multilingual representation space. Besides the standard zero-shot cross-lingual transfer (MacAvaney et al., 2020b; Huang et al., 2021a), other cross-lingual transfer approaches have been applied in CLIR. They include training data translation (Shi et al., 2020), leveraging external word-level alignments (Huang et al., 2021b), and distant supervision for pretraining CLIR models (Yu et al., 2021b). While approaches based on translation have shown competitive for high-resource languages, they are not applicable for low-resource languages for which reliable MT models are missing. Also, translation-based cross-lingual transfer has been shown to suffer from unwanted artifacts such as "translationese" (Zhao et al., 2020b).

Even if one would have sufficient amounts of labeled data in target languages, training language- or language-pair specific neural rerankers for all languages and language pairs would be prohibitively computationally expensive and unsustainable (Strubell et al., 2019). In this chapter, we propose to compose rerankers at retrieval time in a modular way, which enables a more sustainable cross-lingual transfer for CLIR. To this end, we introduce neural reranking models based on parameter efficient fine-tuning (Han et al., 2024). Specifically, our rerankers are based on two styles of modular components: **1)** *Adapters* (Rebuffi et al., 2017; Houlsby et al., 2019; Pfeiffer et al., 2020) and **2)** *Sparse Fine-Tuning Masks* (SFTMs) (Ansell et al., 2022). When integrated into the architecture of a mPLM, both allow for (1) the pre-trained multilingual knowledge to be fully preserved, alleviating the negative interference and forgetting effects, and (2) offer additional language-specific model capacity which is used to improve the models' internal representations for target languages, thus remedying for the curse of multilinguality. To the best of our knowledge, we are the first to investigate the effectiveness of

parameter-efficient transfer methods on CLIR. We provide an extensive evaluation of both approaches in zero-shot transfer for monolingual retrieval (MoIR) and on two standard CLIR benchmarks (Braschler, 2004; Lawrie et al., 2022).

**Contributions.** The main contributions presented in this chapter are three-fold:

**(1)** We introduce modular cross-encoder (CE) re-rankers based on Adapters and Sparse Fine-tuning Masks (SFTMs). By introducing language and task specific adapters (SFTMs) we modularize CEs and decouple specializing rankers for target language semantics from learning to rank.

**(2)** As an additional contribution, we expand the CLEF dataset (Braschler, 2004) and release three new query languages from the Turkic family (Turkish, Kyrgyz, and Uyghur, the latter two being low-resource languages), which are typologically and etymologically distant from Indo-European languages.[2]

**(3)** Our results on CLIR and MoIR show that modular neural rerankers are not only faster to train, but they can also outperform standard ZS-XLT based on fully fine-tuning all mPLM parameters, and especially so in retrieval tasks that involve linguistically distant and low-resource languages. Our rerankers also generally outperform a strong preranker that utilizes machine translation (MT).

**Resource-Lean Transfer.** In this chapter, we investigate ZS-XLT for CLIR (and MoIR) and decompose the retrieval task into two subtasks: learning retrieval-specific features (i.e., learning-to-rank) and learning language-specific features (i.e., language acquisition). For the first subtask, we only use monolingual task supervision to train task adapters (SFTMs). The second task relies on self-supervised learning and requires unlabeled monolingual data in the target language(s) to train language adapters (SFTMs). Both types of resources are easier to obtain than large-scale parallel data (required to train machine translation models) or large-scale CLIR training data, i.e. direct supervision in the target language pair (see Figure 1.4 in Chapter 1).

## 8.2 Methodology

In this chapter, we follow the multi-stage ranking paradigm discussed in Section 6.2. That is, we focus on transferring cross-encoder reranking models. In this context, we introduce adapters and sparse fine-tuning masks (SFTMs), and present how to leverage them as crucial vehicles of the parameter-efficient cross-lingual transfer of the reranking component.

[2] In this manner, our work addresses the calls for more linguistic diversity in NLP and IR research (Bender, 2011; Joshi et al., 2020a; Ponti et al., 2020; Ruder et al., 2021).

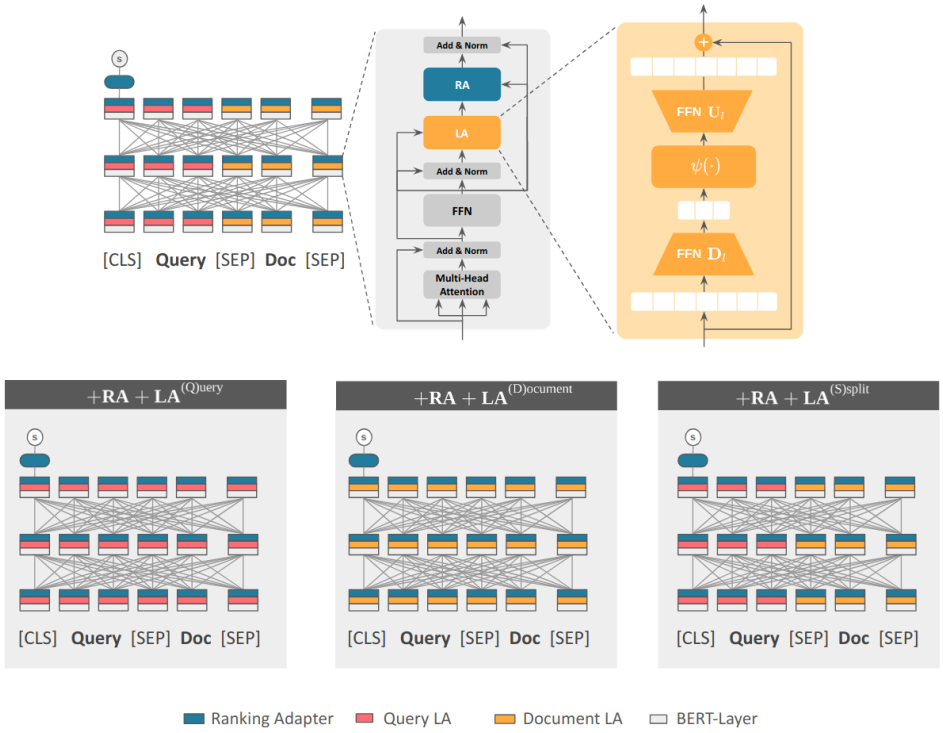


**Figure 8.1:** Overview of parameter-efficient transfer learning for neural (re)ranking with adapters.: A reranker is composed by stacking a pre-trained target Language Adapter (LA) and a Ranking Adapter (RA; trained with source language data) on top of original Transformer layers of an mPLM such as, e.g., multilingual BERT (Devlin et al., 2019).

**Parameter-Efficient Cross-Lingual Ranker Transfer.** We now describe our proposed modular and parameter-efficient framework that allows faster training and more effective cross-lingual transfer of neural rerankers, which we apply on both ZS-XLT for CLIR and MoIR. We first learn language-specific Adapters (LAs) or Sparse Fine-Tuning Masks (SFTMs) via Masked Language Modelling (MLM) on unannotated monolingual corpora of respective languages, while keeping the original mPLM parameters intact. We then train Ranking Adapters (or Ranking SFTMs) using source-language data on top of the source-language LAs (language SFTMs), while keeping all other parameters frozen. At inference time, for a given IR (MoIR or CLIR) task, we compose our reranker by placing the Ranking Adapters (Ranking SFTMs) on top of the LAs (language SFTMs) of the query and/or document languages of that concrete retrieval task.

**Adapters.** Figure 8.1 illustrates our cross-encoder architecture based on Adapters. We train Ranking Adapters (RA) and Language Adapters (LA) based on the architecture of Pfeiffer et al. (2020). In the Transformer architecture, each layer $l$

consists of a multi-head attention block (i.e., sub-layer) and a feed-forward network (FFN), both followed by a residual connection and layer normalization. We refer the reader to Section 2.3 for more details. We denote the residual connection (output of FFN) with $\mathbf{r}_l$ and the hidden state after the layer norm with $\mathbf{h}_l$.

$$\text{LA}(\mathbf{h}_l, \mathbf{r}_l) = \mathbf{U}_l(\psi(\mathbf{D}_l(\mathbf{h}_l)) + \mathbf{r}_l \tag{8.1}$$

$$\text{RA}(\mathbf{h}_l, \mathbf{r}_l) = \mathbf{U}_l(\psi(\mathbf{D}_l(\text{LA}_l))) + \mathbf{r}_l \tag{8.2}$$

Adapters are parameterized by the down-projection matrix $\mathbf{D} \in \mathbb{R}^{h \times d}$ and the up-projection matrix $\mathbf{U} \in \mathbb{R}^{d \times h}$, where $h$ and $d$ denote the hidden size of the Transformer and the bottleneck dimension of the adapter. The ratio between $h$ and $d$ is also called *reduction factor* and corresponds to the level of parameter compression (i.e., how many times fewer parameters are updated if we train adapters instead of updating all Transformer parameters). The forward pass of a Language Adapter consists of a down-projection of $h_l$, a non-linear activation function $\psi(\cdot)$ and an up-projection. Ranking Adapters are stacked on top of LAs and process their output. Both adapters have residual connections to the output of the FFN.[3] We train LAs using the standard MLM objective (Devlin et al., 2019), and we train RAs together with the dense scoring layer using the binary cross-entropy loss.

In CLIR tasks, queries and documents are in different languages. It is thus, in principle, possible to stack the RA on top of (i) the query language adapter $\text{LA}^{\text{Q}}$, (ii) document language adapter $\text{LA}^{\text{D}}$, or by using *split adapters* $\text{LA}^{\text{S}}$: here, we encode query tokens up to the separator token (`[SEP]`) and the other half with the LA of the respective language (see Figure 8.1).

**Sparse Fine-Tuning Masks.** Like adapters, SFTMs (Ansell et al., 2022) aim to decouple task knowledge from language knowledge, but instead of introducing additional parameters, the idea is to directly update only small subsets of mPLM's original parameters (see Figure 8.2). Sparse Fine-Tuning (SFT) consists of two phases. In *Phase 1* we fine-tune all mBERT's parameters $\theta^{(0)}$, resulting in updated parameter values $\theta^{(1)}$. We then select the top $K$ parameters with the largest value change, i.e., those with the largest $|\theta_i^{(0)} - \theta_i^{(1)}|$ values. We then construct a binary mask: the selected $K$ parameters remain trainable, whereas all other parameters are frozen. In *Phase 2* all parameters are reset to $\theta^{(0)}$ and training restarts, but this time only the selected parameters of the mask are updated, yielding $\theta^{(2)}$. The final update (i.e., the SFTM) is then obtained as the difference vector $\text{M} = \theta^{(2)} - \theta^{(0)}$. As is the case with Language Adapters, we obtain the Language Masks (LM) by means of (additional) masked language modeling (MLM) training on language-specific corpora; whereas the Ranking Mask (i.e., the mask for the ranking task, RM) is learned via binary cross-entropy objective on source-language (English)

[3]To alleviate the mismatch between the multilingual vocabulary of the mPLM and the target language vocabulary, Pfeiffer et al. (2020) also additionally place invertible adapters INV on top of the embedding layer along with their inverses $\text{INV}^{-1}$ placed before the output layer. For more details we refer the reader to (Pfeiffer et al., 2020). In our experiments we adopt this variant.

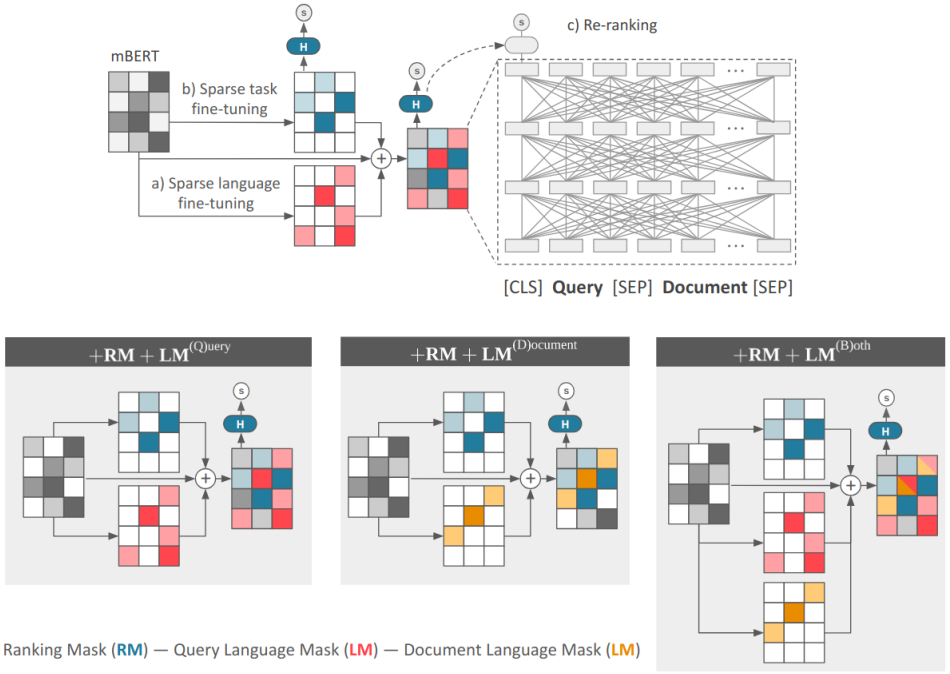


**Figure 8.2:** Overview of parameter-efficient transfer learning for neural (re)ranking with SFTMs: Sparse fine-tuning of a Ranking Mask (RM) and a Language Mask (LM) from mBERT parameters; rerankers are composed by adding the RM and LM values to original mBERT parameters.

relevance judgments. At inference, the reranker is composed as $\theta^{(0)} + \text{RM} + \text{LM}$ (cf., Figure 8.2). In our CLIR settings (Section 8.3), we explore using (i) the query language mask ($\text{LM}^{\text{Q}}$), (ii) document language mask ($\text{LM}^{\text{D}}$) or (iii) the combination of both masks ($\text{LM}^{\text{B}} = \text{LM}^{\text{Q}} + \text{LM}^{\text{D}}$). Note that SFTMs represent a more computationally efficient solutions at inference time, because, unlike adapters, they do not change (i.e., deepen) the Transformer architecture itself.

## 8.3 Experimental Setup

**Adapter and SFTM Training.** We train adapters following the recommendations from Pfeiffer et al. (2020). Unless noted otherwise, we train LAs with the reduction factor of 2 (i.e., $h/d = 2$) on Wikipedias of respective languages, for 250K steps with batch size 64 and learning rate of 1e-4. For RA we report the results with a reduction factor of 16 (Section 8.4.1). RM and LM have a reduction factor of 2. In Section 8.4.2, we ablate the robustness towards different reduction factors: 1, 2, 4, 8, 16, and 32. Following Ansell et al. (2022), for fair comparisons between adapters and SFTMs, we set the mask size $K$ for SFTMs to the same number of parameters that adapters with a certain reduction factor have (see Appendix D.1).

**Reranking Training.** We train mBERT-based[4] rerankers on relevance annotations from MS MARCO (Craswell et al., 2021b), with a linear warm-up over the first 5K updates, in batches of 32 instances with a maximum sequence length of 512 and using a learning rate of 2e-5. We evaluate the model on the validation data every 25K updates and choose the checkpoint with the best validation performance. Once trained, our cross-encoder models rerank top $k = 100$ documents returned by our prerankers (described below). Finally, we also study ensembling the preranker's and reranker's ranked lists via simple rank averaging (ENS) (cf. Section 4.4.1).

**Evaluation Data.** In this chapter, we extend our standard evaluation outlined in Section 3.3. We additionally experiment on the recently introduced HC4 benchmark (Lawrie et al., 2022). HC4 comprises queries *and* document collections in three languages: Persian (FA), Russian (RU) and Chinese (ZH). Compared to CLEF 2003, HC4 collections are much larger, spanning 646K, 486K and 4.72M documents per each respective language, associated with 50 test queries in English and each target language respectively. Consistent with our approach for CLEF, and following (Lawrie et al., 2022), we use *title* and *description* fields as queries. We further evaluate our models in CLIR tasks with CLEF queries posed in lower-resource languages. To this end, (i) we leverage Swahili (SW) and Somali (SO) queries (Bonab et al., 2019), where the queries were obtained via manual translation of English queries; (ii) we create another set of translated CLEF queries in three languages: Turkish (TR), Kyrgyz (KG), and Uyghur (UG). The new set covers one high-resource and two low-resource languages and is intended to facilitate and diversify evaluation of CLIR with low-resource languages in future work. The queries were constructed via the standard post-editing procedure borrowed from other data collection tasks (Glavaš et al., 2020; Ding et al., 2022, e.g.): we had native speakers post-edit query translations obtained from Google Translate.

**Baseline Models.** The primary baseline for our Adapter- and SFTM-based transfer is the standard zero-shot transfer approach discussed in Chapter 6. That is, we use cross-encoders and fine-tune all its parameters by training on MS MARCO. For CLIR experiments, we continue to opt for $\text{DISTIL}_{\text{DmBERT}}$ as our best bi-encoder preranker (PR) on the CLEF dataset (see results presented in Section 5.5). We also couple a state-of-the-art neural MT system of (Fan et al., 2021a) (FAIR-MT), which we use to translate queries to the document language, with the BM25 ranker in the target language.[5] For Kyrgyz and Uyghur, we use another NMT model, provided by the Turkic Interlingua (TIL) community[6] (Mirzakhalov et al., 2021) since we failed to obtain meaningful {KG, UG}$\rightarrow l_2$ translations with FAIR-MT.

[4]Pre-trained `bert-base-multilingual-uncased` weights from the HuggingFace Transformers library (Wolf et al., 2020) are used.

[5]We used the `pyserini` implementation of BM25 (Lin et al., 2021a) with the suggested default parameter configuration.

[6]https://turkic-interlingua.org

| | TR→ X | | | | | DE→X | | |
|---|---|---|---|---|---|---|---|---|
| Model | EN | IT | DE | FI | RU | FI | IT | RU |
| $\text{DIST}_{\text{DmBERT}}$ (PR) | .183 | .251 | .190 | .252 | .260 | .300 | .267 | .284 |
| MonoBERT | .235 | .197 | .208 | .285 | .217 | .329 | .270 | .246 |
| +RA +$\text{LA}^{\text{Split}}$ | .269 | **.253** | **.252*** | **.362** | .186 | .329 | .300 | .223 |
| +RA +$\text{LA}^{\text{Doc}}$ | .252 | .234 | .222 | .267 | **.267** | .350 | .302 | **.315** |
| +RA +$\text{LA}^{\text{Query}}$ | **.270** | .243 | .242 | .293 | .191 | .325 | .279 | .223 |
| +RM +$\text{LM}^{\text{both}}$ | .229 | .228 | .197 | .244* | .168 | .309 | .302 | .191* |
| +RM +$\text{LM}^{\text{Doc}}$ | .231 | .226 | .229 | .317 | .149* | .376 | .304 | .187 |
| +RM +$\text{LM}^{\text{Query}}$ | .239 | .252 | .232 | .316 | .162* | **.391** | **.323*** | .195 |

| | EN→X | | | | FI→X | | | |
|---|---|---|---|---|---|---|---|---|
| Model | FI | IT | RU | DE | IT | RU | **AVG** | **ENS** |
| $\text{DIST}_{\text{DmBERT}}$ (PR) | .294 | .290 | .313 | .247 | .221 | .302 | .261 | - |
| MonoBERT | .339 | .315 | .254 | .295 | .197 | .174 | .254 | .274 |
| +RA +$\text{LA}^{\text{Split}}$ | .363 | .352 | .197 | .317* | **.266** | .207 | .277 | .287 |
| +RA +$\text{LA}^{\text{Doc}}$ | .366* | **.366*** | **.248** | .314* | .220 | **.234** | **.283** | **.298** |
| +RA +$\text{LA}^{\text{Query}}$ | .370 | .355 | .189 | .318 | .247 | .182 | .266 | .285 |
| +RM +$\text{LM}^{\text{both}}$ | .299 | .344 | .181* | .303 | .206 | .108* | .236 | .269 |
| +RM +$\text{LM}^{\text{Doc}}$ | **.394*** | .359 | .173* | **.321*** | .239 | .166* | .262 | .279 |
| +RM +$\text{LM}^{\text{Query}}$ | .359 | .349 | .191 | .310* | .255* | .160 | .267 | .280 |

**Table 8.1:** CLIR results (Mean Average Precision, MAP) with $\text{DIST}_{\text{DmBERT}}$ as preranker. **Bold**: Best neural retrieval model for each language pair. *: significance tested against MonoBERT at $p \leqslant 0.05$, computed via paired two-tailed t-test. We report average results (AVG) and averaged ensemble (ENS) results.

## 8.4 Results and Discussion

We first discuss our main retrieval results for CLIR and MoIR in Section 8.4.1. In Section 8.4.2, we further analyze (i) the trade-off between retrieval speed and retrieval effectiveness in Adapter-based models, and (ii) the impact of different reduction factors for Adapters and SFTMs. In the following, we use superscripts over LAs and LMs denote query language (Q), document language (D), split adapters (S) for LAs, and '(B)oth masks' for LMs (see Section 8.2).

### 8.4.1 Document-Level CLIR and MoIR

**Cross-Lingual Retrieval (CLIR).** Tables 8.1 and 8.2 show the CLIR results, for fourteen language pairs from the augmented CLEF 2003 benchmark[7] using $\text{DISTIL}_{\text{DmBERT}}$ and NMT+BM25 as first-stage retrievers, respectively. With our

[7]We add TR-* pairs to the evaluation, enabled by our EN→TR translations of the queries.

| | TR→X | | | | | DE→X | | |
|---|---|---|---|---|---|---|---|---|
| Model | EN | IT | DE | FI | RU | FI | IT | RU |
| NMT+BM25 (PR) | .392 | .353 | .308 | .307 | .227 | .367 | .385 | .272 |
| MonoBERT | .415 | .375 | .339 | .345 | .307 | .409 | .38 | .322 |
| +RA +LA | .448 | .408 | .353 | .371 | .327 | .413 | .405 | .348 |
| +RM +LM | .447 | .414 | .356 | .386 | .336 | .468 | .407 | .363 |

| | EN→X | | | | FI→X | | | |
|---|---|---|---|---|---|---|---|---|
| Model | FI | IT | RU | DE | IT | RU | **AVG** | **ENS** |
| NMT+BM25 (PR) | .378 | .446 | .285 | .355 | .364 | .271 | .326 | - |
| MonoBERT | .386 | .411 | .351 | .371 | .367 | .34 | .362 | .360 |
| +RA +LA | .388 | .435 | .367 | .385 | .381 | .365 | .384 | .360 |
| +RM +LM | .413 | .429 | .345 | .390 | .395 | .364 | .397 | .374 |

**Table 8.2:** CLIR results (Mean Average Precision, MAP) with NMT+BM25 as Stage 1 preranker. For modular rerankers, we report the numbers with the best-performing configurations from CLEF experiments: +RA +$LA^{D}$ and +RM +$LM^{Q}$; see also the caption of Table 8.1.

preranker $DISTIL_{DmBERT}$ (Table 8.1), Adapter- and SFTM-based rerankers consistently improve the initial preranking results, with gains of up to 2.7 MAP points and EN-RU as the only exception. Compared to full fine-tuning (MonoBERT), our modular reranking variants bring gains between 1 and 4 MAP points on average, across all language pairs. Interestingly, the best adapter configuration (RA +$LA^{D}$, in which at inference we stack the RA on top of the LA of the document collection language) outperforms the best SFTM-based reranker (RM +$LM^{Q}$ and RM +$LM^{D}$) by 1.6 MAP points. Somewhat surprisingly, adapting only to the language of the document collection ($LA^{D}$; $LM^{D}$) yields better performance than adapting to both the query and collection language of the target task ($LA^{S}$; $LM^{B}$).

The language pairs in Tables 8.1 and 8.2 consist of high-resource languages for which large parallel corpora and, consequently, reliable NMT models exist. However, even when starting from a more competitive translation-based preranker (NMT+BM25; Table 8.2), our modular cross-lingual transfer of the reranker yields performance gains. In fact, with this stronger preranker, the gains from modular reranking are even more pronounced: +5/+6 MAP points for Adapters and SFTMs, respectively, compared to preranker and +2/+3 compared to MonoBERT. This could explain why interpolating between the preranking and reranking (ENS, last column) yields further gains with $DISTIL_{DmBERT}$ as the preranker (Table 8.1), but not when we prerank with NMT+BM25 (Table 8.2).

