Multilingual Embeddings: Data, Training, and Understanding

by

Xinyu Crystina Zhang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2026

© Xinyu Crystina Zhang 2026

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Member: **Karen Livescu**
Professor
Computer Science
Toyota Technological Institute at Chicago

Supervisor(s): **Jimmy Lin**
Professor, Chair
David R. Cheriton School of Computer Science
University of Waterloo

Internal Member: **Charles Clarke**
Professor
David R. Cheriton School of Computer Science
University of Waterloo

Internal Member: **Wenhu Chen**
Assistant Professor
David R. Cheriton School of Computer Science
University of Waterloo

Internal-External Member: **Olga Vechtomova**
Professor
Department of Management Science and Engineering
David R. Cheriton School of Computer Science
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis is based in part on five published papers on multilingual retrieval datasets, multilingual dense retrieval models, and the analysis of cross-lingual transfer, together with additional synthesis developed for this dissertation. Xinyu Crystina Zhang is the first author of three of them [149, 150, 146], co-first author of one [151], and co-author of [79]. For the first and co-first author works, she contributed to the research ideation, dataset construction, experimental design, implementation, experiment execution, result analysis, and paper writing. For the co-author work, she contributed to the research ideation, experimental design, result analysis, and paper writing. The publications and co-authors are listed below. An asterisk denotes equal contribution.

- *Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval*. Xinyu Zhang, Xueguang Ma, Peng Shi, Jimmy Lin. In Proceedings of the 1st Workshop on Multilingual Representation Learning, 2021. (Chapter 3)
- *MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages*. Xinyu Zhang*, Nandan Thakur*, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, Jimmy Lin. In Transactions of the Association for Computational Linguistics, 2023. (Chapter 3)
- *Toward Best Practices for Training Multilingual Dense Retrieval Models*. Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, Jimmy Lin. In ACM Transactions on Information Systems, 2023. (Chapter 4 and 5)
- *The Impact of Incidental Multilingual Text on Cross-Lingual Transfer in Monolingual Retrieval*. Andrew Liu*, Edward Xu*, Crystina Zhang, Jimmy Lin. In Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Proceedings, Part III, pages 165–173, 2025. (Chapter 5)
- *Tomato, Tomahto, Tomate: Do Multilingual Language Models Understand Based on Subword-Level Semantic Concepts?* Crystina Zhang, Jing Lu, Vinh Q. Tran, Tal Schuster, Donald Metzler, Jimmy Lin. In Findings of the Association for Computational Linguistics: NAACL 2025, pages 1821–1837, 2025. (Chapter 5)

Abstract

Embedding models have been a central component of modern information access systems, including search engines, question answering systems, retrieval-augmented generation, and nowadays agentic search pipelines. By converting text-space search into vector-space search, embedding models provide semantic matching beyond exact lexical overlap. However, their progress has been uneven across languages. While English dense retrieval has benefited from large training collections, mature benchmarks, and well-studied training recipes, many other languages still lack reliable retrieval resources, practical modeling guidance, and a clear understanding of why multilingual transfer works.

This thesis studies multilingual embedding models for retrieval from three connected perspectives: data, training, and understanding. First, it introduces two multilingual retrieval resources, Mr. TYDI and MIRACL. Mr. TYDI establishes the first large-scale monolingual retrieval benchmark over eleven typologically diverse languages, while MIRACL expands the setting to eighteen languages with ten-times richer human relevance annotations. Together, these datasets provide both the supervision needed to train multilingual dense retrievers and the benchmarks needed to evaluate them reliably across diverse languages and scripts.

Second, we investigate how to train multilingual dense retrievers under realistic resource conditions. Starting from the observation that plain multilingual DPR can perform only marginally better than BM25 in unsupervised settings, this part studies cases where target-language training data, target-language pretrained models, or both may be unavailable. The analysis compares pretrained backbones, translated and in-language data, multi-stage fine-tuning, cross-lingual transfer, knowledge distillation, and monolingual versus multilingual transformers. These experiments provide practical guidance for building effective multilingual retrieval models when resources differ across languages.

Third, this thesis examines how multilingual language models may understand across languages. It analyzes the roles of shared tokens across languages and their impact at the embedding finetuning stage, and then examines how language models may understand token-level semantic concepts, revealing that multilingual understanding and cross-lingual transfer largely depend on token-level semantic structures within multilingual vocabularies and embedding spaces.

Overall, this thesis contributes datasets, training strategies, and model analyses that move multilingual retrieval research from infrastructure to practice to interpretation, advancing the development of retrieval systems that can support information access more reliably across languages, scripts, and resource conditions.

Acknowledgements

This thesis would not have been possible without the people who guided, supported, challenged, and accompanied me throughout this journey. I would first like to thank my supervisor, Jimmy Lin, who guided me since my first publication and set me on the path of research. His mentorship shaped the way I think about research, writing, and presenting my work. I am also deeply grateful for the freedom and flexibility he provided to explore my own research interests and grow as an independent researcher.