Table 8.3 shows the CLIR results for (a) language pairs from extended CLEF with queries written in low-resource languages – Swahili and Somali queries created by (Bonab et al., 2019), as well as our newly introduced query languages Kyrgyz and Uyghur; and (b) three cross-lingual pairs of arguably distant languages (EN-{Farsi, Chinese, Russian}) from the HC4 benchmark (Lawrie et al., 2022).

| | X→EN | | | | EN→X | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | SW | SO | KG | UG | FA | ZH | RU | **AVG** | **ENS** |
| NMT+BM25 (PR) | .325 | .157 | .228 | .091 | .183 | .113 | .186 | .183 | - |
| MonoBERT | .362 | .158 | .255 | .157 | .246 | .172 | .218 | .224 | .216 |
| +RA + $LA^{Doc}$ | **.407** | **.166** | .305 | .155 | .259 | .189 | .234 | .245 | **.228** |
| +RM + $LM^{Doc}$ | .389 | .161 | **.311** | **.165** | **.267** | **.196** | **.241** | **.247** | .225 |

**Table 8.3:** CLIR results on extended CLEF pairs with low-resource query languages (Swahili, Somali, Kyrgyz, and Uyghur) and three language pairs from the HC4 benchmark.

| | CLEF 2003 | | | | | HC4 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | EN | FI | DE | IT | RU | FA | ZH | RU | **AVG** | **ENS** |
| BM25 (PR) | .480 | .505 | .434 | .494 | .361 | .279 | .196 | .228 | .372 | - |
| MonoBERT | .464 | .528 | .444 | .463 | .363 | .356 | .283 | .245 | .393 | .402 |
| +RA + LA | .512 | .537 | .457 | .495 | **.389** | .372 | .284 | .261 | .413 | .410 |
| +RM + LM | **.515** | **.564** | **.459** | **.502** | .379 | **.398** | **.307** | **.264** | **.423** | **.417** |

**Table 8.4:** Results of zero-shot cross-lingual transfer for monolingual retrieval (MoIR) on CLEF 2003 and HC4 datasets. Results with reduction factors of 16 and 2 for Adapters and SFTMs, respectively.

The gains that our SFTM- and Adapter-based modular rerankers bring for language pairs involving low-resource languages, over the MT-based preranker and the full fine-tuning (MonoBERT), are generally more substantial than those for high-resource language pairs: e.g., +8 and +4 MAP points (w.r.t. NMT+BM25 and MonoBERT, respectively) for SW-EN, +8 and +5 points for KG-EN. The gains are similarly prominent for more distant language pairs from the HC4 dataset (+8 MAP points over the NMT+BM25 preranker for EN-FA and EN-ZH). With such prominent gains of the modular reranking over the preranker, it is no surprise that averaging the preranking and reranking document ranks (ENS) reduces the performance of the reranker. We believe that these results in particular emphasize the effectiveness of modular cross-lingual transfer that allows to increase the capacity of mPLMs for individual languages, by means of LMs or LAs. The representations of low-resource languages for which mPLMs have seen little in pre-training, particularly suffer from the curse of multilinguality (Conneau et al., 2020; Lauscher et al., 2020) – this is why, we believe, we generally see particularly prominent gains for those languages when we increase the mPLMs capacity for their representation via LMs or LAs.

**Cross-Lingual Transfer for Monolingual Retrieval (MoIR).** Table 8.4 displays the results of monolingual retrieval with our best-performing modular rerankers for

| Layer | CLIR | MoIR | AVG | Latency | Δ Speed-Up | Δ MAP |
|---|---|---|---|---|---|---|
| None | .282 | .418 | .331 | 34.6 ms | - | - |
| 1-2 | .295 | .412 | .337 | 33.7 ms | +2.6% | +.006 |
| 1-4 | .269 | .395 | .314 | 32.8 ms | +5.0% | −.017 |
| 1-6 | .229 | .375 | .281 | 31.9 ms | +7.7% | −.050 |
| 1-8 | .134 | .284 | .187 | 31.0 ms | +10.4% | −.143 |
| 1-10 | .086 | .210 | .130 | 30.0 ms | +12.9% | −.200 |
| 1-12 | .086 | .208 | .129 | 29.5 ms | +14.2% | −.201 |

**Table 8.5:** Trade-off between efficiency and effectiveness when dropping adapters in +RA + $LA^D$. Average over all CLIR/MoIR setups and all reduction factors.

EN (as the source language) and four target languages (DE, IT, FI, RU).[8] Unlike the fully fine-tuned reranker (MonoBERT), our modular Adapter- and SFTM-based rerankers consistently improve the initial rankings produced by BM25. These results strengthen the finding that our modular rerankers are not just more parameter-efficient (i.e., faster to train), but also lead to better cross-lingual transfer due to decoupling of language- and ranking-specific knowledge. In MoIR tasks the SFTM-based transfer outperforms its Adapter-based counterpart, same as in the case of CLIR with NMT+BM25 preranking (Table 8.1). Also, as in the case of the latter CLIR results (Tables 8.1 and 8.3), interpolating between preranking and reranking results does not bring any gains.

It is worth noting that all MoIR scores are substantially higher than CLIR results from Tables 8.1 and 8.2. This is expected and consistent with our findings in Chapter 6 and 7, and it reflects the fact that matching representations within a language – where models can still rely on exact lexical matches between queries and documents – is easier than aligning text representations across languages.

### 8.4.2 Further Analysis

**Effectiveness vs Efficiency.** Adapters increase query latency because they deepen the Transformer (cf. Section 2.3.1). Rücklé et al. (2021) show that one can drop adapters from lower layers with little effect on performance. Table 8.5 shows the results of a similar analysis, where we drop the adapters from the first $N$ layers at inference. Dropping adapters from only the first two layers (row 1-2) only slightly decreases the MoIR performance whereas it even slightly increases the CLIR results. Dropping adapters from more layers, however, substantially reduces the retrieval performance: e.g., removing adapters from the first 10 layers reduces CLIR performance by almost 20 MAP points, while reducing the query latency by only 13%. While Adapters and SFTMs yield comparable performance in our experi-

[8]Note that in MoIR, the actual reranking is always monolingual (albeit in the target language). Both queries and documents are thus encoded with the same target language LA/LM.

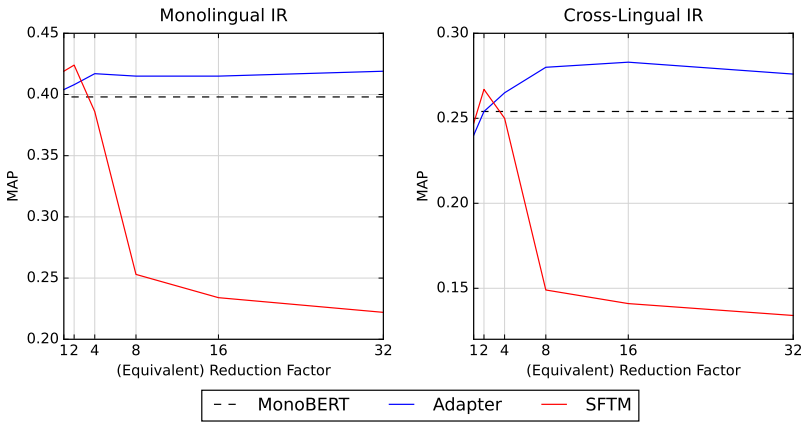


**Figure 8.3:** Retrieval performance at different parameter reduction factors; average MAP performance for MoIR (left, PR: BM25) and CLIR (right, PR: $\text{DIST}_{\text{DmBERT}}$).

ments, these observations favor SFTMs: for the *same query latency*,[9] SFTMs will yield better performance.

**Parameter Efficiency.** We now investigate the relation between various levels of parameter efficiency and retrieval performance. Figure 8.3 shows the performance of our modular rerankers for different (equivalent) parameter reduction factors.[10] For completeness, we also show the average zero-shot results obtained with MonoBERT on MoIR CLIR (Table 8.1) and (Table 8.4). We find that SFTMs exhibit stronger performance with smaller reduction factors (2 and 4), i.e., when we update a larger percentage of mBERT's original parameters. SFTMs shift the pretrained values of mBERT's parameters: this constrains the range of values that individual parameters can take, requiring the modification of the larger number of parameters for injecting complex language- and ranking-specific knowledge. In contrast, Adapters show better performance with higher reduction factor (8, 16, 32), i.e., when we add a relatively smaller number of Adapter parameters. This could be the consequence of the "unconstrained" initialization of the new Adapter parameters, which allows the complementary language- and ranking-specific knowledge to be compressed into a smaller number of parameters.

**Impact of NMT on CLIR.** In the cross-lingual setup the quality of retrieved documents crucially depends on the quality of query translations when NMT is used. This holds especially true in multi-stage retrieval, where the performance of rerankers is directly related to the recall of first-stage retrievers. In Table 8.6 we show original English queries together with their respective translations from

[9] The query latency of an SFTM-based reranker is the same as that of MonoBERT as it does not increase the number of total parameters in the mPLM.

[10] Appendix D.1 lists for each adapter reduction factor the equivalent reduction factor of SFTMs (i.e. their degree of sparsity); detailed results can be found in in Table D.3 and Table D.2.

| QID | English Query *(original)* | NMT: Swahili → English | NMT: Somali → English |
|---|---|---|---|
| 151 | Wonders of **Ancient World** Look for **information** on the **existence** and/or the discovery of remains of the seven wonders of the **ancient world**. | Search for **information** about the **existence** and/or development of the **seventh** universe of the **ancient world**. | Thus, therefore, it is necessary to bear in mind that the truth is the truth, and that the truth is the truth, and that the truth is the truth. |
| 172 | **1995** Athletics **World Records** What **new world records** were achieved during the **1995** athletic world championships in **Gothenburg**? | What **new world records** were recorded at the **1995 World** Horses in **Gothenburg**? | The **1995 World** Trade Organization (WTO) announced that a **new** international trade agreement has led to a global trade agreement in **Gothenburg**. |
| 187 | **Nuclear** Transport in **Germany** Find reports on the protests against the transportation of **radioactive** waste with **Castor containers** in **Germany**. | **Nuclear** Delivery in **Germany** A report on the anti-trafficking of **radioactive** pollutants and **Castor containers** in **Germany**. | The Nugleerka department of Jarmalka Hel has been prepared for the development of the Nugleerka department of **Castor** district in Jarmalka. |
| 200 | Flooding in Holland and **Germany** Find statistics on flood disasters in Holland and **Germany** in **1995**. | The floods in the Netherlands and **Germany** have recorded the floods in the Netherlands and **Germany** in **1995**. | The Netherlands Federation and the United Nations have agreed with the Netherlands Federation and the Netherlands Federation in **1995**. |

**Table 8.6:** Comparison between original CLEF queries and translations from Swahili and Somali to English. Tokens that occur both in the original query and translations are highlighted in **bold** (ignoring case, excluding stopwords). Color highlights for different translation artifacts: hallucinations, topic shifts, slight lexical/semantic variations and copied source words.

Swahili and Somali. As expected, translations from Swahili are generally of higher quality compared to Somali, which explains the big performance gap reported in Table 8.3. In the best case the translation is semantically very close to the original query (cf., SW→EN; QID:172), or it contains only slight lexical (*flooding* vs. *floods*) and semantic variations, e.g., near-synonyms (*Holland* vs. *Netherlands*). In other cases, error propagation from NMT impacts CLIR performance to different extents. Those include, e.g., missing keywords (*statistics*; QID:200), topic shifts (*sports* vs. *business*; SO→EN, QID:172) or queries consisting of unrelated text and repetitions (i.e., 'hallucinations'; SO→EN, QID:151, QID:200). Especially repetitions and hallucinations[11] are known unwanted artifacts in NMT (Fu et al., 2021; Raunak et al., 2021) and can cause retrieval models to emphasize unrelated keywords by inflating their term frequency.[12] Lastly, in cases where source words are copied instead of translated, e.g., *Nugleerka (Nuclear)* or *Jarmalka (Germany)* in QID:187, neural retrieval models need to rely on imperfect internal alignment of word translations (Cao et al., 2020).

[11] This phenomenon has been reported to occur in low-resource and out-of-domain settings (Müller et al., 2020). We confirm this finding as we find hallucinations appearing more often in EN→SO than in EN→SW query translations.

[12] Further investigation of NMT+BM25 on SO→EN reveals that manually filtering out queries containing more than two repetitions/hallucinations leaves us with 22 remaining queries on which results improve from 0.157 to 0.280 MAP.

## 8.5 Conclusion

In this chapter, we introduce a modular and parameter-efficient approach for the zero-shot cross-lingual transfer of rerankers. Our models, based on Adapters and Sparse Fine-Tuning Masks, allow for decoupling of language-specific and task-specific (i.e., ranking) knowledge. We demonstrate that this leads to more effective transfer to cross-lingual IR setups as well as to better cross-lingual transfer for monolingual retrieval in target languages with no relevance judgment improving over strong prerankers based on state-of-the-art NMT. Encouragingly, we observe particularly pronounced gains for low-resource languages included in our evaluation. It is important to notice that the effectiveness of rerankers varies with the degree of sparsity (reduction factor). We hope that our results will encourage a broader investigation of parameter-efficient neural retrieval in monolingual and cross-lingual setups. We make our code, adapters, SFTMs and query translations available at: https://github.com/rlitschk/ModularCLIR.

# Part IV

# Resource-Lean Transfer with Multiple Encoders

# Chapter 9

# Expert Model Selection (Proof of Concept)

[1]In the previous chapters, we evaluate multilingual pre-trained language models (mPLM) in the bi-encoder and cross-encoder paradigms for cross-lingual information retrieval (CLIR). As discussed in (Conneau et al., 2020), these models are pre-trained on imbalanced data where high-resource languages have a larger presence than low-resource languages. The authors further show that, consequently, mPLMs are limited in the number of languages they can reliably encode, which they refer to as *"curse of multilinguality"*. This means, for example, that text from underrepresented languages is split into more subwords.[2] Suboptimal tokenization is known to adversely impact the representation quality (Hangya et al., 2022) and downstream task performance for NLP tasks on low-resource languages and typologically distant languages (Lauscher et al., 2020; Pfeiffer et al., 2021; Rust et al., 2021). It also impacts processing long document in CLIR, where the effective maximum sequence length, i.e. the maximum number of words a model encodes relates to a languages' presence in the pre-training corpus. In this chapter, we aim to conceptually investigate the feasibility of (supervised) IR model selection in the presence of multilingual training data (i.e., multiple monolingual training data). Since training a *single multi-source model* on the concatenation of all source languages would suffer from the curse of multilinguality we propose to separately train *multiple monolingual experts* instead. A routing model then predicts (i.e., selects) the best expert model for each input instance (e.g., query). To do so, the routing model is trained to estimate the experts' performance, and then forwards each input to the model(s) with the highest estimated performance.

[1]This chapter is adapted from: **Robert Litschko**, Ivan Vulić, Željko Agić, and Goran Glavaš. 2020. Towards Instance-Level Parser Selection for Cross-Lingual Transfer of Dependency Parsers. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 3886–3898, Barcelona, Spain (Online).

[2]For example, Uyghur CLEF queries are on average $3.7\times$ longer when tokenized into subwords instead of by whitespace (see Table 3.2 in Chapter 3). In contrast, English subword-tokenized queries are only $1.2\times$ longer than whitespace-tokenized queries (see also Appendix B.2).

At the time of writing, there existed no large-scale retrieval benchmark that contains a large number of languages, including low-resource test languages. We therefore present a *proof of concept* on the example task of *delexicalized cross-lingual transfer of dependency parsers*. For this task, we train source parsers (experts) on 42 training languages from Universal Dependencies dataset (UD v2.3) and evaluate them on 20 diverse and unseen low-resource test languages. In Section 9.1, we describe the general model selection framework. In Section 9.4, we propose a concrete implementation: instance-level parser selection (ILPS), where we train a supervised regression model (i.e., routing model), to predict parser accuracies for individual part-of-speech sequences (i.e., POS-sequences). We compare ILPS against two strong single-best parser selection baselines (SBPS) where we select the best parser at the treebank level and apply it to all test instances.

## 9.1 Introduction

The imbalance of languages in (pre-)training corpora naturally lead to a scenario where some languages are overrepresented while other languages are underrepresented. Conneau et al. (2020) find that multilingual pre-trained language models (mPLMs) suffer from the so-called *curse of multilinguality*. That is, language models have a limited capacity in terms of the number of languages they can reliably encode. Consequently, their zero-shot cross-lingual transfer (ZS-XLT) performance is substantially lower when the target language is typologically different from English (Lauscher et al., 2020). This problem can be remedied, for example, by adding language-specific model parameters such as language adapters (Chapter 8; Pfeiffer et al., 2020, 2022), or by adjusting the vocabulary space of mPLMs to account for underrepresented languages (Wang et al., 2019c; Chung et al., 2020; Liang et al., 2023).

We take a different approach and combine multiple models for different languages. Here, each model is fine-tuned on a single language (or few related languages). We refer those models as expert models and train them independently, thereby avoiding negative interference (Wang et al., 2020b) between high-resource and low-resource languages. At test time, we select the model with the highest expected performance among a pool of expert models. Estimating the expected transfer performance for a given model and target task is a well-studied problem known as *model transferability estimation* (Ding et al., 2024). In the following, we reuse the notation and problem formulation from Ding et al. (2024). We denote a pool of $n_{\mathcal{M}}$ source models $\phi_i$ and their respective training datasets $\mathcal{D}_i^S$ as $\mathcal{M} = \{\phi_i, \mathcal{D}_i\}_{i=1}^{n_{\mathcal{M}}}$. Estimating the performance of a set of candidate models on a given target dataset $\mathcal{D}^T$ yields a set of corresponding transferability scores $\mathcal{S} = \{s_i\}_i^{n_{\mathcal{M}}}$, which capture how well each model is expected to transfer to $\mathcal{D}^T$. We can then compute the true test set accuracies by evaluating each model on the gold labels $\mathcal{A} = \{Acc(\phi_i, \mathcal{D}^T)\}_{i=1}^{n_{\mathcal{M}}}$. Finally, the performance of a transferability estimation method is measured as the correlation between the estimated and

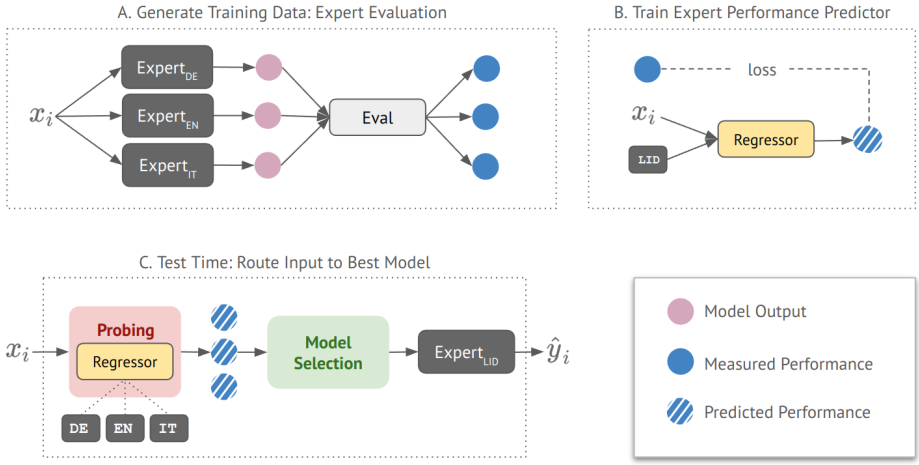


**Figure 9.1:** General instance-based model retrieval framework.

the true transfer performance $C = Cor(\mathcal{S}, \mathcal{A})$. A reliable method for estimating model transferability can be used to "probe" each model and select the best source model for a given target dataset (Chen and Ritter, 2021).[3] We refer to this type of transfer as *single-source transfer*. In a different approach, the multi-source transfer paradigm, a single multilingual model is trained on the concatenation of all datasets from multiple source languages (Lim et al., 2024). In cross-lingual retrieval tasks, instances (i.e. queries) can be written in different low-resource unseen languages (ZS-XLT for CLIR). Motivated by this, we extend the above-described framework to *instance-level transfer* with the goal of selecting the best source model (IR system) for each instance (query). However, at the time of writing, there exists no IR dataset with training splits in many languages (although this is expected to improve over time), so we provide a proof-of-concept study on a task for which training data does exist for many languages – syntactic parsing (see below).

Figure 9.1 illustrates the general framework. For a given set of source languages, we assume access to a set of monolingual expert models. We train a routing model which forwards every input instance to the expert with the highest expected performance. Suppose we have three expert models $\phi_{\text{DE}}$ and $\phi_{\text{EN}}$, and $\phi_{\text{IT}}$. We generate training data for the routing model by measuring the performance of each expert on every other expert's training data (Step A). More precisely, we apply $\phi_{\text{DE}}$ on the training data of $\phi_{\text{EN}}$ and record for each instance $i$ the actual performance $s_i$, and repeat this process for all language pair combinations. Next, in Step B, we train a Transformer-based (Vaswani et al., 2017) regression model (router) to predict $s_i$ from a given input instance $x_i$ and a representation of the expert model

[3]Our work is similar to (Chen and Ritter, 2021). However, the authors assume English training data and fine-tune multiple mBERT-based models (Devlin et al., 2019) under different hyperparameters and random seeds, whereas we assume training data in multiple languages and train multiple monolingual models. We additionally emphasize model selection on the instance-level.

(language embedding). At inference time (Step C), the routing model first predicts the expected performance of every expert (probing) and then selects the model with the best performance estimate (model selection). In the ZS-XLT paradigm input instances $x_i$ are written in unseen test languages. In this modular framework, we train all models (experts, router) independently, which, in addition to mitigating negative interference between languages, also allows us to integrate additional experts models (e.g., off-the-shelf models or external API-based models) while only updating the routing model.

**Proof of Concept.** At the time of writing, there exists no retrieval dataset containing a large number of high-resource languages (with sufficiently large training splits) and low-resource test languages. We therefore present a *proof of concept* on the example task of *delexicalized dependency parsing* where we use the Universal Dependencies dataset (UDv2.3; Nivre et al., 2018). In this simplified setting, the routing model and all parsers (expert models) operate on a shared delexicalized input space: sequences of universal parts-of-speech tags (i.e., UPOS-sentences). This way, and contrary to subword tokenization, we adopt an input space where all languages are treated equally. In cross-lingual transfer of parsers one can either (1) choose the best parser from a set of available parsers, trained on treebanks of various resource-rich languages (*single-best parser selection*, SBPS), or (2) use a parser trained on a mixture of treebanks of (ideally related) resource-rich languages (*multi-source parser transfer*, MSP). Both SBPS and MSP rely on some measure of structural alignment between languages in order to select either the single best source language parser (SBPS) or a set of (syntactically related) source languages (MSP). Existing solutions rely on measures like the Kullback–Leibler (KL) divergence between source- and target-language distributions of POS trigrams (Rosa and Žabokrtský, 2015), which can be unreliable for small target language corpora or instance-level estimation. More recent approaches (Agić, 2017; Lin et al., 2019) choose suitable source languages based on manually coded typological similarities between languages available from databases such as WALS (Dryer and Haspelmath, 2013) or URIEL (Littell et al., 2017). Unlike MSP and SBPS, the idea of our proposed framework is to select the source-language parser for each target instance, dubbed *instance-level parser selection* (ILPS), rather than to use the same parser for all target language instances (as SBPS and MSP do). This is motivated by a simple observation that different source parsers provide most accurate parses for different target POS-sequences (Section 9.3). We empirically show that an oracle ILPS leads to major potential gains compared to an oracle single-best parser selection at the treebank level (SBPS).

We perform a large-scale evaluation of delexicalized dependency parser transfer, encompassing 42 source languages with large(r) treebanks, and 20 target (i.e., test) languages with small(er) treebanks from the Universal Dependencies (UD) v2.3 collection (Nivre et al., 2018). We show that, averaged across all test treebanks, our simple ILPS model substantially outperforms strong SBPS baselines

(Rosa and Žabokrtský, 2015; Lin et al., 2019). We further demonstrate that we can easily aggregate instance-level predictions into an SBPS model, yielding improvements over the existing SBPS baselines for 16/20 and 17/20 test languages. Finally, we show that by ensembling the parses of few-best parsers according to the ILPS model's predictions we can outperform (1) the multi-source parser trained on the treebanks of all 42 source languages and (2) even surpass the performance of an oracle single-best treebank-level parser selection (i.e., oracle SBPS).

**Contributions.** The key contributions of this chapter are summarized as follows:

**(1)** We propose an instance-level (language) expert model selection framework where a routing model estimates for each instance the performance of each (independently trained) expert model and routes it to the expert(s) with the largest predicted performance.

**(2)** On the example of delexicalized dependency parsing, we present a specific implementation of the general framework: instance-level parser selection (ILPS). Using oracle models we show that there is a large gap between selecting the best parser at the treebank-level and instance-level.

**(3)** We provide a large-scale evaluation and compare ILPS against (i) a single best parser selection (SBPS) baseline that selects the best parser based on syntactic similarity, (ii) a SBPS baseline where the parser is selected based on POS n-gram similarity, and (iii) a multi-source parser trained on the concatenation of all treebanks.

**Resource-Lean Transfer.** In this chapter, we follow the ZS-XLT paradigm, i.e. we do not assume any labeled data in the target languages. Different from previous chapters (Chapters 6 to 8), where we transferred a single multilingual model, we now use multiple monolingual models. By training multiple expert models we utilize monolingual training data in multiple languages without risking negative interference between languages. This is a resource-lean approach since monolingual training data is cheaper to obtain than cross-lingual task supervision. From a practical perspective, it allows us to integrate the large number of publicly available retrieval models into a unified framework (cf. Figure 1.4 in Chapter 1) without collecting any training data.

## 9.2 Related Work

In this section, we first discuss two lines of research related to model performance prediction and model selection without access to labeled data. We then discuss mixture-of-expert models originally proposed in (Jacobs et al., 1991) and prior work related to our proof-of-concept task. The work presented in this chapter is also closely related to the task of model transferability estimation, we refer the reader to the recent survey by Ding et al. (2024).

**Query Performance prediction.** A central component of our instance-level expert model selection framework is the routing model, which estimates the expected performance of a model on a given instance. In IR, the task of estimating the expected performance of a retrieval system is known as *query performance prediction* (QPP) (Faggioli et al., 2023c,b). As discussed in (Carmel and Yom-Tov, 2010), QPP methods can be broadly distinguished between (i) pre-retrieval methods which predict the performance based on linguistic cues and corpus-level statistics and (ii) post-retrieval QPP which predict the performance based on retrieved documents. For example, BERT-QPP$_{\text{cross}}$ (Arabzadeh et al., 2021) concatenates a given query with a list of top-k retrieved documents and predicts a score that reflects ranking evaluation measures such as average prediction (cf. Section 3.3). NQA-QPP (Arabzadeh et al., 2021) further includes the standard deviation of top-k pointwise retrieval scores as an additional indicator of query performance. Fusion-based QPP (Roitman, 2018) predicts the performance of ensemble rankings based on the individual rankings to be fused. Our framework is conceptually similar to pre–retrieval QPP, since we do not use the output of expert models to predict their performance. In cross-lingual retrieval, early QPP methods (Kishida et al., 2004b; Kishida, 2008) focus CLIR systems based on machine translation (MT) and train a regression model to predict CLIR performance according to (i) the quality of query translations and (ii) the level of difficulty of a search topic. Our work instead uses semantic representations of the model input (input embeddings) and representations of expert models (e.g., parser embeddings).

**Machine Translation Quality Estimation.** Prior to the paradigm shift from lexical to neural information retrieval, cross-lingual IR (CLIR) systems primarily relied on machine translation (MT) to bridge the language gap between the query and document language (Zhou et al., 2012). Consequently, related works study QPP for CLIR through the lens of predicting the query translation (QT) quality (Kishida, 2008; Lee et al., 2010; Hefny et al., 2011). For example, Kettunen (2009) compares different MT-based CLIR systems on the CLEF 2003 dataset (Braschler, 2004) and finds that the retrieval system with the best results also achieved the highest METEOR scores (Banerjee and Lavie, 2005). Similarly, Lignos et al. (2019) show on three CLIR tasks that the relationship between BLEU (Papineni et al., 2002) and MAP is "approximately linear". Today, QT, also known as the *translate test* approach, is still commonly adopted in CLIR (Ture and Boschee, 2014; Saleh and Pecina, 2020; Lawrie et al., 2022). For example, in Section 8.4.1 we discussed how different MT errors categories such as topic shifts, source word copying and hallucinations can propagate to CLIR and deteriorate its performance.

Standard MT evaluation metrics such as BLEU and METEOR cannot be used in QPP because reference translations are not available at retrieval time. To this end, *reference-free MT quality estimation* (QE) methods (Fonseca et al., 2019; Zouhar et al., 2023) aim to predict translation quality without access to a correct translation. As such, they are similar to post-retrieval QPP in the sense that they

compare model input (source text) and output (translation) to predict a model's performance. Human-targeted Translation Edit Rate (HTER) (Snover et al., 2006) is a widely used measure for evaluating reference-free MT QE methods. It involves human post-editors and measures the number of edits required to obtain a corrected translation. Reference-free MT QE can be framed as a supervised task. For example, Uni+ (Yankovskaya et al., 2019) uses a regression model to directly predict HTER scores from multilingual representations extracted from mBERT (Devlin et al., 2019) and LASER (Artetxe and Schwenk, 2019a). MT-Ranker (Moosa et al., 2024) frames reference-free MT QE as a pair-wise ranking problem. Here, the model concatenates a given source sentence with two translation candidates and predicts a binary score indicating the superior translation. Unsupervised reference-free MT QE approaches use for example MoverScore (Zhao et al., 2020b), BERTScore (Zhang et al., 2020; Zhou et al., 2020; Song et al., 2021), the cosine similarity between multilingual embeddings (Fonseca et al., 2019), or large language models (Chen et al., 2023) to compare translations against source sentences. Importantly, reference-free MT QE can only be employed for selecting MT-based CLIR systems and cannot be used to estimate the performance of, e.g., multilingual bi-encoders.

**Mixture-of-Expert Models.** Our instance-level model selection framework is conceptually related to Mixture-of-Expert (MoE) models (Jacobs et al., 1991; Shazeer et al., 2017; Chen et al., 2019; Fedus et al., 2022; Zuo et al., 2022, *inter alia*), which use gating mechanisms to route instances to internal sub-layers. Our work is most similar to (Chen et al., 2019), who use adversarial training to learn a mixture of language experts (i.e., encoder layers) and a shared language-agnostic feature extractor. Their model routes tokens of test instances to those language experts that are closest to the test language. In another recent work, Cai et al. (2023) adopt MoE for information retrieval. Their model first encodes its input with shared layers. The encoded sequence is forwarded to three types of expert models: a lexical expert generates sparse bag-of-word representations (Formal et al., 2021) which are used for lexical retrieval, a local expert computes relevance scores similar to ColBERT (Khattab and Zaharia, 2020) and a global expert is implemented as a bi-encoder model (Karpukhin et al., 2020). At retrieval time, each expert produces a ranking which are then fused into a single ranking. Contrary to MoE models, we do not train a single model in an end-to-end fashion, but instead maintain independent experts. This allows for integrating closed-source models (Bubeck et al., 2023; OpenAI, 2023) or external embedding APIs (Kamalloo et al., 2023).

**Cross-lingual Transfer of Dependency Parsers.** Parsing languages with no training data has been a very active topic of research for nearly a decade since the pivotal works by (McDonald et al., 2011) and (Petrov et al., 2012). Many diverse approaches are explored along the lines of model transfer, annotation projection, machine translation (Täckström et al., 2013; Guo et al., 2015; Zhang and Barzi-

lay, 2015; Tiedemann and Agić, 2016; Rasooli and Collins, 2017), and selective sharing based on language typology (Naseem et al., 2012) and structural similarity (Ponti et al., 2018; Meng et al., 2019). However, the vast majority of prior work involves bulk evaluation, whereby transfer parsers are validated by mean accuracy on test data. Such evaluation protocols stand in contrast with the fact that languages exhibit high variance in syntactic structure, which calls for a sensitive treatment of *every sentence*. While an oracle single-source parser may be appropriate for the majority of sentences in a given dataset, instance-based treatment closes the gap to the best achievable result given an array of pre-trained parsers, as we also show in Section 9.3. Prior work relied on existing manually curated resources such as the URIEL database (Littell et al., 2017), using the KL-Divergence on POS-trigrams (Rosa and Žabokrtský, 2015), or handcrafted features derived from the datasets at hand. Our work is most similar to the recent work of (Lin et al., 2019): they learn to score and rank languages in order to predict the top transfer languages. However, contrary to their work, our approach does not employ a model to learn the ranking but transforms the labels to directly reflect the ranking when we train the scoring model. In addition, we stress the importance of instance-based learning for cross-lingual parser transfer in particular. Another core difference is that our approach does not rely on external resources. Instead, it relies on trainable parser embeddings that encode the necessary features in a single representation.

## 9.3 Motivating Instance-Level Parser Selection

The idea behind instance-level parser selection is intuitive: given a set of parsers for resource-rich source languages, it is unlikely that the same source-language parser is the best choice for all instances (i.e., UPOS-sentences) of the target language. Therefore, we first investigate the performance of an oracle model that would be able to predict the best source-language parser for each individual POS-sentence from the target-language treebank. To verify this, we rely on the well-known bi-affine parser (Dozat and Manning, 2017; Dozat et al., 2017) and train it on delexicalized UD2.3 treebanks (Nivre et al., 2018) of 42 languages.[4] We then parse the delexicalized treebanks of the 20 low-resource languages with all 42 source parsers, and measure their performance per each instance in each target treebank. We compare the performance of two *oracle* parser selection strategies: (1) single-best parser selection (SBPS), in which for each target test treebank we select the parser that performs best on the entire treebank; and (2) instance-level parser selection strategy (ILPS), where for each UPOS-sentence from each test treebank, we select the parser that produces the best parse for that UPOS-sentence.

The differences in Unlabeled Attachment Scores (UAS) between the two transfer paradigms are shown in Figure 9.2. This clearly demonstrates a large gap in

[4]We selected 42 languages with largest treebanks as the training languages. For languages with multiple treebanks (e.g., EN, CS), we finally chose the treebank for which the parser yielded the best monolingual parsing accuracy.

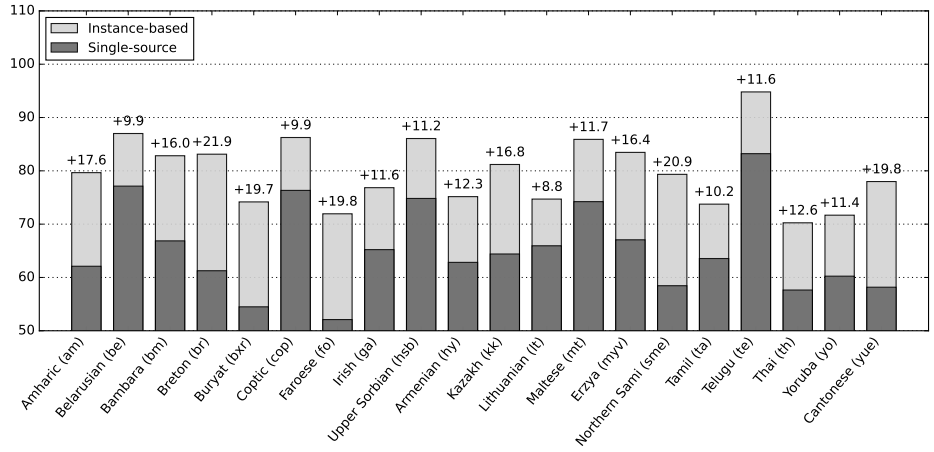


**Figure 9.2:** Comparison of UAS between *oracle* single-best (i.e., treebank-level) parser selection (SBPS) and instance-level parser selection (ILPS) strategies for cross-lingual transfer of delexicalized parsers for 20 low-resource languages from UD2.3, used as test languages throughout the paper. Individual bars annotated with potential gains when moving from SBPS to ILPS.

favor of ILPS: the average gain with ILPS is 14.5 UAS points, and it is prominent for all languages. It suggests that large improvements may be obtained with a model that can predict the best parser at the instance level, that is, for each UPOS-sentence separately. However, these are still oracle scores and we pose the following research questions in this paper: *(Q1) Is it possible to learn an instance-level prediction model to select the best parser given any UPOS-sentence, irrespective to its "language of origin"?*[5] In addition, even with noisy automatic instance-level predictions, one could still, by eliminating the noise through aggregation, use them to inform treebank-level source parser selection. In other words, another research question we pose is: *(Q2) Can we improve single-best global parser selection through aggregating instance-level parser predictions?*

## 9.4 Instance-Level Parser Selection

We now describe a novel ILPS framework based on a supervised regression model that predicts the parser accuracy for any UPOS-sentence. As such, it can be applied on UPOS-sequences of low-resource languages. In Section 9.4.1, we first train a (biaffine) parser on delexicalized treebanks for each of the 42 resource-

[5]Note that in theory the *oracle* gaps in favor of ILPS may be out of reach for automatic ILPS models, due to a potential parsing ambiguity introduced through delexicalization – i.e., the same UPOS-sentence (corresponding to different lexicalized sentences) may appear in the same treebank or across different treebanks with different gold parses. However, we have verified that this phenomenon is rare: ambiguous parses are present only for 1.4% UPOS-sentences in the concatenation of treebanks from 42 languages.

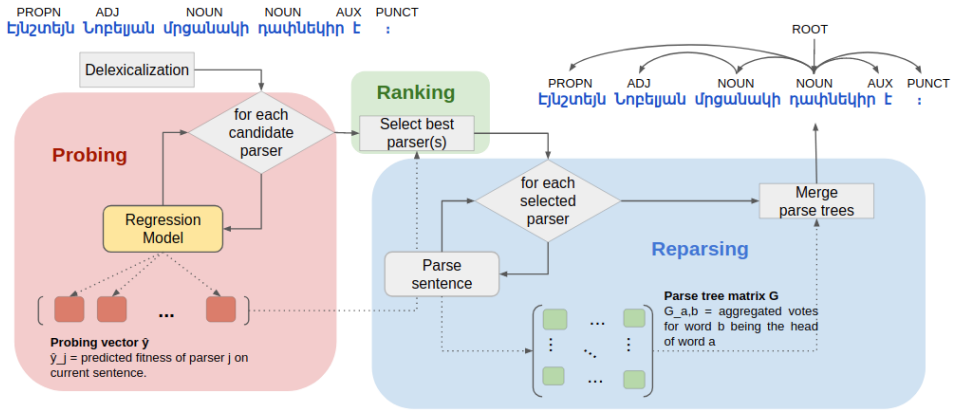


**Figure 9.3:** Illustration of the ILPS framework (at inference time) with three steps using an example sentence in Armenian (HY): (1) Probing – the ILPS regression model predicts the parsing accuracy on a given test UPOS-sentence for each of the 42 parsers; (2) Ranking – rank the parsers w.r.t. parsing accuracy for the instance and selects one or few best-performing parsers; (3) Reparsing – induce the final tree for the UPOS-sentence by merging trees produced by parsers selected in the previous step (only if more than one parser gets selected in step (2)).

rich languages from UD2.3,[6] yielding 42 expert models. We then parse with each parser the 41 treebanks of the other languages. This way we obtain the labels for training the ILPS regression model. The data preparation step is further detailed in Section 9.4.2, while the regression model itself is described in Section 9.4.3. At inference time, the ILPS model predicts the accuracy of each of the 42 parsers for each UPOS-sentence from delexicalized treebanks of the 20 test languages. Note that this constitutes a minimal-resource and true zero-shot language transfer setup: our ILPS regression model does not rely on any information about the test languages nor their respective treebanks. Finally, in Sections 9.4.4 and 9.4.5 we outline different strategies for merging the parse trees based on the predictions of the ILPS regression model. The full ILPS framework is illustrated in Figure 9.3.

### 9.4.1 Biaffine Dependency Parsers

We now briefly describe the biaffine dependency parser model proposed by (Dozat and Manning, 2017; Dozat et al., 2017). In delexicalized dependency parsing we input layer we model with sequences of POS tag embeddings $x_i \in X$. A bi-directional LSTM model (BiLSTM) (Graves et al., 2013) contextualizes the entire sequence individual token representations $R = \text{BiLSTM}(X)$. Biaffine parsers generate dependency graphs in two steps. They first predict whether a dependency relation exists for a pair of tokens, i.e. if an edge connects a pair of tokens. In the

[6] All language codes used throughout this paper are taken directly from the UD2.3 documentation (Nivre et al., 2018).

next step, they assign (i.e., predict) labels to edges to describe the type of grammatical relation. To score the likelihood of the $i^{\text{th}}$ token (dep) being a dependent of the $j^{\text{th}}$ token (head), the parser first transforms input token representations $r_i$ and $r_j$ into a head and dependent representation with feed forward networks:

$$h_i^{(\text{edge-dep})} = \text{FNN}^{(\text{edge-dep})}(r_i) \tag{9.1}$$

$$h_j^{(\text{edge-head})} = \text{FNN}^{(\text{edge-head})}(r_j) \tag{9.2}$$

These are then used as inputs in a biaffine scoring function

$$s_{i,j}^{(\text{edge})} = \text{Biaff}(h_i, h_j) = h_i^\top \mathbf{U} h_j + W(h_i \oplus h_j) + \mathbf{b}, \tag{9.3}$$

which summarizes pairwise feature interactions into an edge score $s_{i,j}^{(\text{edge})} \in \mathbb{R}$. Here, the $\oplus$ symbol denotes concatenation, $\mathbf{U}$ and $W$ are linear projection matrices and $\mathbf{b}$ is a bias term. Finally, edge scores are transformed into binary edge predictions based on their sign $\hat{y}_{i,j}^{(\text{edge})} = \{s_{i,j} \geqslant 0\}$. The computation for edge labelling follows a similar approach, two feed forward networks $\text{FFN}^{(\text{label-head})}$ and $\text{FFN}^{(\text{label-dep})}$ project contextualized representation $r_i$ and $r_j$ into the feature vector $h^{(\text{label-head})}$ and $h^{(\text{label-dep})}$. A biaffine scoring function $\text{Biaff}^{(\text{label})}$ then produces for each connected node pair $k$ label scores. The predicted label is the label with the largest score $\hat{y}^{(\text{label})} = \arg\max_k s_{i,j}^{(\text{label})}$.

### 9.4.2 Preparing ILPS Training Data

We first delexicalize all treebanks before training the parsers. After training a parser for each of the $|L|$ training languages, we measure how each of them performs on treebanks of the other $|L| - 1$ languages. Let $\text{PARSER}_i$ denote the parser of the $i$-th training language and let $\text{SENT}_j = \{\text{POS}\}_{n=1}^{N}$ be an UPOS-sentence of length $N$ from the treebank of the $j$-th language. Next, we must quantify how successful $\text{PARSER}_i$ is on some UPOS-sentence $\text{SENT}_j$. To this end, we use the number of correct dependency heads predicted by $\text{PARSER}_i$ on $\text{SENT}_j$. Using a raw number of correct heads as training labels for the ILPS regression model comes with one disadvantage: such a label would only indicate the suitability of the parser in isolation and not in comparison with other parsers. Therefore, we normalize the number of correct heads for each parser (for any given UPOS-sentence) with the average of the number of correctly predicted heads across all parsers. That is, the label $y_{i,j}$ for $\text{PARSER}_i$ and $\text{SENT}_j$ is computed as follows:

$$y_{i,j} = \frac{\#\text{correct-heads}_{i,j}}{1/|L| \cdot \sum_{l=1}^{|L|} \#\text{correct-heads}_{l,j}} \tag{9.4}$$

The normalization step ensures the comparability across sentences irrespective to their absolute length in tokens. Further, the treebanks of training languages greatly vary in size. To account for the imbalanced treebank sizes, we up-sample all below-average treebanks and down-sample all above-average treebanks.

### 9.4.3 ILPS Regression Model

Our instance-level parser selection model is a regression model based on a Transformer architecture encoder (Vaswani et al., 2017) for UPOS-sentences. The encoding of the input UPOS-sentence is forwarded, together with the embedding vector representing the parser language, to a multi-layer perceptron. It predicts the score representing the prediction of the normalized number of correct heads that the parser is expected to yield.

**Parser and POS-tag embeddings.** We learn $|L|$ parser embeddings, $\{\mathbf{p}_i\}_{i=1}^{|L|}$, one for each language (PARSER$_i$) and $K$ embedding vectors $\{\mathbf{t}_k\}_{k=1}^{K}$, one for each UPOS-tag (Petrov et al., 2012). We initialize both parser and POS-tag embeddings randomly. POS-tag embeddings are then updated during the pre-training of the POS-sentence encoder.

**UPOS-sentence encoder.** We encode UPOS-sentences with the Transformer encoder (Vaswani et al., 2017). Let the UPOS-sentence SENT$_j$ $= \{t_1^j, t_2^j, \ldots, t_T^j\}$ be a sequence of $T$ UPOS-tags. We encode each token $t_j^i$ ($i \in \{1, \ldots, K\}$, $j \in \{0, 1, \ldots, T\}$) with a vector $\mathbf{t}_j^i$ which is the concatenation of the UPOS-tag embedding and a positional embedding for the position $j$.[7] Let $Transform$ denote the encoder stack of the Transformer model with $N_T$ layers, each coupling a multi-head attention net with a feed-forward net (see Section 2.3 for more details). We then apply $Transform$ to the UPOS-tag sequence and obtain contextualized UPOS-tag representations as follows:

$$\{\mathbf{tt}_j^i\}_{j=1}^{T} = Transform\left(\{\mathbf{t}_j^i\}_{j=1}^{T}\right); \tag{9.5}$$

Following Devlin et al. (2019), we pre-train the parameters of the $Transform$ encoder and the UPOS-tag embeddings via the masked language modeling objective on the concatenation of all training treebanks. As in the original work, we consider 15% of randomly selected tokens in each sentence (but no more than 20 tokens) for replacement. In 80% of the cases, we replace the UPOS-tag with the [MASK] token, in 10% of the cases we keep the original UPOS-tag, and in remaining 10% of the cases we replace it with a randomly chosen tag.

We fine-tune the pre-train ed $Transform$ encoder and the UPOS-tag embeddings on the main ILPS regression task. At this step, similar to (Devlin et al., 2019), we prepend each UPOS-sentence with a special sentence start token $t_0^j = [ss]$, with the aim of using the transformed representation of that token as the sentence encoding.[8] We take the transformed vector of the [$ss$] token, i.e., $\mathbf{tt}_0^j$ as the final fixed-size representation of the UPOS-sentence.

[7] We adopt the wavelength-based positional encoding from the original Transformer model (Vaswani et al., 2017).

[8] This eliminates the need for an additional self-attention layer for aggregating transformed token vectors into a sentence encoding. We omitted preprending the UPOS-sentences with the sentence start token in pre-training due to the lack of any sentence-level pre-training objective.

**Feed-forward regressor and loss function.** For any given training instance consisting of the tuple $(\text{PARSER}_i, \text{SENT}_j, y_{i,j})$, we concatenate the parser's embedding $\mathbf{p}_i$ and the UPOS-sentence encoding $\mathbf{tt}_0^j$, and feed it to a feed-forward regression network (i.e., a multi-layer perceptron, MLP), whose goal is to predict $y_{i,j}$:

$$\hat{y}_{i,j} = \text{MLP}([\mathbf{p}_i; \mathbf{tt}_0^j]) \tag{9.6}$$

We define the loss function to be a simple root mean square error (RMSE) over the examples in one mini-batch as follows:

$$\mathcal{L} = \sqrt{\frac{1}{N_B} \sum_{i,j} (y_{i,j} - \hat{y}_{i,j})^2} \tag{9.7}$$

where $N_B$ is the number of instances in the batch.

### 9.4.4 Ranking and Ensembling

We can directly use the vector of scores $\hat{\mathbf{y}}_j = \{\hat{y}_{i,j}\}_{i=1}^{|L|}$ to rank the $|L|$ parsers according to their (predicted) parsing accuracy for the UPOS-sentence $SENT_j$ from some test treebank.

**Pure ILPS.** This local parser ranking, based only on the predicted parser performance for the current UPOS-sentence $SENT_j$, is used to select one or few best parsers for that UPOS-sentence. If we select only a single best parser and only according to the instance-level predictions, we refer to the *pure* instance-level parser selection (ILPS) setup:

$$i_{\text{ILPS}}(j) = \arg\max_i \{\hat{y}_{i,j} \,|\, i \in \{1, 2, \ldots, |L|\}\} \tag{9.8}$$

**SBPS from ILPS predictions.** ILPS predictions can be easily aggregated to produce a treebank-level estimate of the source parsers' performance for a test language. This brings the ILPS paradigm back into the single-best parser selection (SBPS) realm, hopefully with SBPS estimates originating from our ILPS predictions being more robust than competing SBPS metrics (Rosa and Žabokrtský, 2015; Lin et al., 2019). For a treebank of an unseen test language consisting of $M$ POS-sentences, we get the global parser's performance estimates $\bar{y}_i$ simply by averaging ILPS predictions for that parser, $\hat{y}_{i,j}$, over all $M$ test POS-sentences:

$$\bar{y}_i = \frac{1}{M} \sum_{j=1}^{M} \hat{y}_{i,j} \tag{9.9}$$

The best treebank-level parser is then selected as the one with the highest aggregate score $\bar{y}_j$:

$$i_{\text{SBPS}_{\text{ILPS}}} = \arg\max_i \{\bar{y}_i \,|\, i \in \{1, 2, \ldots, |L|\}\} \tag{9.10}$$

**Ensembling.** It is often the case – both at the instance level and at the treebank level – that two or more parsers yield similar performance. In such cases, one would expect to benefit from aggregating the predictions made by those parsers. We refer to the settings in which we consider more than one parser as ensembling (**Ens**) settings. Note that ensembling is equally applicable to both the pure ILPS setup as well as to the previously outlined $\text{SBPS}_{\text{ILPS}}$ setup in which we aggregate instance-level predictions to select the best "treebank-level" parser. In both cases, we must determine a threshold $\tau \in [0, 1]$ that defines the set of "good enough" parsers, in relative terms w.r.t. the performance of the best parser. The sets of parsers whose trees are to be merged are obtained as follows:

$$\{i_{\text{ILPS}}\}_\tau(j) = \{i | \forall i : \hat{y}_{i,j} \geqslant \max(\hat{y}_{i,j}) \cdot \tau\}, \tag{9.11}$$

$$\{i_{\text{SBPS}_{\text{ILPS}}}\}_\tau = \{i | \forall i : \bar{y}_i \geqslant \max(\bar{y}_i) \cdot \tau\}. \tag{9.12}$$

where Eq (9.11) refers to the pure ILPS setting, and Eq. (9.12) refers to the $\text{SBPS}_{\text{ILPS}}$ setting.

### 9.4.5 Reparsing

After selecting multiple parsers in the ensemble settings, we need to merge their produced parse trees into a final tree. Such a step is commonly referred to as *reparsing* (Sagae and Lavie, 2006). Here we resort to a standard reparsing procedure in which we: (1) merge the trees produced by individual parsers into a weighted graph $G$ – the parser $i$ contributes to an edge with the weight $w_i = \hat{y}_{i,j}$ (for pure ILPS; for $\text{SBPS}_{\text{ILPS}}$, $w_i = \bar{y}_i$) if the parser $i$ predicted that edge, and with $w_i = 0$ otherwise; (2) induce the Maximum Spanning Tree (MST) of $G$ (Edmonds, 1967) as the final parse of the input UPOS-sentence (see again Figure 9.3).

## 9.5 Experimental Setup

**Evaluation Data.** We perform all experiments on the UD v2.3 dataset,[9] as it contains a wide array of both resource-rich languages with large treebanks – split into train, development, and test portions – and low-resource languages with small test treebanks. For our experiments, we select 42 languages with the largest treebanks as our resource-rich source languages for training, and a set of 20 typologically diverse low-resource languages for testing.[10] Following established practice (Wang and Eisner, 2018), at inference we use gold UPOS-tags of test treebanks for all models in comparison.[11] We evaluate parsing results in terms of *Unlabeled Attachment Score* (UAS), which is the percentage of correctly classified dependency heads, ignoring the type of dependency relation.

[9] https://universaldependencies.org/

[10] We provide the full list of languages with the corresponding treebank sizes in Appendix E.

[11] While this does not affect the fairness of model comparisons (since all models, including baselines, are exposed to gold UPOS-tags), it does render reported results upper bounds w.r.t. the realistic low-resource setting where one would resort to noisier, automatically induced UPOS-tags.

**ILPS Hyperparameters** are optimized via fixed-split cross-validation on our training set (see Section 9.4.2). We set the embedding size for both parser embeddings and UPOS-tag embeddings, as well as the hidden size of the feed-forward Transformer layers to 256. The Transformer encoder has $N_T = 3$ layers with 8 attention heads in each layer. We update the model in mini-batches of 16 examples, using Adam (Kingma and Ba, 2015) with the default parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$, with an initial learning rate set to $10^{-4}$. The regression MLP has 2 hidden layers with 256 units each, plus a linear projection layer that compresses the 256-dimensional vector into a single prediction score. We perform early stopping based on the loss on the development set. For all ensembles, we set the parser inclusion threshold $\tau$ to 0.9.

**Oracle scores and baselines.** In order to provide more context for the reported ILPS scores, we also report the results of two *oracle* methods described in Section 9.3: the oracle single-best parser selection (OR-SBPS), and the oracle instance-level best parser selection (OR-ILPS). We compare to three competitive baselines: (1) the standard multi-source parser (MSP) baseline which trains a single parser model on the concatenation of all training treebanks (McDonald et al., 2011);[12] and two competitive SBPS baselines, (2) KL-SBPS – treebank-level parser selection based on the Kullback-Leibler divergence between UPOS-tag trigram distributions of the source and target language treebanks (Rosa and Žabokrtský, 2015) and (3) L2V-SBPS – treebank-level parser selection based on the cosine similarity between the syntax-based vectors of the source and target language from WALS (Lin et al., 2019).

**Ensembles.** We evaluate two ensembles based on the predictions of our ILPS-based regression model, described in Section 9.4.4: (1) an instance-level ensemble in which we merge the trees of the best parsers for each sentence (ENS-ILPS) and (2) ENS-SBPS$_{\text{ILPS}}$ – an ensemble merging the trees of treebank-level best parsers, where the treebank-level estimates are aggregated from the instance-level predictions. We evaluate comparable ensembles (i.e., with the same parser inclusion performance threshold $\tau = 0.9$) for both SBPS baselines: ENS-KL-SBPS and ENS-L2V-SBPS.

## 9.6 Results and Discussion

We first show the results for single-best parser selection models. We then proceed to a more realistic *ensemble* setup in which the models are allowed to select more than just one parser.

[12] We have run two variants of the multi-source model (MSP): a) *balanced* (trained on the treebanks downsampled or upsampled to the average treebank size as done in Section 9.4.2); b) *all* (trained on the concatenation of the full treebanks without any adjustment). For brevity, we report the results only with the latter, as it produced stronger overall performance.

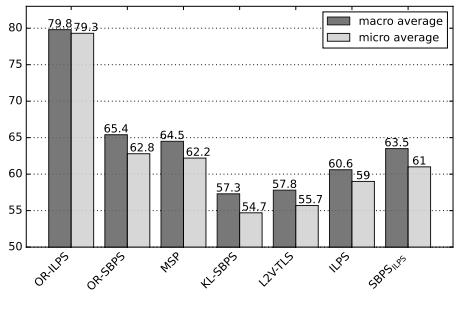


**Figure 9.4:** Performance (UAS) for single-parser selection models, micro- and macro- averaged, respectively, across 20 test languages.

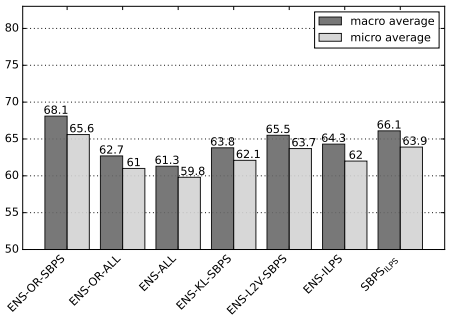


**Figure 9.5:** Performance (UAS) for ensemble models (i.e., few-parser selection), micro- and macro- averaged, respectively, across 20 test languages.

**Single-parser selection.** We report results (UAS) for all single-parser selection methods (i.e., no ensembles) along with the oracle scores on all 20 test treebanks. Table 9.1 provides performance per language, and Figure 9.4 shows the summary of the results. Our *pure* instance-based parser selection model (ILPS) significantly[13] outperforms both SBPS baselines (KL-SBPS and L2V-SBPS) averaged across all languages (see Figure 9.4). Individual instance-level predictions made by ILPS, however, do seem to be rather noisy. This is supported by the observation that $\text{SBPS}_{\text{ILPS}}$ significantly outperforms ILPS. Since $\text{SBPS}_{\text{ILPS}}$ is a simple treebank-level aggregation of ILPS sentence-level predictions, the gain can only be explained as the product of noise elimination through aggregation. ILPS outperforms KL-SBPS and L2V-SBPS on 13/20 and 14/20 test languages, respectively, whereas $\text{SBPS}_{\text{ILPS}}$ improves on 17/20 and 16/20 languages over the respective baselines. This first set of results, the preliminary comparison with well-established and competitive baselines for delexicalized parser transfer, validates the viability of the instance-based parser selection paradigm.

ILPS and $\text{SBPS}_{\text{ILPS}}$ still do not match the performance of the multi-source parser (MSP) in this simple single-parser-selection setup. We find this somewhat expected: ILPS and $\text{SBPS}_{\text{ILPS}}$ are based on parsers trained on single treebanks, whereas MSP is trained on the concatenation of all training treebanks. Therefore, we include MSP as a baseline in our ensemble evaluation as well.

**Ensemble evaluation results.** We show the results for the ensemble models in Table 9.2. A summary of results for this setup is provided in Figure 9.5. Allowing for the selection of more than a single parser in cases in which our ILPS-based predictions warrant so (i.e., when two or more parsers yield similarly good performance for some low-resource language) allows $\text{SBPS}_{\text{ILPS}}$ (i.e., its ensemble version, Ens-

[13] Significance tested with the Student's two-tailed t-test at $p = 0.01$ for sets of sentence-level UAS scores.

| | am | be | bm | br | bxr | cop | fo | ga | hsb | hy | kk |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Oracles* | | | | | | | | | | | |
| OR-ILPS | 79.7 | 87.3 | 82.8 | 83.1 | 74.2 | 86.3 | 71.9 | 76.8 | 86.1 | 75.2 | 81.2 |
| OR-SBPS | 62.1 | 77.4 | 66.9 | 61.3 | 54.5 | 76.3 | 52.1 | 65.2 | 74.8 | 62.8 | 64.4 |
| *Baselines* | | | | | | | | | | | |
| MSP | 56.7 | **78.7** | **70.7** | **62.8** | **57.8** | **77.2** | **51.4** | **66.3** | **78.2** | **67.9** | **64.9** |
| KL-SBPS | 47.5 | 77.1 | 54.2 | 56.2 | 48.9 | 65.9 | 48.7 | 65.2 | 72.9 | 62.8 | 57.5 |
| L2V-TLS | 26.9 | 70.4 | 55.6 | 58.7 | 53.8 | 69.9 | 48.9 | 61.3 | 71.1 | 57.5 | 64.4 |
| *ILPS models (ours)* | | | | | | | | | | | |
| ILPS | 57.1 | 75.3 | 60.2 | 60.5 | 53.5 | 70.9 | 49.5 | 62.2 | 72.5 | 56.4 | 58.6 |
| $\text{SBPS}_{\text{ILPS}}$ | **62.1** | 77.4 | 62.7 | 60.5 | 54.5 | 75.3 | 48.6 | 61.4 | 72.9 | 62.8 | 64.1 |
| | lt | mt | myv | sme | ta | te | th | yo | yue | **Ma** | **Mi** |
| *Oracles* | | | | | | | | | | | |
| OR-ILPS | 74.7 | 85.9 | 83.5 | 79.3 | 73.8 | 94.8 | 70.3 | 71.7 | 78.0 | 79.8 | 79.3 |
| OR-SBPS | 65.9 | 74.2 | 67.1 | 58.5 | 63.6 | 83.2 | 57.6 | 60.2 | 58.2 | 65.4 | 62.8 |
| *Baselines* | | | | | | | | | | | |
| MSP | **63.3** | **77.4** | 65.3 | **61.4** | 59.5 | 75.5 | 56.3 | **59.0** | 37.6 | **64.5** | **62.2** |
| KL-SBPS | 46.8 | 69.0 | 54.3 | 57.2 | 47.3 | 74.8 | 35.2 | 47.9 | **56.7** | 57.3 | 54.7 |
| L2V-SBPS | 49.9 | 70.4 | **67.1** | 57.2 | 54.5 | **83.2** | 47.6 | 45.2 | 41.7 | 57.8 | 55.7 |
| *ILPS models (ours)* | | | | | | | | | | | |
| ILPS | 56.2 | 71.7 | 61.2 | 55.1 | 58.0 | 71.4 | 52.9 | 54.9 | 53.5 | 60.6 | 59.0 |
| $\text{SBPS}_{\text{ILPS}}$ | 58.6 | 73.2 | 67.1 | 57.2 | **63.6** | 77.0 | **57.6** | 57.2 | 56.2 | 63.5 | 61.0 |

**Table 9.1:** Results for single-parser selection models. Results for 42 parsers (an exception is the MSP model which trains a single parser on the concatenation of all training treebanks) on 20 low-resource test languages. Ma & Mi: average performance across 20 languages, macro- and micro-averaged scores, respectively. The best result in each column, not considering oracle scores, is in bold.

$\text{SBPS}_{\text{ILPS}}$) to significantly outperform the strong MSP baseline. The two SBPS baseline methods in their ensemble variants (Ens-KL-SBPS and Ens-L2V-SBPS) reduce the gap in comparison with the previous single-parser selection setup (see Table 9.1 again). However, our treebank-level parser selection model based on instance-level predictions (Ens-$\text{SBPS}_{\text{ILPS}}$) still significantly outperforms the ensembles of the other two SBPS methods.

Encouragingly, both Ens-ILPS and Ens-$\text{SBPS}_{\text{ILPS}}$ outperform the *oracle* Ens-Or-All, which merges parses produced by all training parsers, using their gold performance on the test treebanks for weighting the individual parser contributions. Furthermore, Ens-$\text{SBPS}_{\text{ILPS}}$ also improves over the oracle single-parser selection OR-SBPS reported in Table 9.1. In summary, we believe these results provide sufficient evidence for the viability of the ILPS transfer paradigm and warrant further research efforts in this direction.

| | am | be | bm | br | bxr | cop | fo | ga | hsb | hy | kk |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Oracle ensembles* | | | | | | | | | | | |
| ENS-OR-SBPS | 62.9 | 79.2 | 70.3 | 64.2 | 62.1 | 78.9 | 50.5 | 68.6 | 78.5 | 66.3 | 69.2 |
| ENS-OR-ALL | 59.6 | 78.0 | 70.8 | 63.5 | 49.8 | 78.4 | 50.6 | 67.6 | 76.2 | 58.6 | 52.6 |
| *Baseline ensembles* | | | | | | | | | | | |
| MSP | 56.7 | 78.7 | 70.7 | 62.8 | 57.8 | 77.2 | **51.4** | 66.3 | **78.2** | **67.9** | 64.9 |
| ENS-ALL | 59.2 | 77.1 | 70.5 | 63.0 | 45.6 | 78.2 | 50.3 | 67.2 | 75.4 | 57.4 | 47.5 |
| ENS-KL-SBPS | **60.7** | **79.2** | **71.2** | 63.5 | 56.6 | 78.1 | 50.6 | 67.7 | 76.0 | 57.2 | 58.6 |
| ENS-L2V-SBPS | 60.4 | 78.7 | 68.9 | **63.8** | **61.5** | 77.5 | 50.7 | 68.1 | 76.7 | 59.1 | **70.7** |
| *ILPS model-based ensembles (ours)* | | | | | | | | | | | |
| ENS-ILPS | 59.6 | 78.7 | 68.2 | 62.8 | 56.1 | 77.9 | 50.8 | 67.2 | 76.5 | 60.8 | 61.7 |
| ENS-SBPS$_{\text{ILPS}}$ | 60.0 | 78.7 | 70.8 | **63.8** | 61.0 | **78.4** | 50.5 | **68.2** | 77.5 | 58.9 | 68.1 |
| | lt | mt | myv | sme | ta | te | th | yo | yue | **Ma** | **Mi** |
| *Oracle ensembles* | | | | | | | | | | | |
| ENS-OR-SBPS | 65.9 | 78.3 | 66.8 | 64.3 | 65.3 | 83.9 | 61.8 | 62.8 | 61.7 | 68.1 | 65.6 |
| ENS-OR-ALL | 56.2 | 76.6 | 64.5 | 52.3 | 48.0 | 67.4 | 59.8 | 62.2 | 61.3 | 62.7 | 61.0 |
| *Baseline ensembles* | | | | | | | | | | | |
| MSP | **63.3** | **77.4** | 65.3 | **61.4** | 59.5 | 75.5 | 56.3 | 59.0 | 37.6 | 64.5 | 62.2 |
| ENS-ALL | 56.0 | 76.2 | 64.1 | 52.0 | 38.4 | 66.2 | 58.3 | 61.6 | **61.6** | 61.3 | 59.8 |
| ENS-KL-SBPS | 53.5 | 76.5 | 65.3 | 53.2 | 58.8 | 68.4 | 57.4 | **62.2** | 61.1 | 63.8 | 62.1 |
| ENS-L2V-SBPS | 54.5 | 77.0 | 65.2 | 50.9 | **66.5** | 74.3 | 60.2 | 61.6 | 62.8 | 65.5 | 63.7 |
| *ILPS model-based ensembles (ours)* | | | | | | | | | | | |
| ENS-ILPS | 60.6 | 76.5 | 63.4 | 56.5 | 61 | 72.8 | 57.4 | 60.0 | 57.4 | 64.3 | 62.0 |
| ENS-SBPS$_{\text{ILPS}}$ | 62.9 | 76.7 | **66.8** | 53.6 | 65.3 | **78.5** | **60.5** | 62.0 | 60.1 | **66.1** | **63.9** |

**Table 9.2:** Results for ensemble-based parser selection models. Additional models: ENS-OR-ALL – merges parses by all 42 parsers, but uses oracle performance as parser weights; ENS-ALL – ensembles all 42 parsers, with equal weights. An exception is the MSP model which is not an ensemble model, but rather trains a single parser on the concatenation of all training treebanks. Ma & Mi: average performance across 20 languages, macro- and micro-averaged scores, respectively. The best result in each column, not considering oracle scores, is in bold.

## 9.7 Conclusion

This chapter is motivated by the observation that multilingual models are effectively limited in the number of languages they can encode (curse of multilinguality). To overcome this limitation, we explore zero-shot transfer with multiple encoders. We present a proof of concept on the task of delexicalized dependency parsing, which allows us to experiment on a large number of languages. On this task, we demonstrate that there is a large disparity between mean test-set and per-instance accuracy in cross-lingual parser transfer setups. We showed convincing

evidence that one source parser is not the optimal choice for all target-language sentences. Motivated by the analysis, we proposed a novel approach to close this gap in this proof-of-concept study: instance-based parser selection. Our framework provides competitive results, where in the ensemble setting, we outperform all baselines, and markedly even the single-source oracle parser selection, while using a simple thresholding heuristic to select the parsers.

We see the proposed framework as the first exploratory step in the direction of robust instance-level model transfer, which opens several avenues for future research. While this proof-of-concept work assumed the existence of gold POS tags, we will also experiment with the same approach "in the wild", with learned or transferred POS taggers, and we will also extend the study to lexicalized parser transfer following the latest developments in the domain of lexicalized multilingual and cross-lingual parsing (Üstün et al., 2020; Glavaš and Vulić, 2021). In Section 10.2, we outline possible directions for future work and discuss how the instance-level model selection framework can be applied in information retrieval tasks where queries are formulated in different (unseen) languages.

# Chapter 10

# Conclusion

In this thesis, we study resource-lean cross-lingual transfer methods for cross-lingual information retrieval (CLIR) as introduced in Section 1.1.2. We experimented on a total of forty-seven language pairs including thirty-three cross-lingual and fourteen monolingual retrieval setups. In this chapter, we first synthesize our key insights and findings (Section 10.1) and then conclude with a discussion of possible directions for future work (Section 10.2).

## 10.1 Summary of Findings

In this section, we synthesize our contributions into six key findings (**A–F**). We first summarize *Part II Resource-Learn Transfer of Bi-Encoders* (Chapters 4 and 5) into three takeaways related to the role of context, contextualization and multilingual representation spaces (**A–C**). We then summarize our insights from *Part III Resource-Lean Transfer of Cross-Encoders* (Chapters 6 to 8) and discuss two main findings related to zero-shot cross-lingual transfer of rerankers (**D–E**). Finally, we discuss our insights from comparing the similarity between (i) the query and document language and (ii) the training and test language (**F**).

**A. Cross-lingual word embeddings are resource-lean and effective.** In Chapter 4, we proposed two CLIR models based on cross-lingual word embeddings (CLWE). The first model, Term-by-Term Query Translation (TbT-QT) casts CLIR into a noisy variant of monolingual IR (MoIR) by replacing query words with their cross-lingual nearest neighbors, followed by lexical retrieval. The second model, Bag-of-Words Aggregation (BoW-Agg) follows a non-parametric bi-encoder approach (see Section 4.1) and represents queries and documents as the (weighted) sum of their constituent CLWEs. Both models are resource-lean since they do not rely on any CLIR task supervision. All models outperform a purely lexical retrieval baseline (query likelihood model, see Section 3.3). We also find that they perform much worse than the resource-intensive CLIR approach based on machine translation (MT). While CLWEs seem to fall behind MT-based models and multilingual

sentence encoders (Chapter 5), it is important to emphasize their minimal cross-lingual supervision requirements (bilingual dictionaries). While mBERT and NMT systems only cover 1% of the world's over 7,000 languages (Wang et al., 2022a), existing lexical resources such as the PanLex database (Kamholz et al., 2014) cover thousands of languages and have been used in prior CLIR research (Vulić et al., 2019; Jiang et al., 2020b). In fact, we showed that even fully unsupervised approaches, which induce CLWEs without any bilingual supervision perform competitively to their supervised counterparts. As such, CLWEs are a resource-lean alternative for scaling up CLIR to many languages. Furthermore, CLWEs obtained from post-hoc alignment of monolingual embedding spaces do not suffer from the "curse of multilinguality" (Conneau et al., 2020) because language-specific model parameters remain distinct from another.

**B. Weakly aligned contextual representations do not outperform cross-lingual word embeddings.** Weak alignment in multilingual pre-trained language models (mPLM) refers to the phenomenon where text embeddings from different languages form languages clusters (Cao et al., 2020; Roy et al., 2020; Wang et al., 2022c), rather than exhibiting a strong cross-lingual semantic alignment. In line with prior work, we find that this has a detrimental effect when we use off-the-shelf mPLMs as bi-encoders in CLIR, where representations of queries and documents need to be matched across different languages. Our document-level CLIR experiments show that neither static nor dynamic representations extracted from mPLMs manage to outperform CLWEs, even after controlling for the maximum sequence length limitation of mPLMs. This demonstrates that multilingual masked language modelling alone does not suffice to yield high quality representation spaces for CLIR. We further find that specializing mPLMs for sentence-level similarity tasks, i.e. multilingual sentence encoders can outperform CLWE-based approaches by a large margin. However, to train such encoders we typically need access to parallel data and labeled data in related tasks. Since this data could also be used to train MT systems in the first place, one could argue that multilingual sentence encoders are resource-intensive. In our experiments we used off-the-shelf models. In future work, we plan to control for this aspect and conduct a fair side-by-side comparison under fixed resource constraints.

**C. Excessive context and insufficient context degrade the effectiveness of multilingual text encoders in unsupervised CLIR.** In Chapter 5, we investigate multilingual text encoders including mPLMs and encoders specialized for sentence-similarity (sentence encoders). Here, we focus on the importance of context for unsupervised document-level CLIR. Due to their maximum sequence length constraint multilingual encoders can only process a limited number of tokens. When relevant information appears outside this context window it cannot be encoded, which, consequently, degrades performance in unsupervised CLIR (insufficient context). In our experiments we also observe the opposite effect, where increasing

the input sequence length can harm the retrieval effectiveness due to "too much" context (excessive context). This is expected for unsupervised CLIR where superfluous information may obfuscate relevance signals contained in documents. We show that representing documents by multiple embeddings corresponding to different text segments (localized relevance matching) is an effective approach for unsupervised CLIR with long documents. Lastly, we find that filtering stopwords deprives sentence encoders of important syntactic context and causes their CLIR performance to decrease (cf. $\text{DISTIL}_{\text{FILTER}}$ in Section 5.5.1).

**D. Monolingual overfitting negatively impacts zero-shot transfer for CLIR and can be regularized with code switching.** While there exist large-scale training data for English-to-English retrieval (EN–EN) we often lack equivalent training resources for cross-lingual retrieval settings (X–Y). Throughout most part of this thesis we used with the CLEF 2003 dataset a CLIR test collection, which does not have sufficient data to train neural retrieval models from scratch (Section 3.3). We investigated to what extent monolingual information retrieval (MoIR) training data can be used to obtain effective CLIR models. We specifically evaluated the zero-shot transfer performance of two cross-encoder models trained on different EN–EN retrieval datasets. Our results show larger gains when these models are transferred into a monolingual setting (X–X) and smaller gains when they are transferred into a cross-lingual setting (X–Y). This gap can largely be attributed to what refer to as "monolingual overfitting" (Section 6.4.2). That is, supervised reranking models trained on MoIR data are biased towards exact token matches, which cannot be exploited at test time when the query and document language are different from another, as it is the case in CLIR. We show that monolingual overfitting can be regularized by code-switching query and document tokens into different languages (Chapter 7). This approach reduces the importance of lexical matches during training and requires minimal bilingual supervision. We find that it consistently improves results in CLIR and maintains stable results in MoIR.

**E. Decomposing CLIR into language acquisition and learning-to-rank is a resource-lean and effective way to improve zero-shot rerankers.** Modular deep learning (Pfeiffer et al., 2023) allows us to independently specialize models towards different tasks ("skills") while also mitigating task interference. We show that training CLIR models can be decomposed into learning the retrieval task (relevance matching) and learning the semantics of the query and document language (language acquisition). Relevance matching is similar to semantic matching and is characterized by the importance of exact matching signals, the importance of query keywords, and diverse matching requirements[1] (Guo et al., 2016; Rao et al., 2019b). The task of specializing rankers towards semantics of the target language pair is necessary to perform relevance matching across language boundaries. In

[1]Long documents covering a single topic can be matched globally against the query, and long documents covering multiple topics should be matched locally (Guo et al., 2016, Section 5.5.3).

Chapter 8, we study two parameter-efficient transfer methods to modularize CLIR for zero-shot reranking: Adapters (Pfeiffer et al., 2020) and Sparse Fine-tuning Masks (SFTMs) (Ansell et al., 2022). Specifically, we compose rerankers by combining independently trained language adapters (SFTMs) with ranking adapters (SFTMs). Our results on CLIR show consistent gains compared to the standard zero-shot cross-lingual transfer approach, in which we train rerankers only on EN–EN data.[2] This shows that modularizing CLIR into its constituent tasks is an effective and resource-lean way to mitigate the requirement for direct CLIR supervision in cross-lingual reranking. However, it is important to note that a later study by Yang et al. (2022b) found that adapters perform less well in CLIR with DPR (Karpukhin et al., 2020) and ColBERT (Khattab and Zaharia, 2020).

**F. The effectiveness of CLIR varies with language proximity.** A common way to measure linguistic similarity between languages is to compare their language family (see Table 3.1 in Section 3.3). On unsupervised document-level CLIR with CLWEs (Chapter 4) and multilingual text encoders (Chapter 5), we generally obtain the best retrieval performance for the language pairs EN–{IT, DE} and DE–IT, i.e. when both the query and document language are from the same family (Indo-European). Exceptions to this are language pairs involving Finnish (Uralic), for which most models yield better results than for language pairs involving Russian (Indo-European). However, like most European languages, Finnish uses the Latin script whereas Russian uses the Cyrillic script.[3] A likely explanation is that language pairs in different scripts have substantially fewer vocabulary tokens in common, if any. For example, named entities are oftentimes not translated between languages that use the same script, but transliterated between different alphabets. These findings suggest that script differences also play a crucial role in CLIR.

In our zero-shot cross-lingual transfer experiments (Chapters 6 to 8) we find that the transfer performance is also impacted by the language proximity between the training and test languages. In line with prior work by Lauscher et al. (2020), we find for both CLIR and MoIR that the zero-shot transfer of rerankers (trained on EN–EN retrieval data) is more effective when the query and document languages are (i) well represented in the pre-training corpus of the underlying pre-trained language model and (ii) typologically close to English. Our results show that zero-shot cross-lingual transfer of rerankers performs worst on low-resource languages, which do not meet those requirements (see Table 8.3 in Section 8.4.1).

[2]Note that we train models on MS MARCO (Nguyen et al., 2016) and evaluate them on the CLEF 2003 dataset (Braschler, 2004), i.e. we evaluate zero-shot cross-lingual and cross-domain transfer. Also, the effectiveness of modular rerankers depends on the right level of sparsity.

[3]We observe similar performance drops with other language pairs written in different alphabets. For example, our results on cross-lingual passage re-ranking reveal that EN–{AR, RU} performs substantially worse than EN–{DE, IT}, DE–RU performs worse than DE–{IT, NL}, and AR–X exhibits the worst performance overall (see Section 7.4).

## 10.2 Future work

**Machine translation is *not* all you need.** As discussed in Section 3.2, there are two widely used approaches of using machine translation (MT) in cross-lingual retrieval: one can either use MT to translate queries at retrieval time (*translate test*) or to translate training data typically from English into the query and document language (*translate train*). In Chapters 4 to 5, we followed the translate test approach and show that, when combined with lexical retrieval models, MT outperforms cross-lingual word embeddings, multilingual pre-trained language models, and multilingual sentence encoders. In Chapters 7 and 8, we show for zero-shot cross-lingual transfer of rerankers that also the translate-train approach outperforms CLIR models that do not rely on MT.[4] At the same time, results on the recent CLIR benchmarks XOR-QA (Asai et al., 2021b) and HC4 (Lawrie et al., 2022) reveal that current state-of-the-art MT systems still underperform in comparison to human translation, highlighting further room for improvement. These findings might suggest that MT is all you need, and that future work should focus on improving translation systems as means to bridge the language gap (Artetxe et al., 2023; Ebing and Glavaš, 2023). This raises the question, does MT replace the need for cross-lingual transfer by means of multilingual representations?

Despite strong empirical results, MT-based CLIR has its limitations. First and foremost, training MT models is a resource-intensive task requiring access to large-scale parallel data. Our analysis on low-resource languages (Section 7) reveal that recent MT models fail when this requirement is not met. We specifically find that translation errors such as hallucinations, repetitions and topic shifts can propagate to CLIR and cause a dramatic decrease in performance (see Table 8.1 in Chapter 8). For these languages, obtaining bilingual lexicons is arguably a more cost-effective alternative. In future work, we aim to compare CLWE-based methods against MT-based methods in truly low-resource settings. Another limitation of MT-based CLIR systems, as discussed in Section 3.2, is the cultural gap between training and test data. Consequently, following the translate-train approach biases ranking models towards domains and topics found in the source language but not in the target languages. It is therefore crucial to involve native speakers in the development of CLIR benchmarks (Zhang et al., 2023c) such that the data reflects cultural diversity and local interests. Lastly, translating queries to different document languages at test time is costly and increases the query latency (Moraes et al., 2021), especially in multilingual retrieval with translations into multiple languages.

**Query-level (re)ranker selection for cross-lingual and multilingual information retrieval.** In Chapter 9, we introduce a framework for (delexicalized) cross-lingual transfer of dependency parsers. Our proof-of-concept work demonstrates that our supervised routing model can outperform global model selection approaches

[4] In Chapter 7 we call this approach `Fine-tuning` due to the fact that the mMARCO dataset (Bonifacio et al., 2021) is a machine translated dataset.

based on typological language similarity (Littell et al., 2017; Lin et al., 2019). This raises the research question of whether one can train a routing model (i.e. query performance prediction model, QPP) to select for each query the best (re)ranking model from a large pool of models specialized for different languages and language families? And how does this compare to (i) training one multilingual (i.e., multi-source) ranking model on the concatenation of each expert model's training data and (ii) selecting rankers based on language similarity?

In future work, we plan to apply our instance-level expert model selection framework for CLIR, where instances are queries written in seen and unseen languages and expert models are retrieval systems specialized for different languages. As a first step, we plan to adopt this framework for the multi-stage retrieval paradigm (Nogueira et al., 2019b) and investigate to what extent a multilingual QPP model can identify the best reranker on a query-level.[5] In multi-stage retrieval, for some queries, it might be preferable to route queries to multiple rankers (ensemble) or to not rerank a first-stage result at all (Gao et al., 2021b). Query-level reranker selection is similar to cross-lingual query performance prediction (Kishida, 2008) and extends it to multiple retrieval models specialized for a single language or a few related languages. Motivated by mitigating the "curse of multilinguality" (Conneau et al., 2020) in multilingual models, we plan to compare our approach against a single multilingual model, using the MIRACL dataset (Zhang et al., 2023c).

**Parametric Information Asymmetry.** Information asymmetry, as discussed in Section 1.1.1, can be inspected in a transparent way by comparing the amount and nature of information available in different languages. This is not possible with large language models (LLM) such as GPT-4 (OpenAI, 2023) and Llama-3 (Meta, 2024), because their black-box nature obfuscates their internal knowledge (Mallen et al., 2023). For example, Zhang et al. (2023b) show that LLMs exhibit varying degrees of multilingual language understanding capabilities. However, it is unclear how well LLMs can (1) cross-lingually retrieve internal knowledge when the original data or the question is written in a low-resource language, and (2) how this knowledge interacts with externally retrieved information (Lewis et al., 2020b). That is, do LLMs, similar to weakly aligned bi-encoders (Roy et al., 2020), favor information sources written in the same language as the prompt? Also, how do LLMs respond to prompts when "multilingual parametric knowledge" stores conflicting information (Xu et al., 2024; Xie et al., 2024)? Shedding light on these questions improves the interpretability of LLMs and our understanding how LLMs "reason" under parametric information asymmetry, especially when this information is originally written in low-resource languages.

[5] In single-stage retrieval, one could adopt the method proposed in (Cai et al., 2023) for CLIR by incorporating multilingual dense retrievers (Izacard et al., 2021; Zhang et al., 2021) and MT-based lexical retrievers.

# Bibliography


Abdou, S. and Savoy, J. (2005). Report on CLIR task for the NTCIR-5 evaluation campaign. In *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-5, National Center of Sciences, Tokyo, Japan, December 6-9, 2005*.

Abdul-Jaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., Smucker, M. D., and Wade, C. (2004). Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189.

Adriani, M. and Van Rijsbergen, C. (1999). Term similarity-based query expansion for cross-language information retrieval. In *International Conference on Theory and Practice of Digital Libraries*, pages 311–322.

Agić, Ž. (2017). Cross-lingual parser selection for low-resource languages. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 1–10, Gothenburg, Sweden.

Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., and Lin, J. (2019). Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3490–3496, Hong Kong, China.

Albalak, A., Levy, S., and Wang, W. Y. (2023). Addressing issues of cross-linguality in open-retrieval question answering systems for emergent domains. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 1–10, Dubrovnik, Croatia.

Ansell, A., Ponti, E., Korhonen, A., and Vulić, I. (2022). Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland.

Aoyama, T. and Schneider, N. (2022). Probe-less probing of BERT's layer-wise linguistic knowledge with masked word prediction. In *Proceedings of the 2022*

*Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 195–201, Hybrid: Seattle, Washington + Online.

Arabzadeh, N., Khodabakhsh, M., and Bagheri, E. (2021). Bert-qpp: Contextualized pre-trained transformers for query performance prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2857–2861, New York, NY, USA.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Artetxe, M., Goswami, V., Bhosale, S., Fan, A., and Zettlemoyer, L. (2023). Revisiting machine translation for cross-lingual classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6489–6499, Singapore.

Artetxe, M., Labaka, G., and Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia.

Artetxe, M., Labaka, G., and Agirre, E. (2020). Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online.

Artetxe, M. and Schwenk, H. (2019a). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, pages 597–610.

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

Asai, A., Kasai, J., Clark, J., Lee, K., Choi, E., and Hajishirzi, H. (2021a). XOR QA: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online.

Asai, A., Kasai, J., Clark, J. H., Lee, K., Choi, E., and Hajishirzi, H. (2021b). Xor qa: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564.

Asai, A., Longpre, S., Kasai, J., Lee, C.-H., Zhang, R., Hu, J., Yamada, I., Clark, J. H., and Choi, E. (2022). MIA 2022 shared task: Evaluating cross-lingual

open-retrieval question answering for 16 diverse languages. In *Proceedings of the Workshop on Multilingual Information Access (MIA)*, pages 108–120, Seattle, USA.

Asai, A., Schick, T., Lewis, P., Chen, X., Izacard, G., Riedel, S., Hajishirzi, H., and Yih, W.-t. (2023). Task-aware retrieval with instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3650–3675, Toronto, Canada.

Askari, A., Aliannejadi, M., Kanoulas, E., and Verberne, S. (2023). A test collection of synthetic documents for training rankers: Chatgpt vs. human experts. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 5311–5315.

Azad, H. K. and Deepak, A. (2019). Query expansion techniques for information retrieval: A survey. *Information Processing & Management*, 56(5):1698–1735.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stocia, A., Tiwary, S., and Wang, T. (2016). Ms marco: A human generated machine reading comprehension dataset. In *CoCo@NIPS*.

Ballesteros, L. and Croft, B. (1996). Dictionary methods for cross-lingual information retrieval. In *International conference on database and expert systems applications*, pages 791–801.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

Bansal, R., Pruthi, D., and Belinkov, Y. (2022). Measures of information reflect memorization patterns. In *Advances in Neural Information Processing Systems*.

Barrón-Cedeno, A., Rosso, P., Lalitha Devi, S., Clough, P., and Stevenson, M. (2013). Pan@ fire: Overview of the cross-language !ndian text re-use detection competition. In *Multilingual Information Access in South Asian Languages: Second International Workshop, FIRE 2010, Gandhinagar, India, February 19-21, 2010 and Third International Workshop, FIRE 2011, Bombay, India, December 2-4, 2011, Revised Selected Papers*, pages 59–70.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: the long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bender, E. (2019). The #benderrule: On naming the languages we study and why it matters. *The Gradient*. [Online; accessed 4-May-2024].

Bender, E. M. (2011). On achieving and evaluating language-independence in nlp. *Linguistic Issues in Language Technology*, 6.

Berger, A., Caruana, R., Cohn, D., Freitag, D., and Mittal, V. (2000). Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199.

Bevilacqua, M., Ottaviano, G., Lewis, P., Yih, S., Riedel, S., and Petroni, F. (2022). Autoregressive search engines: Generating substrings as document identifiers. In *Advances in Neural Information Processing Systems*.

Blloshmi, R., Pasini, T., Campolungo, N., Banerjee, S., Navigli, R., and Pasi, G. (2021). IR like a SIR: Sense-enhanced Information Retrieval for Multiple Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1030–1041, Online and Punta Cana, Dominican Republic.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017a). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017b). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Bonab, H., Allan, J., and Sitaraman, R. (2019). Simulating clir translation resource scarcity using high-resource languages. In *Proceedings of ICTIR*, page 129–136.

Bonifacio, L., Abonizio, H., Fadaee, M., and Nogueira, R. (2022). Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392.

Bonifacio, L. H., Campiotti, I., Jeronymo, V., Lotufo, R., and Nogueira, R. (2021). mmarco: A multilingual version of the ms marco passage ranking dataset. *arXiv preprint arXiv:2108.13897*.

Braschler, M. (2001). CLEF 2000 — overview of results. In *Cross-Language Information Retrieval and Evaluation*, pages 89–101, Darmstadt, Germany.

Braschler, M. (2002). CLEF 2001 — overview of results. In *Evaluation of Cross-Language Information Retrieval Systems*, pages 9–26.

Braschler, M. (2003). CLEF 2002 — overview of results. In *Advances in Cross-Language Information Retrieval*, pages 9–27.

Braschler, M. (2004). Clef 2003 – overview of results. In *Comparative Evaluation of Multilingual Information Access Systems*, pages 44–63, Berlin, Heidelberg.

Braschler, M., Krause, J., Peters, C., and Schäuble, P. (1999). Cross-language information retrieval (clir) track overview. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7). NIST Special Publication*, number 500–242.

Braschler, M., Krause, J., Peters, C., and Schäuble, P. (2000). Cross-language information retrieval (clir) track overview. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8). NIST Special Publication*, pages 25–34.

Brauwers, G. and Frasincar, F. (2023). A general survey on attention mechanisms in deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3279–3298.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. In *Proceedings of NeurIPS*.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Cai, Y., Fan, Y., Bi, K., Guo, J., Chen, W., Zhang, R., and Cheng, X. (2023). Came: Competitively learning a mixture-of-experts model for first-stage retrieval. *arXiv preprint arXiv:2311.02834*.

Callahan, E. S. and Herring, S. C. (2011). Cultural bias in wikipedia content on famous persons. *Journal of the American society for information science and technology*, 62(10):1899–1915.

Cao, S., Kitaev, N., and Klein, D. (2020). Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

Cao, Y., Shi, J., Li, J., Liu, Z., and Li, C. (2017). On modeling sense relatedness in multi-prototype word embedding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 233–242, Taipei, Taiwan.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. (2023). Quantifying memorization across neural language models. In *International Conference on Learning Representations*.

Carmel, D. and Yom-Tov, E. (2010). *Estimating the query difficulty for information retrieval*. Morgan & Claypool Publishers.

Carterette, B. (2017). Statistical significance testing in information retrieval: Theory and practice. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1387–1389.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of SemEval*, pages 1–14.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium.

Chang, T. A., Arnett, C., Tu, Z., and Bergen, B. (2024). When is multilinguality a curse? language modeling for 250 high- and low-resource languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4074–4096, Miami, Florida, USA.

Chapelle, O., Joachims, T., Radlinski, F., and Yue, Y. (2012). Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)*, 30(1):1–41.

Chen, J., Zhang, R., Guo, J., Liu, Y., Fan, Y., and Cheng, X. (2022a). Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, CIKM '22, page 191–200, New York, NY, USA.

Chen, K., Chen, H., Kando, N., Kuriyama, K., Lee, S., Myaeng, S., Kishida, K., Eguchi, K., and Kim, H. (2002). Overview of CLIR task at the third NTCIR workshop. In *Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering, NTCIR-3, Tokyo, Japan, October 8-10, 2002*.

Chen, T., Zhang, M., Lu, J., Bendersky, M., and Najork, M. (2022b). Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models. In *Advances in Information Retrieval*, pages 95–110, Cham.

Chen, X., Awadallah, A. H., Hassan, H., Wang, W., and Cardie, C. (2019). Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy.

Chen, X., He, B., Hui, K., Sun, L., and Sun, Y. (2021). Simplified tinybert: Knowledge distillation for document retrieval. In *ECIR (2)*, volume 12657 of *Lecture Notes in Computer Science*, pages 241–248.

Chen, Y. and Ritter, A. (2021). Model selection for cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5675–5687, Online and Punta Cana, Dominican Republic.

Chen, Y., Wang, R., Jiang, H., Shi, S., and Xu, R. (2023). Exploring the use of large language models for reference-free text quality evaluation: An empirical study. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 361–374, Nusa Dua, Bali.

Chen, Z., Eavani, H., Chen, W., Liu, Y., and Wang, W. Y. (2020). Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online.

Chidambaram, M., Yang, Y., Cer, D., Yuan, S., Sung, Y., Strope, B., and Kurzweil, R. (2019). Learning cross-lingual sentence representations via a multi-task dual-encoder model. In *Proceedings of the ACL Workshop on Representation Learning for NLP*, pages 250–259.

Choi, J., Jung, E., Suh, J., and Rhee, W. (2021). Improving bi-encoder document ranking models with two rankers and multi-teacher distillation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2192–2196, New York, NY, USA.

Chung, H. W., Garrette, D., Tan, K. C., and Riesa, J. (2020). Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online.

Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., and Palomaki, J. (2020). TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Conneau, A. and Kiela, D. (2018). SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32.

Contractor, D., Kothari, G., Faruquie, T. A., Subramaniam, L. V., and Negi, S. (2010). Handling noisy queries in cross language faq retrieval. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 87–96.

Contractor, D., Venkata Subramaniam, L., P, D., and Mittal, A. (2013). Text retrieval using sms queries: Datasets and overview of fire 2011 track on sms-based faq retrieval. In *Multilingual Information Access in South Asian Languages: Second International Workshop, FIRE 2010, Gandhinagar, India, February 19-21, 2010 and Third International Workshop, FIRE 2011, Bombay, India, December 2-4, 2011, Revised Selected Papers*, pages 86–99.

Cook, T. D., Campbell, D. T., and Shadish, W. (2002). *Experimental and quasi-experimental designs for generalized causal inference*, volume 1195.

Cooper, W. S. (1971). A definition of relevance for information retrieval. *Information storage and retrieval*, 7(1):19–37.

Cormack, G. V., Clarke, C. L. A., and Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA.

Craswell, N. (2009). *Mean Reciprocal Rank*, pages 1703–1703. Boston, MA.

Craswell, N., Mitra, B., Yilmaz, E., and Campos, D. (2021a). Overview of the TREC 2020 deep learning track. In *Text REtrieval Conference (TREC)*.

Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Lin, J. (2021b). Ms marco: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1566–1576.

Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Lin, J. (2022). Overview of the trec 2021 deep learning track. In *Text REtrieval Conference (TREC)*. NIST.

Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Lin, J., Voorhees, E. M., and Soboroff, I. (2023). Overview of the trec 2022 deep learning track. In *Text REtrieval Conference (TREC)*. NIST.

Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Voorhees, E. M. (2020a). Overview of the trec 2019 deep learning track. In *Text REtrieval Conference (TREC)*.

Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Voorhees, E. M. (2020b). Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

Dai, Z. and Callan, J. (2019). Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 985–988.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy.

Dai, Z., Zhao, V. Y., Ma, J., Luan, Y., Ni, J., Lu, J., Bakalov, A., Guu, K., Hall, K., and Chang, M.-W. (2023). Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*.

Davis, M. W. and Dunning, T. E. (1995). A trec evaluation of query translation methods for multi-lingual text retrieval. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4). NIST Special Publication*, volume 483.

Davison, J., Feldman, J., and Rush, A. (2019). Commonsense knowledge mining from pretrained models. 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 1173–1178, Hong Kong, China.

De Bruyn, M., Lotfi, E., Buhmann, J., and Daelemans, W. (2021). MFAQ: a multilingual FAQ dataset. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 1–13.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Dementieva, D. and Panchenko, A. (2021). Cross-lingual evidence improves monolingual fake news detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 310–320, Online.

Deshpande, A., Talukdar, P., and Narasimhan, K. (2022). When is BERT multilingual? isolating crucial ingredients for cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3610–3623, Seattle, United States.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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.

Ding, B., Hu, J., Bing, L., Aljunied, M., Joty, S., Si, L., and Miao, C. (2022). GlobalWoZ: Globalizing MultiWoZ to develop multilingual task-oriented dialogue systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1639–1657, Dublin, Ireland.

Ding, Y., Jiang, B., Yu, A., Zheng, A., and Liang, J. (2024). Which model to transfer? a survey on transferability estimation. *arXiv preprint arXiv:2402.15231*.

Dinu, G., Lazaridou, A., and Baroni, M. (2014). Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*.

Doddington, G., Liggett, W., Martin, A., Przybocki, M., and Reynolds, D. (1998). Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation. Technical report, National Inst of Standards and Technology Gaithersburg Md.

Dozat, T. and Manning, C. D. (2017). Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Dozat, T., Qi, P., and Manning, C. D. (2017). Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.

Drozd, A., Gladkova, A., and Matsuoka, S. (2016). Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530, Osaka, Japan.

Dryer, M. S. and Haspelmath, M. (2013). *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Dufter, P. and Schütze, H. (2020). Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online.

Dziri, N., Milton, S., Yu, M., Zaiane, O., and Reddy, S. (2022). On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States.

Eberhard, D. M., Simons, G. F., and Fennig, C. D. e. (2022). *Ethnologue: Languages of the World*.

Ebing, B. and Glavaš, G. (2023). To translate or not to translate: A systematic investigation of translation-based cross-lingual transfer to low-resource languages. *arXiv preprint arXiv:2311.09404*.

Edmonds, J. (1967). Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Einolghozati, A., Arora, A., Sainz-Maza Lecanda, L., Kumar, A., and Gupta, S. (2021). El volumen louder por favor: Code-switching in task-oriented semantic parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1009–1021, Online.

Ethayarajh, K. (2019). How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China.

Faggioli, G., Dietz, L., Clarke, C. L. A., Demartini, G., Hagen, M., Hauff, C., Kando, N., Kanoulas, E., Potthast, M., Stein, B., and Wachsmuth, H. (2023a). Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '23, page 39–50, New York, NY, USA.

Faggioli, G., Ferro, N., Mothe, J., Raiber, F., and Fröbe, M. (2023b). Report on the 1st workshop on query performance prediction and its evaluation in new tasks (qpp++ 2023) at ecir 2023. In *ACM SIGIR Forum*, volume 57, pages 1–7.

Faggioli, G., Formal, T., Marchesin, S., Clinchant, S., Ferro, N., and Piwowarski, B. (2023c). Query performance prediction for neural ir: Are we there yet? In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part I*, page 232–248, Berlin, Heidelberg.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Auli, M., and Joulin, A. (2021a). Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Fan, Y., Guo, J., Ma, X., Zhang, R., Lan, Y., and Cheng, X. (2021b). A linguistic study on relevance modeling in information retrieval. In *Proceedings of the Web Conference 2021*, pages 1053–1064.

Faruqui, M. and Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden.

Fedus, W., Zoph, B., and Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270.

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. (2022). Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland.

Ferro, N. and Peters, C. (2019). *From Multilingual to Multimodal: The Evolution of CLEF over Two Decades*, pages 3–44. Cham.

Fetahu, B., Fang, A., Rokhlenko, O., and Malmasi, S. (2021). Gazetteer enhanced named entity recognition for code-mixed web queries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1677–1681.

Fonseca, E., Yankovskaya, L., Martins, A. F. T., Fishel, M., and Federmann, C. (2019). Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy.

Formal, T., Piwowarski, B., and Clinchant, S. (2021). Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Fu, Z., Lam, W., So, A. M.-C., and Shi, B. (2021). A theoretical analysis of the repetition problem in text generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12848–12856.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Galke, L., Saleh, A., and Scherp, A. (2017). Word embeddings for practical information retrieval. In *Informatik 2017*, pages 2155–2167.

Galuščáková, P., Oard, D. W., and Nair, S. (2021). Cross-language information retrieval. *arXiv preprint arXiv:2111.05988*.

Ganguly, D., Bandyopadhyay, A., Mitra, M., and Jones, G. J. F. (2016). Retrievability of code mixed microblogs. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 973–976.

Gao, L., Dai, Z., and Callan, J. (2020). Modularized transfomer-based ranking framework. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4180–4190, Online.

Gao, L., Dai, Z., and Callan, J. (2021a). COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042, Online.

Gao, L., Dai, Z., and Callan, J. (2021b). Rethink training of bert rerankers in multi-stage retrieval pipeline. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, pages 280–286.

Gao, L., Dai, Z., Chen, T., Fan, Z., Van Durme, B., and Callan, J. (2021c). Complement lexical retrieval model with semantic residual embeddings. In *Advances in Information Retrieval*, pages 146–160, Cham.

Gautam, D., Kodali, P., Gupta, K., Goel, A., Shrivastava, M., and Kumaraguru, P. (2021). CoMeT: Towards code-mixed translation using parallel monolingual sentences. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 47–55, Online.

Geigle, G., Pfeiffer, J., Reimers, N., Vulić, I., and Gurevych, I. (2022). Retrieve fast, rerank smart: Cooperative and joint approaches for improved cross-modal retrieval. *Transactions of the Association for Computational Linguistics*, 10:503–521.

Glavaš, G., Franco-Salvador, M., Ponzetto, S. P., and Rosso, P. (2018). A resource-light method for cross-lingual semantic textual similarity. *Knowledge-based systems*, 143:1–9.

Glavaš, G., Karan, M., and Vulić, I. (2020). Xhate-999: Analyzing and detecting abusive language across domains and languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365.

Glavaš, G., Litschko, R., Ruder, S., and Vulić, I. (2019). How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721, Florence, Italy.

Glavaš, G. and Vulić, I. (2021). Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Gospodinov, M., MacAvaney, S., and Macdonald, C. (2023). Doc2query–: When less is more. In *European Conference on Information Retrieval*, pages 414–422.

Graham, M. and Zook, M. (2013). Augmented realities and uneven geographies: exploring the geolinguistic contours of the web. *Environment and Planning A*, 45(1):77–99.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649.

Guerreiro, N. M., Voita, E., and Martins, A. (2023). Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia.

Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., and Tao, D. (2024). A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Guo, J., Che, W., Yarowsky, D., Wang, H., and Liu, T. (2015). Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1234–1244, Beijing, China.

Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016). 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.

Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wu, C., Croft, W. B., and Cheng, X. (2020). A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.

Guo, M., Shen, Q., Yang, Y., Ge, H., Cer, D., Hernandez Abrego, G., Stevens, K., Constant, N., Sung, Y.-H., Strope, B., and Kurzweil, R. (2018). Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of WMT*, pages 165–176.

Guo, Z., Schlichtkrull, M., and Vlachos, A. (2022). A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Gupta, A. and Srikumar, V. (2021). X-fact: A new benchmark dataset for multilingual fact checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 675–682, Online.

Gupta, P., Clough, P., Rosso, P., Stevenson, M., and Banchs, R. E. (2013). Pan@fire: Overview of the cross-language !ndian news story search (cl!nss) track. In *Proceedings of the 4th and 5th Annual Meetings of the Forum for Information Retrieval Evaluation*, FIRE '12 & '13, New York, NY, USA.

Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Hangya, V., Saadi, H. S., and Fraser, A. (2022). Improving low-resource languages in pre-trained multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11993–12006, Abu Dhabi, United Arab Emirates.

Harman, D. (1992). Evaluation issues in information retrieval. *Information Processing & Management*, 28(4):439–40.

Harris, Z. S. (1954). Distributional structure. *WORD*, 10(2-3):146–162.

He, J., Zhang, Z., Berg-Kirkpatrick, T., and Neubig, G. (2019). Cross-lingual syntactic transfer through unsupervised adaptation of invertible projections. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3211–3223, Florence, Italy.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hefny, A., Darwish, K., and Alkahky, A. (2011). Is a query worth translating: ask the users! In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*, pages 238–250.

Hercig, T. and Kral, P. (2021). Evaluation datasets for cross-lingual semantic textual similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 524–529, Held Online.

Hermann, K. M. and Blunsom, P. (2014). Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota.

Heyman, G., Vulić, I., and Moens, M.-F. (2017a). Bilingual lexicon induction by learning to combine word-level and character-level representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1085–1095, Valencia, Spain.

Heyman, G., Vulić, I., and Moens, M.-F. (2017b). Bilingual lexicon induction by learning to combine word-level and character-level representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1085–1095, Valencia, Spain.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hofmann, K., Li, L., Radlinski, F., et al. (2016). Online evaluation for information retrieval. *Foundations and Trends® in Information Retrieval*, 10(1):1–117.

Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., and Hanbury, A. (2020a). Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666*.

Hofstätter, S., Khattab, O., Althammer, S., Sertkan, M., and Hanbury, A. (2022). Introducing neural bag of whole-words with colberter: Contextualized late interactions using enhanced reduction. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, CIKM '22, page 737–747, New York, NY, USA.

Hofstätter, S., Zamani, H., Mitra, B., Craswell, N., and Hanbury, A. (2020b). Local self-attention over long text for efficient document retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2021–2024.

Hoshen, Y. and Wolf, L. (2018). Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 469–478, Brussels, Belgium.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. (2020). XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceeding of ICML*, pages 4411–4421.

Huang, K.-H., Ahmad, W., Peng, N., and Chang, K.-W. (2021a). Improving zero-shot cross-lingual transfer learning via robust training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1684–1697, Online and Punta Cana, Dominican Republic.

Huang, K.-H., Zhai, C., and Ji, H. (2022). CONCRETE: Improving cross-lingual fact-checking with cross-lingual retrieval. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1024–1035, Gyeongju, Republic of Korea.

Huang, Z., Bonab, H., Sarwar, S. M., Rahimi, R., and Allan, J. (2021b). Mixed attention transformer for leveraging word-level knowledge to neural cross-lingual information retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 760–770.

Huang, Z., Yu, P., and Allan, J. (2023). Improving cross-lingual information retrieval on low-resource languages via optimal transport distillation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1048–1056.

Hui, K., Yates, A., Berberich, K., and de Melo, G. (2017). 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, Copenhagen, Denmark.

Hui, K., Yates, A., Berberich, K., and De Melo, G. (2018). Co-pacrr: A context-aware neural ir model for ad-hoc retrieval. In *Proceedings of WSDM*, pages 279–287.

Imani, A., Lin, P., Kargaran, A. H., Severini, S., Jalili Sabet, M., Kassner, N., Ma, C., Schmid, H., Martins, A., Yvon, F., and Schütze, H. (2023). Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada.

Irvine, A. and Callison-Burch, C. (2017). A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. (2021). Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Izacard, G. and Grave, E. (2021). Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy.

Jiang, Z., El-Jaroudi, A., Hartmann, W., Karakos, D., and Zhao, L. (2020a). Cross-lingual information retrieval with BERT. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 26–31, Marseille, France.

Jiang, Z., El-Jaroudi, A., Hartmann, W., Karakos, D., and Zhao, L. (2020b). Cross-lingual information retrieval with BERT. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 26–31, Marseille, France.

Johnson, J., Douze, M., and Jégou, H. (2017). Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Joshi, P., Santy, S., Budhiraja, A., Bali, K., and Choudhury, M. (2020a). The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online.

Joshi, P., Santy, S., Budhiraja, A., Bali, K., and Choudhury, M. (2020b). The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online.

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., and Grave, E. (2018). Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium.

K, K., Wang, Z., Mayhew, S., and Roth, D. (2020). Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Kamalloo, E., Zhang, X., Ogundepo, O., Thakur, N., Alfonso-hermelo, D., Rezagholizadeh, M., and Lin, J. (2023). Evaluating embedding APIs for information retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 518–526, Toronto, Canada.

Kamholz, D., Pool, J., and Colowick, S. M. (2014). PanLex: Building a resource for panlingual lexical translation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 3145–3150.

Kando, N. (2000). Ntcir workshop: Japanese-and chinese-english cross-lingual information retrieval and multi-grade relevance judgments. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 24–35.

Kando, N. (2007). Overview of the sixth NTCIR workshop. In *Proceedings of the 6th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-6, National Center of Sciences, Tokyo, Japan, May 15-18, 2007*.

Kando, N., Kuriyama, K., Nozue, T., Eguchi, K., Kato, H., and Hidaka, S. (1999). Overview of IR tasks. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, NTCIR-1, Tokyo, Japan, August 30 - September 1, 1999*.

Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. (2023). Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707.

Kantor, P. and Voorhees, E. (2000). The trec-5 confusion track. *Information Retrieval*, 2(2-3):165–176.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online.

Kelly, D. and Teevan, J. (2003). Implicit feedback for inferring user preference: a bibliography. In *Acm Sigir Forum*, volume 37, pages 18–28.

Kenter, T., Borisov, A., and de Rijke, M. (2016). Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 941–951, Berlin, Germany.

Kettunen, K. (2009). Choosing the best mt programs for clir purposes–can mt metrics be helpful? In *Advances in Information Retrieval: 31th European Conference on IR Research, ECIR 2009, Toulouse, France, April 6-9, 2009. Proceedings 31*, pages 706–712.

Khattab, O. and Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kishida, K. (2008). Prediction of performance of cross-language information retrieval using automatic evaluation of translation. *Library & Information Science Research*, 30(2):138–144.

Kishida, K., Chen, K., Lee, S., Kuriyama, K., Kando, N., Chen, H., Myaeng, S., and Eguchi, K. (2004a). Overview of CLIR task at the fourth NTCIR workshop. In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization, NTCIR-4, National Center of Sciences, Tokyo, Japan, June 2-4, 2004*.

Kishida, K., Kuriyama, K., Kando, N., and Eguchi, K. (2004b). Prediction of performance on cross-lingual information retrieval by regression models. In *NTCIR*.

Klementiev, A., Titov, I., and Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Kohavi, R., Longbotham, R., Sommerfield, D., and Henne, R. M. (2009). Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery*, 18:140–181.

Kolbitsch, J. and Maurer, H. A. (2006). The transformation of the web: How emerging communities shape the information we consume. *J. Univers. Comput. Sci.*, 12(2):187–213.

Kong, W., Khadanga, S., Li, C., Gupta, S. K., Zhang, M., Xu, W., and Bendersky, M. (2022). Multi-aspect dense retrieval. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3178–3186, New York, NY, USA.

Kornai, A. (2013). Digital language death. *PloS one*, 8(10):e77056.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Kudugunta, S., Caswell, I. R., Zhang, B., Garcia, X., Xin, D., Kusupati, A., Stella, R., Bapna, A., and Firat, O. (2023). MADLAD-400: A multilingual and document-level large audited dataset. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Kumar, A., Makhija, P., and Gupta, A. (2020). Noisy text data: Achilles' heel of BERT. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21, Online.

Kurland, O. and Culpepper, J. S. (2018). Fusion in information retrieval: Sigir 2018 half-day tutorial. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1383–1386.

Lample, G., Conneau, A., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word translation without parallel data. In *International Conference on Learning Representations*.

Lan, W., Chen, Y., Xu, W., and Ritter, A. (2020). An empirical study of pre-trained transformers for Arabic information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4727–4734, Online.

Lauscher, A., Ravishankar, V., Vulić, I., and Glavaš, G. (2020). From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online.

Lavrenko, V., Choquette, M., and Croft, W. B. (2002). Cross-lingual relevance models. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 175–182.

Lawrie, D., MacAvaney, S., Mayfield, J., McNamee, P., Oard, D. W., Soldaini, L., and Yang, E. (2023). Overview of the trec 2022 neuclir track. In *Proceedings of The Thirty-First Text REtrieval Conference (TREC). NIST Special Publication*.

Lawrie, D., MacAvaney, S., Mayfield, J., McNamee, P., Oard, D. W., Soldaini, L., and Yang, E. (2024). Overview of the trec 2023 neuclir track. In *Proceedings of The Thirty-Second Text REtrieval Conference (TREC). NIST Special Publication*.

Lawrie, D., Mayfield, J., Oard, D. W., and Yang, E. (2022). Hc4: A new suite of test collections for ad hoc clir. In *European Conference on Information Retrieval*, pages 351–366.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

LeCun, Y. and Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Lee, C.-J., Chen, C.-H., Kao, S.-H., and Cheng, P.-J. (2010). To translate or not to translate? In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 651–658.

Lee, J., Lee, D., Kim, J., and Hwang, S.-w. (2023). C2lir: Continual cross-lingual transfer for low-resource information retrieval. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II*, pages 466–474.

Lee, J.-T., Kim, S.-B., Song, Y.-I., and Rim, H. C. (2008). Bridging lexical gaps between queries and questions on large online q&a collections with compact translation models. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 410–418.

Lee, K., Chang, M.-W., and Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy.

Levow, G.-A., Oard, D. W., and Resnik, P. (2005). Dictionary-based techniques for cross-language information retrieval. *Information processing & management*, 41(3):523–547.

Lewis, P., Oguz, B., Rinott, R., Riedel, S., and Schwenk, H. (2020a). MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020b). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Li, B. and Cheng, P. (2018). Learning neural representation for CLIR with adversarial framework. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1861–1870, Brussels, Belgium.

Li, C., Yates, A., MacAvaney, S., He, B., and Sun, Y. (2023a). Parade: Passage representation aggregation fordocument reranking. *ACM Transactions on Information Systems (TOIS)*, 42(2).

Li, M., Lin, S.-C., Oguz, B., Ghoshal, A., Lin, J., Mehdad, Y., Yih, W.-t., and Chen, X. (2023b). CITADEL: Conditional token interaction via dynamic lexical routing for efficient and effective multi-vector retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11891–11907, Toronto, Canada.

Li, W. Y., Weeds, J., and Weir, D. (2022). MuSeCLIR: A multiple senses and cross-lingual information retrieval dataset. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1128–1135, Gyeongju, Republic of Korea.

Liang, D., Gonen, H., Mao, Y., Hou, R., Goyal, N., Ghazvininejad, M., Zettlemoyer, L., and Khabsa, M. (2023). XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore.

Liang, Y., Duan, N., Gong, Y., Wu, N., Guo, F., Qi, W., Gong, M., Shou, L., Jiang, D., Cao, G., Fan, X., Zhang, R., Agrawal, R., Cui, E., Wei, S., Bharti, T., Qiao, Y., Chen, J.-H., Wu, W., Liu, S., Yang, F., Campos, D., Majumder, R., and Zhou, M. (2020). XGLUE: A new benchmark dataset for cross-lingual pretraining, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online.

Lignos, C., Cohen, D., Lien, Y.-C., Mehta, P., Croft, W. B., and Miller, S. (2019). 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.

Lim, S. H., Yun, T., Kim, J., Choi, J., and Kim, T. (2024). Analysis of multi-source language training in cross-lingual transfer. *arXiv preprint arXiv:2402.13562*.

Limkonchotiwat, P., Ponwitayarat, W., Udomcharoenchaikit, C., Chuangsuwanich, E., and Nutanong, S. (2022). CL-ReLKT: Cross-lingual language knowledge transfer for multilingual retrieval question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2141–2155, Seattle, United States.

Lin, B. Y., Lee, S., Khanna, R., and Ren, X. (2020). Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6862–6868, Online.

Lin, J., Ma, X., Lin, S.-C., Yang, J.-H., Pradeep, R., and Nogueira, R. (2021a). Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

Lin, J., Nogueira, R., and Yates, A. (2021b). Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.

Lin, J., Nogueira, R., and Yates, A. (2022). *Pretrained Transformers for Text Ranking*. Springer Nature.

Lin, P., Martins, A. F., and Schütze, H. (2024). Xampler: Learning to retrieve cross-lingual in-context examples. *arXiv preprint arXiv:2405.05116*.

Lin, S.-C., Yang, J.-H., and Lin, J. (2021c). In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 163–173, Online.

Lin, Y.-H., Chen, C.-Y., Lee, J., Li, Z., Zhang, Y., Xia, M., Rijhwani, S., He, J., Zhang, Z., Ma, X., Anastasopoulos, A., Littell, P., and Neubig, G. (2019). Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy.

Litschko, R., Müller-Eberstein, M., van der Goot, R., Weber-Genzel, L., and Plank, B. (2023). Establishing trustworthiness: Rethinking tasks and model evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Singapore.

Littell, P., Mortensen, D. R., Lin, K., Kairis, K., Turner, C., and Levin, L. (2017). URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain.

Liu, F., Vulić, I., Korhonen, A., and Collier, N. (2021). Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. *arXiv preprint arXiv:2104.08027*.

Liu, Q., Kusner, M. J., and Blunsom, P. (2020). A survey on contextual embeddings. *arXiv preprint arXiv:2003.07278*.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luan, Y., Eisenstein, J., Toutanova, K., and Collins, M. (2021). Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

MacAvaney, S., Cohan, A., and Goharian, N. (2020a). SLEDGE-Z: A zero-shot baseline for COVID-19 literature search. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4171–4179, Online.

MacAvaney, S., Macdonald, C., and Ounis, I. (2022). Streamlining evaluation with ir-measures. In *European Conference on Information Retrieval*, pages 305–310.

MacAvaney, S., Soldaini, L., and Goharian, N. (2020b). Teaching a new dog old tricks: Resurrecting multilingual retrieval using zero-shot learning. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 246–254.

MacAvaney, S., Yates, A., Cohan, A., and Goharian, N. (2019). CEDR: contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1101–1104.

Majumder, P., Mitra, M., Agrawal, M., and Mehta, P. (2013a). Fire '12 & '13: Proceedings of the 4th and 5th annual meetings of the forum for information retrieval evaluation. New York, NY, USA.

Majumder, P., Pal, D., Bandyopadhyay, A., and Mitra, M. (2013b). Overview of fire 2010. In *Multilingual Information Access in South Asian Languages*, pages 252–257, Berlin, Heidelberg.

Malkin, D., Limisiewicz, T., and Stanovsky, G. (2022). A balanced data approach for evaluating cross-lingual transfer: Mapping the linguistic blood bank. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4903–4915, Seattle, United States.

Mallen, A., Asai, A., Zhong, V., Das, R., Hajishirzi, H., and Khashabi, D. (2022). When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., and Hajishirzi, H. (2023). When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada.

Mass, Y., Carmeli, B., Roitman, H., and Konopnicki, D. (2020). Unsupervised FAQ retrieval with question generation and BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 807–812, Online.

Mayfield, J., Yang, E., Lawrie, D., Barham, S., Weller, O., Mason, M., Nair, S., and Miller, S. (2023). Synthetic cross-language information retrieval training data. *arXiv preprint arXiv:2305.00331*.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy.

McDonald, R., Petrov, S., and Hall, K. (2011). Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. (2022). Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.

Meng, T., Peng, N., and Chang, K.-W. (2019). Target language-aware constrained inference for cross-lingual dependency parsing. 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 1117–1128, Hong Kong, China.

Meta (2024). Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/. [Online; accessed 4-May-2024].

Metzler, D., Tay, Y., Bahri, D., and Najork, M. (2021). Rethinking search: Making domain experts out of dilettantes. *SIGIR Forum*, 55(1).

Mielke, S. J. (2016). Language diversity in ACL 2004 - 2016. https://sjmielke.com/acl-language-diversity.htm. [Online; accessed 4-May-2024].

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013c). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013d). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia.

Milajevs, D. and et al, D. K. (2014). Evaluating neural word representations in tensor-based compositional settings. In *EMNLP*, pages 708–719.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Miquel-Ribé, M. and Laniado, D. (2019). Wikipedia cultural diversity dataset: A complete cartography for 300 language editions. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 620–629.

Mirzadeh, S. I., Farajtabar, M., Pascanu, R., and Ghasemzadeh, H. (2020). Understanding the role of training regimes in continual learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 7308–7320.

Mirzakhalov, J., Babu, A., Ataman, D., Kariev, S., Tyers, F., Abduraufov, O., Hajili, M., Ivanova, S., Khaytbaev, A., Laverghetta Jr., A., Moydinboyev, B., Onal, E., Pulatova, S., Wahab, A., Firat, O., and Chellappan, S. (2021). A large-scale study of machine translation in Turkic languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5876–5890, Online and Punta Cana, Dominican Republic.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio.

Mitra, B. and Craswell, N. (2017). Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*.

Mitra, B., Diaz, F., and Craswell, N. (2017). 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.

Mitra, M. and Majumdar, P. (2008). FIRE: Forum for information retrieval evaluation. In *Proceedings of the 2nd workshop on Cross Lingual Information Access (CLIA) Addressing the Information Need of Multilingual Societies*.

Moosa, I. M., Zhang, R., and Yin, W. (2024). Mt-ranker: Reference-free machine translation evaluation by inter-system ranking. In *The Twelfth International Conference on Learning Representations*.

Moraes, G., Bonifácio, L. H., Rodrigues de Souza, L., Nogueira, R., and Lotufo, R. (2021). A cost-benefit analysis of cross-lingual transfer methods. *arXiv preprint arXiv:2105.06813*.

Müller, M., Rios, A., and Sennrich, R. (2020). Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164.

Naseem, T., Barzilay, R., and Globerson, A. (2012). Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 629–637, Jeju Island, Korea.

Naseri, S., Dalton, J., Yates, A., and Allan, J. (2021). CEQE: contextualized embeddings for query expansion. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I*, volume 12656 of *Lecture Notes in Computer Science*, pages 467–482.

Neeman, E., Aharoni, R., Honovich, O., Choshen, L., Szpektor, I., and Abend, O. (2023). DisentQA: Disentangling parametric and contextual knowledge with counterfactual question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10056–10070, Toronto, Canada.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. (2016). MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Ni, J., Qu, C., Lu, J., Dai, Z., Hernandez Abrego, G., Ma, J., Zhao, V., Luan, Y., Hall, K., Chang, M.-W., and Yang, Y. (2022). Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates.

Ni, J., Zhu, C., Chen, W., and McAuley, J. (2019). Learning to attend on essential terms: An enhanced retriever-reader model for open-domain question answering. 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 335–344, Minneapolis, Minnesota.

Nie, E., Liang, S., Schmid, H., and Schütze, H. (2023). Cross-lingual retrieval augmented prompt for low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8320–8340, Toronto, Canada.

Nie, J.-Y. (2010). Cross-language information retrieval. *Synthesis Lectures on Human Language Technologies*, 3(1):1–125.

Niu, J., Lu, W., and Penn, G. (2022). Does BERT rediscover a classical NLP pipeline? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3143–3153, Gyeongju, Republic of Korea.

Nivre, J., Abrams, M., Agić, Ž., Ahrenberg, L., Antonsen, L., Aplonova, K., Aranzabe, M. J., et al. (2018). Universal Dependencies 2.3.

Nogueira, R. and Cho, K. (2019). Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Nogueira, R., Jiang, Z., Pradeep, R., and Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online.

Nogueira, R., Lin, J., and Epistemic, A. (2019a). From doc2query to doctttttquery. *Online preprint*, 6.

Nogueira, R., Yang, W., Cho, K., and Lin, J. (2019b). Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424*.

Nogueira, R., Yang, W., Lin, J., and Cho, K. (2019c). Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

Ogundepo, O., Gwadabe, T., Rivera, C., Clark, J., Ruder, S., Adelani, D., Dossou, B., Diop, A., Sikasote, C., Hacheme, G., et al. (2023). Afriqa: Cross-lingual open-retrieval question answering for African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14957–14972, Singapore.

Ogundepo, O., Zhang, X., Sun, S., Duh, K., and Lin, J. (2022). Africlirmatrix: Enabling cross-lingual information retrieval for african languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8721–8728.

OpenAI (2022). Introducing chatgpt. https://openai.com/index/chatgpt/. [Online; accessed 4-May-2024].

OpenAI (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Palchowdhury, S., Majumder, P., Pal, D., Bandyopadhyay, A., and Mitra, M. (2013). Overview of fire 2011. In *Multilingual Information Access in South Asian Languages*, pages 1–12, Berlin, Heidelberg.

Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., and Cheng, X. (2016). Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Peng, H., Wang, X., Hu, S., Jin, H., Hou, L., Li, J., Liu, Z., and Liu, Q. (2022). COPEN: Probing conceptual knowledge in pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5015–5035, Abu Dhabi, United Arab Emirates.

Peskov, D., Hangya, V., Boyd-Graber, J., and Fraser, A. (2021). Adapting entities across languages and cultures. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3725–3750, Punta Cana, Dominican Republic.

Peters, C., Braschler, M., and Clough, P. (2012). *Cross-Language Information Retrieval*, pages 57–84. Springer, Berlin, Heidelberg.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana.

Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey.

Pfeiffer, J., Goyal, N., Lin, X., Li, X., Cross, J., Riedel, S., and Artetxe, M. (2022). Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States.

Pfeiffer, J., Ruder, S., Vulić, I., and Ponti, E. (2023). Modular deep learning. *Transactions on Machine Learning Research*.

Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. (2020). MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings EMNLP*, pages 7654–7673.

Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. (2021). UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic.

Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy.

Piroi, F., Lupu, M., Hanbury, A., and Zenz, V. (2011). Clef-ip 2011: Retrieval in the intellectual property domain. In *CLEF (notebook papers/labs/workshop)*.

Plank, B. and van Noord, G. (2011). Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1566–1576, Portland, Oregon, USA.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 275–281, New York, NY, USA.

Ponti, E. M., Glavaš, G., Majewska, O., Liu, Q., Vulić, I., and Korhonen, A. (2020). XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online.

Ponti, E. M., Reichart, R., Korhonen, A., and Vulić, I. (2018). Isomorphic transfer of syntactic structures in cross-lingual NLP. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1531–1542, Melbourne, Australia.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Potthast, M., Barrón-Cedeno, A., Stein, B., and Rosso, P. (2011). Cross-language plagiarism detection. *Language Resources and Evaluation*, 45:45–62.

Qian, Y., Lee, J., Duddu, S. M. K., Dai, Z., Brahma, S., Naim, I., Lei, T., and Zhao, V. Y. (2022). Multi-vector retrieval as sparse alignment. *arXiv preprint arXiv:2211.01267*.

Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W. X., Dong, D., Wu, H., and Wang, H. (2021). RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online.

Radlinski, F. and Craswell, N. (2013). Optimized interleaving for online retrieval evaluation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 245–254.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., and Shoham, Y. (2023). In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Rao, J., Liu, L., Tay, Y., Yang, W., Shi, P., and Lin, J. (2019a). Bridging the gap between relevance matching and semantic matching for short text similarity modeling. 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 5370–5381.

Rao, J., Liu, L., Tay, Y., Yang, W., Shi, P., and Lin, J. (2019b). Bridging the gap between relevance matching and semantic matching for short text similarity modeling. 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 5370–5381, Hong Kong, China.

Rasooli, M. S. and Collins, M. (2017). Cross-lingual syntactic transfer with limited resources. *Transactions of the Association for Computational Linguistics*, pages 279–293.

Raunak, V., Menezes, A., and Junczys-Dowmunt, M. (2021). The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online.

Rebuffi, S.-A., Bilen, H., and Vedaldi, A. (2017). Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, volume 30.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China.

Reimers, N. and Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online.

Reisinger, J. and Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117, Los Angeles, California.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126.

Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceeding of SIGIR*, pages 232–241.

Rocchio Jr, J. J. (1971). Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020a). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020b). A primer in BERTology: What we know about how BERT works. *Transactions of the ACL*.

Roitman, H. (2018). Enhanced performance prediction of fusion-based retrieval. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '18, page 195–198, New York, NY, USA.

Rosa, R. and Žabokrtský, Z. (2015). KLcpos3 - a language similarity measure for delexicalized parser transfer. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 243–249, Beijing, China.

Roy, D., Bhatia, S., and Jain, P. (2022). Information asymmetry in wikipedia across different languages: A statistical analysis. *Journal of the Association for Information Science and Technology*, 73(3):347–361.

Roy, D., Ganguly, D., Bhatia, S., Bedathur, S., and Mitra, M. (2018). Using word embeddings for information retrieval: How collection and term normalization choices affect performance. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1835–1838.

Roy, D., Paul, D., Mitra, M., and Garain, U. (2016). Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*.

Roy, U., Constant, N., Al-Rfou, R., Barua, A., Phillips, A., and Yang, Y. (2020). LAReQA: Language-agnostic answer retrieval from a multilingual pool. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5919–5930, Online.

Rücklé, A., Geigle, G., Glockner, M., Beck, T., Pfeiffer, J., Reimers, N., and Gurevych, I. (2021). AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic.

Ruder, S. (2020). Why You Should Do NLP Beyond English. http://ruder.io/nlp-beyond-english. [Online; accessed 4-May-2024].

Ruder, S., Constant, N., Botha, J., Siddhant, A., Firat, O., Fu, J., Liu, P., Hu, J., Garrette, D., Neubig, G., and Johnson, M. (2021). XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic.

Ruder, S., Vulić, I., and Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing*, pages 318–362. Cambridge.

Rust, P., Pfeiffer, J., Vulić, I., Ruder, S., and Gurevych, I. (2021). How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online.

Sagae, K. and Lavie, A. (2006). Parser combination by reparsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 129–132, New York City, USA.

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of linguistics*, 20:33–53.

Saleh, S. and Pecina, P. (2020). 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.

Samir, F., Park, C. Y., Field, A., Shwartz, V., and Tsvetkov, Y. (2024). Locating information gaps and narrative inconsistencies across languages: A case study of LGBT people portrayals on Wikipedia. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6747–6762, Miami, Florida, USA.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., and Zaharia, M. (2022). ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States.

Sasaki, S., Sun, S., Schamoni, S., Duh, K., and Inui, K. (2018). 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, New Orleans, Louisiana.

Schamoni, S., Hieber, F., Sokolov, A., and Riezler, S. (2014). 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, Baltimore, Maryland.

Schäuble, P. and Sheridan, P. (1998). Cross-language information retrieval (clir) track overview. In *Proceedings of the Sixth Text REtrieval Conference (TREC-6). NIST Special Publication*, number 500-240, pages 31–43.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Semnani, S., Yao, V., Zhang, H., and Lam, M. (2023). WikiChat: Stopping the hallucination of large language model chatbots by few-shot grounding on Wikipedia. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2387–2413, Singapore.

Senel, L. K., Ebing, B., Baghirova, K., Schuetze, H., and Glavaš, G. (2024). Kardeş-NLU: Transfer to low-resource languages with the help of a high-resource cousin – a benchmark and evaluation for Turkic languages. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1672–1688, St. Julian's, Malta.

Sennrich, R., Haddow, B., and Birch, A. (2016). 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.

Shalaby, W. and Zadrozny, W. (2019). Patent retrieval: a literature review. *Knowledge and Information Systems*, 61(2):631–660.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Shen, J., Chen, P.-J., Le, M., He, J., Gu, J., Ott, M., Auli, M., and Ranzato, M. (2021). The source-target domain mismatch problem in machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1519–1533, Online.

Shi, P., Bai, H., and Lin, J. (2020). Cross-lingual training of neural models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2768–2773, Online.

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

Shi, P., Zhang, R., Bai, H., and Lin, J. (2021). Cross-lingual training of dense retrievers for document retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 251–253, Punta Cana, Dominican Republic.

Shi, P., Zhang, R., Bai, H., and Lin, J. (2022). XRICL: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5248–5259, Abu Dhabi, United Arab Emirates.

Siddhant, A., Bapna, A., Firat, O., Cao, Y., Chen, M. X., Caswell, I., and Garcia, X. (2022). Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*.

Singh, J., McCann, B., Keskar, N. S., Xiong, C., and Socher, R. (2019). XLDA: Cross-lingual Data Augmentation for Natural Language Inference and Question Answering. *arXiv preprint arXiv:1905.11471*.

Smith, S. L., Turban, D. H., Hamblin, S., and Hammerla, N. Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *International Conference on Learning Representations*.

Snæbjarnarson, V., Simonsen, A., Glavaš, G., and Vulić, I. (2023). Transfer to a low-resource language via close relatives: The case study on Faroese. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 728–737, Tórshavn, Faroe Islands.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA.

Soboroff, I. (2021). Overview of trec 2021. In *Proceedings of The Thirtieth Text REtrieval Conference (TREC). NIST Special Publication*, number 500-335.

Søgaard, A. (2022). Should we ban English NLP for a year? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5260, Abu Dhabi, United Arab Emirates.

Song, Y., Zhao, J., and Specia, L. (2021). SentSim: Crosslingual semantic evaluation of machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3143–3156, Online.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Spark-Jones, K. (1975). Report on the need for and provision of an 'ideal' information retrieval test collection. *Computer Laboratory*.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

Sun, L., Luisier, F., Batmanghelich, K., Florencio, D., and Zhang, C. (2023). From characters to words: Hierarchical pre-trained language model for open-vocabulary language understanding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3605–3620, Toronto, Canada.

Sun, S. and Duh, K. (2020). Clirmatrix: A massively large collection of bilingual and multilingual datasets for cross-lingual information retrieval. In *Proceedings*

*of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4160–4170.

Täckström, O., McDonald, R., and Nivre, J. (2013). Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia.

Tan, S. and Joty, S. (2021). Code-mixing on sesame street: Dawn of the adversarial polyglots. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3596–3616, Online.

Tarunesh, I., Kumar, S., and Jyothi, P. (2021). From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3154–3169, Online.

Tay, Y., Tran, V. Q., Dehghani, M., Ni, J., Bahri, D., Mehta, H., Qin, Z., Hui, K., Zhao, Z., Gupta, J., Schuster, T., Cohen, W. W., and Metzler, D. (2022). Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems*.

Tenney, I., Das, D., and Pavlick, E. (2019a). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S., Das, D., and Pavlick, E. (2019b). What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Thakur, N., Ni, J., Ábrego, G. H., Wieting, J., Lin, J., and Cer, D. (2023). Leveraging llms for synthesizing training data across many languages in multilingual dense retrieval. *arXiv preprint arXiv:2311.05800*.

Thakur, N., Reimers, N., Daxenberger, J., and Gurevych, I. (2020). Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240*.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Tian, F., Dai, H., Bian, J., Gao, B., Zhang, R., Chen, E., and Liu, T.-Y. (2014). A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 151–160, Dublin, Ireland.

Tiedemann, J. and Agić, v. (2016). Synthetic treebanking for cross-lingual dependency parsing. *Journal of Artificial Intelligence Research*, 55:209–248.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *URL https://arxiv. org/abs/2307.09288*.

Tran, H.-N., Aizawa, A., and Takasu, A. (2024). An encoding–searching separation perspective on bi-encoder neural search. *arXiv preprint arXiv:2408.01094*.

Turc, I., Lee, K., Eisenstein, J., Chang, M.-W., and Toutanova, K. (2021). Revisiting the primacy of english in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.16171*.

Ture, F. and Boschee, E. (2014). Learning to translate: A query-specific combination approach for cross-lingual information retrieval. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 589–599, Doha, Qatar.

Üstün, A., Bisazza, A., Bouma, G., and van Noord, G. (2020). UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

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

Voorhees, E. M. (2004). Overview of the TREC 2004 robust track. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004*, volume 500-261 of *NIST Special Publication*.

Voorhees, E. M. (2019). The evolution of cranfield. *Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF*, pages 45–69.

Voorhees, E. M. and Harman, D. (1999). Overview of the eighth text retrieval conference (TREC-8). In *Proceedings of The Eighth Text REtrieval Conference,*

*TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*.

Voorhees, E. M., Harman, D. K., et al. (2005). *TREC: Experiment and evaluation in information retrieval*, volume 63. MIT Press.

Vulić, I., Glavaš, G., Reichart, R., and Korhonen, A. (2019). Do we really need fully unsupervised cross-lingual embeddings? 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 4407–4418, Hong Kong, China.

Vulić, I. and Moens, M.-F. (2015). 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.

Vulić, I. and Moens, M.-F. (2016). Bilingual distributed word representations from document-aligned comparable data. *Journal of Artificial Intelligence Research*, 55:953–994.

Vulić, I., Ponti, E. M., Litschko, R., Glavaš, G., and Korhonen, A. (2020). Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019a). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C.-C. J. (2019b). Evaluating word embedding models: Methods and experimental results. *APSIPA transactions on signal and information processing*, 8:e19.

Wang, D. and Eisner, J. (2018). Synthetic data made to order: The case of parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1325–1337, Brussels, Belgium.

Wang, H., Yu, D., Sun, K., Chen, J., and Yu, D. (2019c). Improving pre-trained multilingual model with vocabulary expansion. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 316–327, Hong Kong, China.

Wang, J. and Komlodi, A. (2018). Switching languages in online searching: A qualitative study of web users' code-switching search behaviors. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, CHIIR '18, page 201–210, New York, NY, USA.

Wang, S., Zhuang, S., and Zuccon, G. (2021a). Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 317–324, New York, NY, USA.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. (2020a). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Wang, X., Macdonald, C., Tonellotto, N., and Ounis, I. (2021b). Pseudo-relevance feedback for multiple representation dense retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 297–306.

Wang, X., Ruder, S., and Neubig, G. (2022a). Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland.

Wang, Y., Guo, Q., Yao, W., Zhang, H., Zhang, X., Wu, Z., Zhang, M., Dai, X., zhang, M., Wen, Q., Ye, W., Zhang, S., and Zhang, Y. (2024). Autosurvey: Large language models can automatically write surveys. In *Advances in Neural Information Processing Systems*.

Wang, Y., Hou, Y., Wang, H., Miao, Z., Wu, S., Sun, H., Chen, Q., Xia, Y., Chi, C., Zhao, G., Liu, Z., Xie, X., Sun, H., Deng, W., Zhang, Q., and Yang, M. (2022b). A neural corpus indexer for document retrieval. In *Advances in Neural Information Processing Systems*.

Wang, Y., Wu, A., and Neubig, G. (2022c). English contrastive learning can learn universal cross-lingual sentence embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9122–9133, Abu Dhabi, United Arab Emirates.

Wang, Z., Lipton, Z. C., and Tsvetkov, Y. (2020b). On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online.

Wiedemann, G., Remus, S., Chawla, A., and Biemann, C. (2019). Does BERT make any sense? interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 161–170, Erlangen, Germany.

Wikimedia Commons (2020). File:percentwikipediasgraph.png — wikimedia commons, the free media repository. https://commons.wikimedia.org/w/index

.php?title=File:PercentWikipediasGraph.png&oldid=452261307. [Online; accessed 4-May-2024].

Wikimedia Commons (2022). Data:wikipedia statistics/data.tab. https://commons.wikimedia.org/w/index.php?title=Data:Wikipedia_statistics/data.tab&oldid=721573759. [Online; accessed 4-May-2024].

Wikimedia Commons (2023). Data:wikipedia statistics/data.tab. https://commons.wikimedia.org/w/index.php?title=Data:Wikipedia_statistics/data.tab&oldid=837390511. [Online; accessed 4-May-2024].

Wikimedia Meta-Wiki (2019). List of wikipedias/table — meta, discussion about wikimedia projects. https://meta.wikimedia.org/w/index.php?title=List_of_Wikipedias/Table&oldid=19671823. [Online; accessed 4-May-2024].

Wikimedia Meta-Wiki (2020). List of wikipedias/table — meta, discussion about wikimedia projects. https://meta.wikimedia.org/w/index.php?title=List_of_Wikipedias/Table&oldid=20913043. [Online; accessed 4-May-2024].

Wikimedia Meta-Wiki (2021). List of wikipedias/table — meta, discussion about wikimedia projects. https://meta.wikimedia.org/w/index.php?title=List_of_Wikipedias/Table&oldid=22519389. [Online; accessed 4-May-2024].

William, C. C. (1967). The cranfield tests on index language devices. In *Aslib Proceedings*, volume 19, pages 173–194.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Womser-Hacker, C. (2001). Multilingual topic generation within the clef 2001 experiments. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 389–393.

Wu, S. and Dredze, M. (2019). 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, Hong Kong, China.

Wu, W., Jiang, C., Jiang, Y., Xie, P., and Tu, K. (2023). Do PLMs know and understand ontological knowledge? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3080–3101, Toronto, Canada.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wu, Z., Chen, Y., Kao, B., and Liu, Q. (2020). Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online.

Xie, J., Zhang, K., Chen, J., Lou, R., and Su, Y. (2024). Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Xin, J., Nogueira, R., Yu, Y., and Lin, J. (2020). Early exiting bert for efficient document ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 83–88.

Xing, C., Wang, D., Liu, C., and Lin, Y. (2015). Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado.

Xiong, C., Dai, Z., Callan, J., Liu, Z., and Power, R. (2017). 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.

Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P. N., Ahmed, J., and Overwijk, A. (2021). Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Xu, R., Qi, Z., Wang, C., Wang, H., Zhang, Y., and Xu, W. (2024). Knowledge conflicts for llms: A survey. *arXiv preprint arXiv:2403.08319*.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online.

Yang, E., Nair, S., Chandradevan, R., Iglesias-Flores, R., and Oard, D. W. (2022a). C3: Continued pretraining with contrastive weak supervision for cross language ad-hoc retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2507–2512, New York, NY, USA.

Yang, E., Nair, S., Lawrie, D., Mayfield, J., and Oard, D. W. (2022b). Parameter-efficient zero-shot transfer for cross-language dense retrieval with adapters. *arXiv preprint arXiv:2212.10448*.

Yang, J., Ma, S., Zhang, D., Wu, S., Li, Z., and Zhou, M. (2020a). Alternating language modeling for cross-lingual pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9386–9393.

Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Hernandez Abrego, G., Yuan, S., Tar, C., Sung, Y.-h., Strope, B., and Kurzweil, R. (2020b). Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online.

Yang, Y., Hernandez Abrego, G., Yuan, S., Guo, M., Shen, Q., Cer, D., Sung, Y.-h., Strope, B., and Kurzweil, R. (2019a). Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax. In *Proceedings of IJCAI*, pages 5370–5378.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019b). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32.

Yankovskaya, E., Tättar, A., and Fishel, M. (2019). Quality estimation and translation metrics via pre-trained word and sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 101–105, Florence, Italy.

Yilmaz, Z. A., Wang, S., Yang, W., Zhang, H., and Lin, J. (2019). 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.

Yong, Z.-X., Menghini, C., and Bach, S. H. (2023). Low-resource languages jailbreak gpt-4. *arXiv preprint arXiv:2310.02446*.

Young, H. (2015). The digital language divide. https://web.archive.org/web/20240106081518/http://labs.theguardian.com/digital-language-divide/. [Online; accessed 4-May-2024].

Yu, P. and Allan, J. (2020). 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.

Yu, P., Fei, H., and Li, P. (2021a). Cross-lingual language model pretraining for retrieval. In *Proceedings of the Web Conference 2021*, WWW '21, page 1029–1039, New York, NY, USA.

Yu, P., Fei, H., and Li, P. (2021b). Cross-lingual language model pretraining for retrieval. In *Proceedings of the Web Conference 2021*, pages 1029–1039.

Zachte, E. (2019). Wikipedia Statistics - Tables - Article count (official) — stats.wikimedia.org. https://stats.wikimedia.org/EN/TablesArticlesTotal.htm. [Accessed 04-05-2024].

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. (2020). Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297.

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214.

Zhang, H., Gong, Y., Shen, Y., Lv, J., Duan, N., and Chen, W. (2022a). Adversarial retriever-ranker for dense text retrieval. In *International Conference on Learning Representations*.

Zhang, R., Guo, J., Lan, Y., Xu, J., and Cheng, X. (2018). Aggregating neural word embeddings for document representation. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*, volume 10772 of *Lecture Notes in Computer Science*, pages 303–315.

Zhang, S., Liang, Y., Gong, M., Jiang, D., and Duan, N. (2022b). Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000, Dublin, Ireland.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Zhang, X., Li, M., and Lin, J. (2023a). Improving out-of-distribution generalization of neural rerankers with contextualized late interaction. *arXiv preprint arXiv:2302.06589*.

Zhang, X., Li, S., Hauer, B., Shi, N., and Kondrak, G. (2023b). Don't trust ChatGPT when your question is not in English: A study of multilingual abilities and types of LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7915–7927, Singapore.

Zhang, X., Ma, X., Shi, P., and Lin, J. (2021). Mr. tydi: A multi-lingual benchmark for dense retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137.

Zhang, X., Thakur, N., Ogundepo, O., Kamalloo, E., Alfonso-Hermelo, D., Li, X., Liu, Q., Rezagholizadeh, M., and Lin, J. (2023c). MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

Zhang, Y. and Barzilay, R. (2015). Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1857–1867, Lisbon, Portugal.

Zhang, Y., Gaddy, D., Barzilay, R., and Jaakkola, T. (2016). Ten pairs to tag – multilingual POS tagging via coarse mapping between embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1307–1317, San Diego, California.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy.

Zhao, W., Eger, S., Bjerva, J., and Augenstein, I. (2020a). Inducing language-agnostic multilingual representations. *arXiv preprint arXiv:2008.09112*.

Zhao, W., Glavaš, G., Peyrard, M., Gao, Y., West, R., and Eger, S. (2020b). On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1656–1671, Online.

Zheng, H., Zhang, X., Chi, Z., Huang, H., Tan, Y., Lan, T., Wei, W., and Mao, X.-L. (2022). Cross-lingual phrase retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4193–4204, Dublin, Ireland.

Zhou, D., Truran, M., Brailsford, T., Wade, V., and Ashman, H. (2012). Translation techniques in cross-language information retrieval. *ACM Computing Surveys (CSUR)*, 45(1):1–44.

Zhou, L., Ding, L., and Takeda, K. (2020). Zero-shot translation quality estimation with explicit cross-lingual patterns. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1068–1074, Online.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Zhuang, S., Ren, H., Shou, L., Pei, J., Gong, M., Zuccon, G., and Jiang, D. (2022). Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128*.

Zouhar, V., Dhuliawala, S., Zhou, W., Daheim, N., Kocmi, T., Jiang, Y. E., and Sachan, M. (2023). Poor man's quality estimation: Predicting reference-based MT metrics without the reference. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1311–1325, Dubrovnik, Croatia.

Zuo, S., Zhang, Q., Liang, C., He, P., Zhao, T., and Chen, W. (2022). MoEBERT: from BERT to mixture-of-experts via importance-guided adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1610–1623, Seattle, United States.

Zweigenbaum, P., Sharoff, S., and Rapp, R. (2018). Overview of the third BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of 11th workshop on building and using comparable corpora*, pages 39–42.

# Appendix A

# Wikipedia over Time

| Year | EN | EN % | Top-10 | Top-10 % | Total |
|---|---|---|---|---|---|
| 2023 | 6.8M | 10.9% | 30.4M | 48.8% | 62.3M |
| 2022 | 6.6M | 10.9% | 29.7M | 49.4% | 60.2M |
| 2021 | 6.4M | 11.1% | 29.2M | 50.3% | 58.0M |
| 2020 | 6.2M | 11.2% | 28.4M | 51.2% | 55.5M |
| 2019 | 6.0M | 11.6% | 27.8M | 53.6% | 51.7M |
| 2018 | 5.8M | 11.8% | 27.2M | 55.2% | 49.3M |
| 2017 | 5.5M | 11.8% | 26.0M | 55.6% | 46.8M |
| 2016 | 5.3M | 12.4% | 23.6M | 55.0% | 42.9M |
| 2015 | 5.0M | 13.4% | 19.5M | 52.1% | 37.4M |
| 2014 | 4.7M | 13.9% | 17.9M | 52.8% | 33.9M |
| 2013 | 4.4M | 14.6% | 16.2M | 53.5% | 30.2M |
| 2012 | 4.1M | 17.2% | 13.5M | 56.6% | 23.8M |
| 2011 | 3.7M | 18.0% | 12.1M | 59.0% | 20.6M |
| 2010 | 3.4M | 19.7% | 10.5M | 60.4% | 17.3M |
| 2009 | 3.0M | 20.8% | 9.0M | 62.5% | 14.4M |
| 2008 | 2.5M | 21.2% | 7.5M | 63.3% | 11.8M |
| 2007 | 2.0M | 22.2% | 5.9M | 65.4% | 9.0M |
| 2006 | 1.4M | 24.1% | 4.1M | 69.9% | 5.8M |
| 2005 | 827K | 27.6% | 2.3M | 74.9% | 3.0M |
| 2004 | 419K | 32.2% | 1.0M | 77.5% | 1.3M |
| 2003 | 184K | 46.5% | 359K | 90.7% | 396K |
| 2002 | 98K | 72.1% | 135K | 99.3% | 136K |
| 2001 | 17K | 89.5% | 19K | 100% | 19K |

**Table A.1:** Distribution of number of Wikipedia articles over time. For each year, we show the number of English Wikipedia articles (EN), the number of articles belonging to the ten largest languages (Top-10) and the total number of articles.

| 2023 | | 2022 | | 2021 | | 2020 | | 2019 | | 2018 | | 2017 | | 2016 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| en | 6.76M | en | 6.6M | en | 6.43M | en | 6.22M | en | 5.99M | en | 5.8M | en | 5.5M | en | 5.3M |
| ceb | 6.12M | ceb | 6.13M | ceb | 6.09M | ceb | 5.46M | ceb | 5.38M | ceb | 5.4M | ceb | 5.4M | sv | 3.7M |
| de | 2.87M | de | 2.76M | sv | 2.77M | sv | 3.46M | sv | 3.75M | sv | 3.8M | sv | 3.7M | ceb | 3.6M |
| fr | 2.58M | sv | 2.56M | de | 2.65M | de | 2.52M | de | 2.38M | de | 2.3M | de | 2.1M | de | 2.0M |
| sv | 2.57M | fr | 2.48M | fr | 2.39M | fr | 2.28M | fr | 2.17M | fr | 2.1M | fr | 1.9M | nl | 1.9M |
| nl | 2.15M | nl | 2.11M | nl | 2.08M | nl | 2.04M | nl | 1.99M | nl | 2M | nl | 1.9M | fr | 1.8M |
| ru | 1.96M | ru | 1.88M | ru | 1.78M | ru | 1.69M | ru | 1.59M | ru | 1.5M | ru | 1.4M | ru | 1.4M |
| es | 1.92M | es | 1.83M | es | 1.74M | it | 1.66M | it | 1.57M | es | 1.5M | es | 1.4M | es | 1.3M |
| it | 1.84M | it | 1.79M | it | 1.73M | es | 1.65M | es | 1.57M | it | 1.5M | it | 1.4M | it | 1.3M |
| arz | 1.62M | arz | 1.62M | arz | 1.51M | pl | 1.45M | pl | 1.38M | pl | 1.3M | pl | 1.3M | war | 1.3M |
| 2015 | | 2014 | | 2013 | | 2012 | | 2011 | | 2010 | | 2009 | | 2008 | |
| en | 5.0M | en | 4.7M | en | 4.4M | en | 4.1M | en | 3.7M | en | 3.4M | en | 3.0M | en | 2.5M |
| sv | 2.4M | sv | 2.0M | de | 1.7M | de | 1.6M | de | 1.4M | de | 1.2M | de | 1.1M | de | 915K |
| de | 1.9M | de | 1.8M | nl | 1.7M | fr | 1.3M | fr | 1.2M | fr | 1.0M | fr | 880K | fr | 733K |
| nl | 1.8M | nl | 1.8M | sv | 1.6M | nl | 1.1M | nl | 997K | it | 756K | pl | 651K | ja | 551K |
| fr | 1.7M | fr | 1.6M | fr | 1.5M | it | 981K | it | 869K | pl | 752K | ja | 643K | pl | 551K |
| ceb | 1.7M | war | 1.3M | ru | 1.1M | es | 949K | es | 858K | ja | 725K | it | 640K | it | 520K |
| ru | 1.3M | ru | 1.2M | es | 1.1M | ru | 937K | pl | 853K | es | 691K | nl | 569K | nl | 501K |
| war | 1.3M | it | 1.2M | it | 1.1M | pl | 931K | ru | 795K | nl | 653K | es | 543K | pt | 428K |
| es | 1.2M | ceb | 1.2M | pl | 1M | ja | 842K | ja | 787K | pt | 636K | pt | 509K | es | 425K |
| it | 1.2M | es | 1.1M | war | 959K | pt | 733K | pt | 685K | ru | 633K | ru | 471K | ru | 342K |
| 2007 | | 2006 | | 2005 | | 2004 | | 2003 | | 2002 | | | | | |
| en | 2.0M | en | 1.4M | en | 827K | en | 419K | en | 184K | en | 98K | | | | |
| de | 747K | de | 568K | de | 354K | de | 189K | de | 45K | de | 11K | | | | |
| fr | 588K | fr | 407K | fr | 211K | ja | 95K | ja | 25K | pl | 6.6K | | | | |
| ja | 452K | pl | 323K | ja | 175K | fr | 70K | fr | 22K | eo | 4.3K | | | | |
| pl | 443K | ja | 310K | pl | 153K | sv | 52K | nl | 18K | nl | 4.1K | | | | |
| nl | 387K | nl | 256K | it | 123K | pl | 47K | pl | 17K | fr | 3.9K | | | | |
| it | 381K | it | 223K | nl | 117K | nl | 46K | sv | 17K | sv | 3.9K | | | | |
| pt | 329K | pt | 202K | sv | 117K | es | 35K | es | 13K | es | 2.4K | | | | |
| es | 306K | sv | 188K | pt | 90K | pt | 28K | eo | 9.5K | da | 0.54K | | | | |
| sv | 251K | es | 176K | es | 80K | it | 27K | da | 8.8K | it | 0.51K | | | | |

**Table A.2:** For each year, we list the ten largest Wikipedia language editions (Top-10) and their total number of articles.

For Figure 1.1 (Chapter 1), Table A.1 and Table A.2 we use data provided by different Wikimedia projects. We specifically use article counts from Dec 2002 until Dec 2018 provided by Wikistats 1 (Zachte, 2019).[1] Article counts from 2019 until 2023 are provided by (Wikimedia Meta-Wiki, 2019, 2020, 2021) and (Wikimedia Commons, 2022, 2023). For those, we use article counts from Dec 31 of each year.

[1] https://stats.wikimedia.org/EN/TablesArticlesTotal.htm

# Appendix B

# Further Analysis of CLEF 2003

## B.1 Information Asymmetry in CLEF

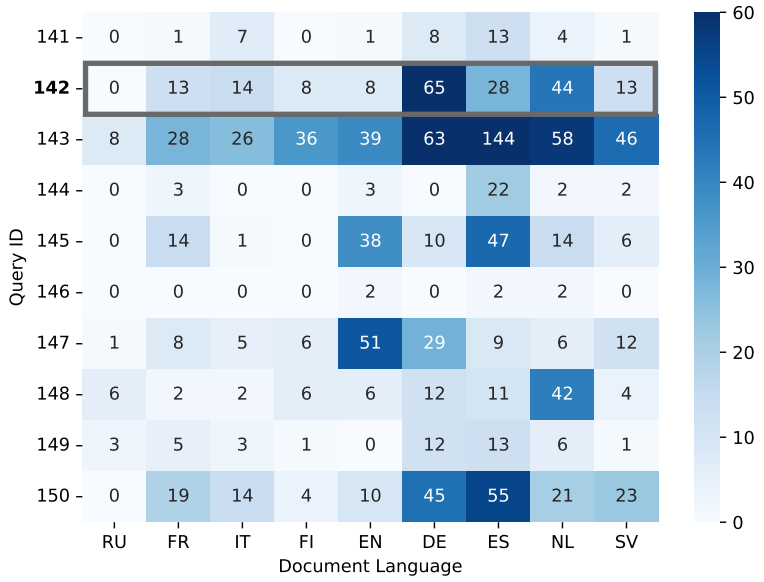


**Figure B.1:** Distribution of relevant documents across all nine CLEF document languages on a sample of ten CLEF 2003 queries (Braschler, 2004). The highlighted query (QID: 141) and two of its relevant documents are shown in Section 3.1.

## B.2 Query and Document Token Distribution

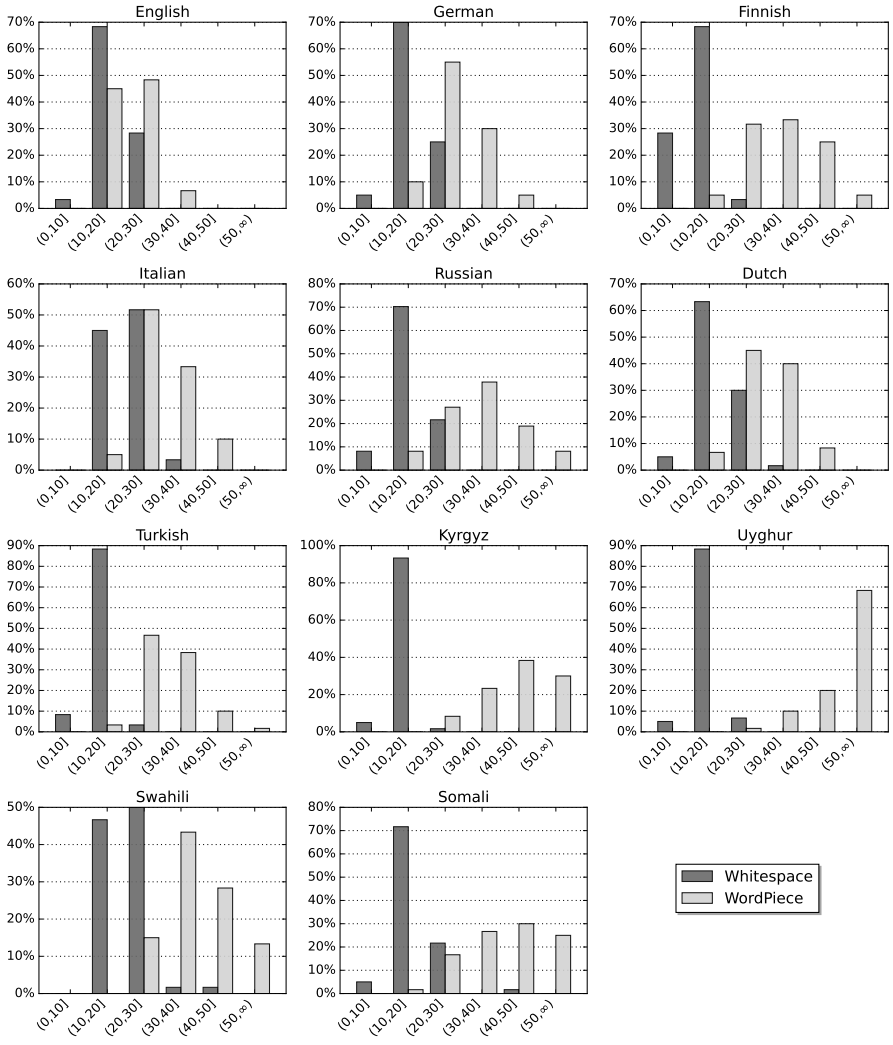


**Figure B.2:** CLEF 2003 relative query length distribution based on whitespace tokenization and WordPiece tokenization (Wu et al., 2016) used by mBERT (Devlin et al., 2019). The bottom five languages are not part of the original CLEF dataset and have been contributed by us in Chapter 8 (Turkish, Kyrgyz, Uyghur) and by Bonab et al. (2019) (Swahili, Somali).

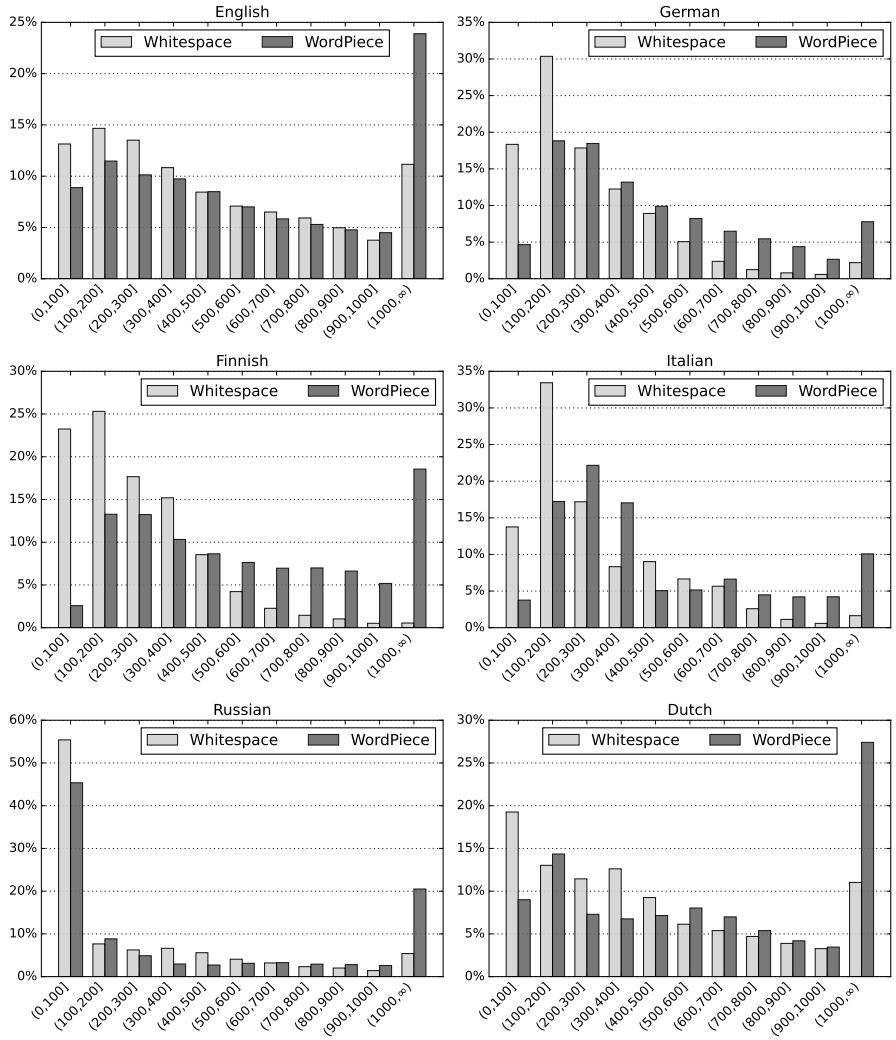


**Figure B.3:** CLEF 2003 relative document length distribution based on whitespace tokenization and WordPiece tokenization (Wu et al., 2016) used by mBERT (Devlin et al., 2019). Documents are formed by concatenating title and body text. The distribution for the BPE tokenizer is similar to WordPiece (Sennrich et al., 2016) (used by XLM (Conneau and Lample, 2019)) and has been left out for brevity.

# Appendix C

# Experimental Details for Chapter 7

| | EN | DE | RU | AR | NL | IT | AVG |
|---|---|---|---|---|---|---|---|
| `Zero-shot` | 35.4 | 25.5 | 25.5 | 22.6 | 28.1 | 27.3 | 25.8 |
| `Fine-tuning` | 35.4 | 30.8 | 29.6 | 27.8 | 31.5 | 31.5 | 30.2 |
| *Bilingual Code Switching* | | | | | | | |
| $p = 0.25$ | – | 26.7 | 26.3 | 23.9 | 27.9 | 27.7 | 26.5 |
| $p = 0.50$ | – | 27.1 | 26.1 | 23.8 | 28.5 | 27.6 | 26.6 |
| $p = 0.75$ | – | 26.6 | 25.9 | 23.8 | 27.9 | 28.1 | 26.5 |
| $p = 1.00$ | – | 25.4 | 24.6 | 22.4 | 27.3 | 26.8 | 25.3 |
| *Multilingual Code Switching* | | | | | | | |
| $p = 0.25$ | 35.1 | 27.1 | 25.1 | 21.8 | 28.2 | 27.7 | 26.0 |
| $p = 0.50$ | 34.6 | 26.7 | 25.4 | 22.0 | 27.8 | 27.4 | 25.8 |
| $p = 0.75$ | 33.9 | 26.1 | 25.4 | 22.2 | 27.7 | 27.2 | 25.7 |
| $p = 1.00$ | 31.7 | 25.8 | 25.3 | 21.8 | 26.9 | 26.5 | 25.3 |

**Table C.1:** Code Switching ablation results for different translation probabilities $p$. We present the results on seen languages in terms of mean average precision. The average is computed over all languages, excluding English.

| | EN→X | | | | DE→X | | | AR→X | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DE | IT | AR | RU | IT | NL | RU | IT | RU | AVG |
| `Zero-Shot` | 28.7 | 24.0 | 15.0 | 19.1 | 20.5 | 20.5 | 13.5 | 8.3 | 7.5 | 17.5 |
| `Fine-tuning` | 30.2 | 31.0 | 26.9 | 28.8 | 27.7 | 28.5 | 26.5 | 24.5 | 23.5 | 27.5 |
| *Bilingual Code Switching* | | | | | | | | | | |
| $p = 0.25$ | 26.9 | 26.9 | 19.0 | 22.7 | 21.5 | 23.6 | 18.6 | 16.2 | 14.9 | 21.1 |
| $p = 0.50$ | 27.4 | 27.9 | 23.5 | 23.5 | 21.1 | 23.6 | 18.7 | 16.5 | 15.1 | 21.9 |
| $p = 0.75$ | 27.4 | 28.6 | 20.7 | 23.5 | 21.4 | 23.3 | 18.5 | 16.7 | 15.3 | 21.7 |
| $p = 1.00$ | 26.3 | 27.8 | 20.1 | 23.1 | 20.1 | 22.4 | 17.9 | 14.5 | 13.8 | 20.7 |
| *Multilingual Code Switching* | | | | | | | | | | |
| $p = 0.25$ | 26.9 | 26.4 | 18.1 | 22.1 | 20.3 | 23.6 | 18.2 | 15.2 | 14.4 | 20.6 |
| $p = 0.50$ | 27.3 | 27.0 | 19.2 | 23.0 | 20.7 | 23.6 | 18.9 | 16.5 | 15.5 | 21.3 |
| $p = 0.75$ | 26.9 | 26.8 | 22.9 | 22.9 | 20.3 | 23.2 | 18.4 | 16.1 | 15.0 | 21.4 |
| $p = 1.00$ | 25.4 | 24.9 | 18.0 | 21.8 | 19.8 | 22.6 | 18.3 | 15.2 | 14.6 | 20.1 |

**Table C.2:** Code Switching ablation results for different translation probabilities $p$. We present the results on seen cross-lingual language pairs in terms of mean average precision (MAP).

# Appendix D

# Experimental Details for Chapter 8

| Reduction factor | Bottleneck dimension | Number of mPLM Parameters | Equivalent Reduction Factor (Sparsity) |
|---|---|---|---|
| 1 | 768 | 14,174,208 | 8.47% |
| 2 | 384 | 7,091,712 | 4.24% |
| 4 | 192 | 3,550,464 | 2.12% |
| 8 | 96 | 1,779,840 | 1.06% |
| 16 | 48 | 894,528 | 0.53% |
| 32 | 24 | 451,872 | 0.27% |

**Table D.1:** Overview of (equivalent) reduction factors. The bottleneck dimension is to the hidden size after down-projection.

## D.1 Ablation study: Reduction factors

Complementary to our analysis in Section 8.4.2 (Figure 8.3), we now explain how to calculate for a given adapter reduction factor the equivalent reduction factor for Sparse Fine-Tuning Masks (SFTM). The equivalent reduction factor is based on the size of the underlying pre-trained language model, mBERT[1], which has 167,357,185 parameters, 12 layers and a hidden size of $h = 768$. For example, suppose an adapter uses a reduction factor of 16, resulting into a bottleneck dimension of $d = 768/16 = 48$. Consequently, each layer $i$ is augmented by a down-projection matrix $\mathbf{D}_i \in \mathbb{R}^{768\times 48}$, an up-projection matrix $\mathbf{U}_i \in \mathbb{R}^{48\times 768}$ and their respective bias vectors $\mathbf{b}_i^d \in \mathbb{R}^{48}$ and $\mathbf{b}_i^u \in \mathbb{R}^{768}$. The total number of trainable adapter parameters is $12 \cdot (768 \cdot 48 \cdot 2) + 12 \cdot (48 + 768) = 894,528$. The

[1] `bert-base-uncased-multilingual`

| | TR→X | | | | | EN→X | | | | DE→X | | | FI→X | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RF | EN | IT | DE | FI | RU | FI | IT | RU | DE | FI | IT | RU | IT | RU | **AVG** |
| 1 | .210 | .213 | .203 | .248 | .160 | .342 | .334 | .190 | .299 | .318 | .288 | .219 | .186 | .144 | .240 |
| 2 | .222 | .200 | .208 | .271 | .248 | .346 | .341 | .209 | .308 | .317 | .277 | .252 | .192 | .167 | .254 |
| 4 | .241 | .214 | .220 | .285 | .231 | .354 | .351 | .208 | .316 | .329 | .276 | .265 | .220 | .203 | .265 |
| 8 | .246 | .245 | .226 | .283 | .261 | .362 | .350 | .236 | .314 | .366 | .288 | .300 | .215 | .222 | .280 |
| 16 | **.252** | .234 | .222 | .267 | **.267** | .366 | .366 | **.248** | .314 | .350 | .302 | **.315** | .220 | **.234** | **.283** |
| 32 | **.252** | .236 | **.228** | .278 | .252 | .362 | **.374** | .222 | .316 | .347 | .298 | .275 | .213 | .215 | .276 |
| 1 | .213 | .212 | .217 | .280 | .153 | .349 | .327 | .188 | .299 | .345 | .291 | .188 | .230 | .165 | .247 |
| 2 | .239 | **.252** | .232 | **.316** | .162 | .359 | .348 | .191 | .310 | **.391** | **.323** | .195 | **.255** | .160 | .267 |
| 4 | .228 | .240 | .208 | .235 | .131 | .391 | .356 | .210 | .314 | .350 | .295 | .181 | .233 | .126 | .250 |
| 8 | .051 | .098 | .074 | .063 | .023 | .399 | .361 | .235 | **.318** | .107 | .155 | .043 | .119 | .036 | .149 |
| 16 | .048 | .075 | .070 | .052 | .023 | **.404** | .353 | .219 | .317 | .097 | .144 | .042 | .093 | .034 | .141 |
| 32 | .049 | .061 | .065 | .051 | .026 | .382 | .345 | .194 | .313 | .095 | .134 | .042 | .080 | .034 | .134 |

**Table D.2:** CLIR results w.r.t. different (equivalent) reduction factors (RF). The top half shows results obtained with document language adapters ($+\text{RA} + \text{LA}^{\text{Doc}}$), the bottom half shows results obtained with query language masks ($+\text{RM} + \text{LM}^{\text{Query}}$). For our cross-lingual retrieval experiments we use $\text{DIST}_{\text{DmBERT}}$ as a first-stage retriever (cf. Section 5.3.1). We highlight the best performance in **bold**.

| RF | FA | ZH | RU | EN | FI | DE | IT | RU | **AVG** |
|---|---|---|---|---|---|---|---|---|---|
| 1 | .351 | .291 | .241 | .506 | .543 | .462 | .485 | .356 | .404 |
| 2 | .350 | .290 | .250 | .519 | .544 | .456 | .482 | .370 | .408 |
| 4 | .372 | .284 | .261 | **.521** | .531 | **.465** | .499 | .402 | .417 |
| 8 | .365 | .296 | .260 | .501 | .548 | .460 | **.502** | .388 | .415 |
| 16 | .372 | .290 | .262 | .519 | .537 | .457 | .495 | .389 | .415 |
| 32 | .361 | .292 | **.265** | .515 | .540 | .463 | .500 | **.418** | .419 |
| 1 | .386 | .295 | .264 | .512 | .555 | .464 | .496 | .381 | .419 |
| 2 | **.398** | **.307** | .264 | .515 | **.564** | .459 | **.502** | .379 | **.424** |
| 4 | .247 | .235 | .264 | .509 | .547 | .457 | .490 | .337 | .386 |
| 8 | .058 | .043 | .212 | .518 | .320 | .338 | .416 | .120 | .253 |
| 16 | .046 | .035 | .208 | .513 | .272 | .319 | .374 | .107 | .234 |
| 32 | .043 | .033 | .200 | .507 | .236 | .296 | .344 | .119 | .222 |

**Table D.3:** Ablation results of zero-shot cross-lingual transfer for monolingual retrieval (MoIR) with respect to different (equivalent) reduction factors (RF). For our monolingual experiments we use BM25 as a first-stage retriever (cf. Section 3.3). We highlight the best performance in **bold**.

equivalent reduction factor for SFTMs, i.e. the mask size corresponds to 894K and has a sparsity of 0.53% ($894K/167M$).

In Table D.2 and Table D.3 we highlight the detailed results of our ablation study presented in Section 8.4.2. We can see that the reduction factor leading to the best performance (highlighted in bold) is highly dependent on the target language(s). On both tasks, CLIR and MoIR, adapters are generally more robust towards different reduction factors than SFTMs.

# Appendix E

# Train and Test Token Distribution

| Test Language | #sentences | #tokens |
|---|---|---|
| Erzya (myv) | 1550 | 15790 |
| Faroese (fo) | 1208 | 10002 |
| Amharic (am) | 1074 | 10010 |
| Kazakh (kk) | 1047 | 10007 |
| Bambara (bm) | 1026 | 13823 |
| Thai (th) | 1000 | 22322 |
| Buryat (bxr) | 908 | 10032 |
| Breton (br) | 888 | 10054 |
| North Sami (sme) | 865 | 10010 |
| Cantonese (yue) | 650 | 6264 |
| Upper Sorbian (hsb) | 623 | 10736 |
| Maltese (mt) | 518 | 11073 |
| Armenian (hy) | 470 | 11438 |
| Irish (ga) | 454 | 10138 |
| Coptic (cop) | 267 | 6541 |
| Telugu (te) | 146 | 721 |
| Tamil (ta) | 120 | 1989 |
| Yoruba (yo) | 100 | 2666 |
| Belarusian (be) | 68 | 1382 |
| Lithuanian (lt) | 55 | 1060 |
| Average | 1094 | 8802 |

**Table E.1:** List of 20 unseen test languages in the Universal Dependencies dataset (v2.3 collection). Number of sentences (#sentences) and total token count (#tokens).

| Train Language | #sentences | #tokens |
|---|---|---|
| Estonian (et) | 24384 | 341122 |
| Korean (ko) | 23010 | 296446 |
| Latin (la) | 16809 | 293306 |
| Norwegian (no) | 15696 | 243887 |
| Finnish (fi) | 14981 | 127602 |
| French (fr) | 14450 | 354699 |
| Spanish (es) | 14305 | 444617 |
| German (de) | 13814 | 263804 |
| Polish (pl) | 13774 | 104750 |
| Hindi (hi) | 13304 | 281057 |
| Catalan (ca) | 13123 | 417587 |
| Italian (it) | 13121 | 276019 |
| English (en) | 12543 | 204585 |
| Dutch (nl) | 12269 | 186046 |
| Czech (cs) | 10160 | 133637 |
| Portuguese (pt) | 9664 | 255755 |
| Bulgarian (bg) | 8907 | 124336 |
| Slovak (sk) | 8483 | 80575 |
| Romanian (ro) | 8043 | 185113 |
| Latvian (lv) | 7163 | 113405 |
| Japanese (ja) | 7133 | 160419 |
| Croatian (hr) | 6983 | 154055 |
| Slovenian (sl) | 6478 | 112530 |
| Arabic (ar) | 6075 | 223881 |
| Basque (eu) | 5396 | 72974 |
| Ukrainian (uk) | 5290 | 88043 |
| Hebrew (he) | 5241 | 137721 |
| Persian (fa) | 4798 | 121064 |
| Indonesian (id) | 4477 | 97531 |
| Danish (da) | 4383 | 80378 |
| Swedish (sv) | 4303 | 66645 |
| Urdu (ur) | 4043 | 108690 |
| Chinese (zh) | 3997 | 98608 |
| Russian (ru) | 3850 | 75964 |
| Turkish (tr) | 3685 | 37918 |
| Serbian (sr) | 2935 | 65764 |
| Galician (gl) | 2272 | 79327 |
| Greek (el) | 1662 | 42326 |
| Uyghur (ug) | 1656 | 19262 |
| Vietnamese (vi) | 1400 | 20285 |
| Afrikaans (af) | 1315 | 33894 |
| Hungarian (hu) | 910 | 20166 |
| Average (avg) | 8483 | 158233 |

**Table E.2:** List of 42 source languages on which we trained monolingual parsers. Number of sentences (#sentences) and total token count (#tokens).