I would like to thank Charles, Wenhu, Olga, and Karen, for being my committee members and providing valuable feedback and suggestions. In addition, I am grateful for the support of the David R. Cheriton Scholarship from 2023 to 2025, which not only alleviated my financial worries but also encouraged me to continue working on multilingual research.

Among the most valuable memories from this journey are the people I met along the way. Waterloo does not have pleasant weather for half of the year, and it is the people around that make it a pleasant stay. I would like to thank my labmates and collaborators, Ralph, Nandan, Xueguang, Minghan, Jack, and many more, for not only the interesting conversations and collaborations, for the help and support, but also the many fun moments we had all together. Badminton and bouldering were my favorite refreshing activities, and I could not have enjoyed more during the ski season when we travelled together to Tremblant.

I'm also grateful to have had many great mentors during my internships: Jing, Vinh, Tal, and Donald at Google, Sebastian and Patrick at Cohere, and Andrew at MPI, who provided not only new perspectives on the joint projects, but also generous guidance and support beyond the projects themselves, contributing significantly to my personal growth.

I would also like to thank Charlie, Renee, Tamer, Victor, and all other faculty members in Waterloo who showed great kindness, support, and guidance to me during the job search. The one-on-one meetings and feedback greatly encouraged me during the stressful time, and drives me to provide the same support to others in the future.

I would never have chosen to start my PhD journey without the rewarding experiences I had during my undergraduate research programs and internships. I would like to thank Gary Shueng Han Chan, Lei Chen, and Sung Kim for their guidance, and for giving me the opportunity to work on projects where I could learn by exploring, I would also like to thank all my mentors, Gary and Kyungmin, who spent countless hours guiding me, encouraging me, and helping me grow. I've always believed that my two stays in third-language countries have unconsciously shaped the choice of my research directions.

Although it has been long, I would like to thank my local friends, who filled my memory in a place where I did not know the local language with joy, and for touring me around.

I would like to thank my family. To my parents, thanks for raising me up with your best efforts, for respecting my choices even if they are not always the same as what you expected, and for always letting me know that there is a home to return to. To Jerry, my beloved partner, thanks for being by my side, sharing my every moment of joy and sorrow, for being my support and trusting me even when I do not, and most importantly, for teaching me how to love. Although not related by blood or law, I would like to thank "the other" Xinyu and Xueguang, the other members in my "cat family", for all the fun we had, all the companionship and support during the best and toughest times. It never occurred to me that I would find a place called family, thousands of miles away from home, where I could take a breath, be at ease, and stay true to who I am.

Outside the research groups, I am also so fortunate to have had friends along the way, who carried me through the ordinary and extraordinary parts of this journey: the celebrations, the doubts, the quiet days, and the times when distance made support feel even more precious. I would like to thank Yue, it's amazing that we can still understand and support each other without sharing same path for over ten years since high school. I would like to thank Emily, Ivana, and Rebecca. The readings we shared and our trips together brought me out of everyday life, and gave me space to wonder and reflect. These conversations reminded me that there is a life beyond deadlines and experiments, and that learning can be gentle, expansive, and shared.

Dedication

To my parents, my husband, and my "cat family".

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract v

Acknowledgements vi

Dedication viii

List of Figures xiii

List of Tables xvi

1 Introduction 1

1.1 Overview 2

1.2 Organization 3

2 Background 4

2.1 Ad-hoc Retrieval 4

2.1.1 Bag-of-Words Algorithms 4

2.1.2 Embedding Models . . . 5
2.1.3 Cross-Encoder Models . . . 6
2.2 Multilingual Retrieval . . . 6
2.2.1 Prior to Embedding-based Retrievers . . . 7
2.2.2 Embedding-based Retrievers . . . 7
2.3 Beyond Retrieval . . . 8
2.3.1 MLM pretraining . . . 8
2.3.2 Natural Language Understanding Tasks . . . 8
2.3.3 Impact of Cross-Lingual Shared Tokens . . . 9
2.3.4 Token-Level Semantic Concepts . . . 10

3 Dataset and Benchmarks **11**
3.1 Mr. TYDI . . . 12
3.1.1 Background . . . 14
3.1.2 Dataset Construction . . . 16
3.1.3 Baselines . . . 17
3.1.4 Experiment Results . . . 19
3.2 MIRACL . . . 21
3.2.1 Comparison to Mr. TYDI . . . 22
3.2.2 Overview . . . 23
3.2.3 Dataset Construction . . . 26
3.2.4 Quality Control . . . 29
3.2.5 Analysis . . . 33
3.2.6 Experiments . . . 36
3.3 Discussion . . . 37

4 Training **39**
4.1 Experimental Design . . . 40
4.1.1 Design Space . . . 40
4.1.2 Datasets . . . 42
4.1.3 Data Preparation . . . 43
4.1.4 Experimental Setup . . . 44
4.2 Have Model, No Data . . . 47
4.2.1 English Pre–Fine-Tuning Data . . . 50
4.2.2 Multilingual Pre–Fine-Tuning Data . . . 51
4.2.3 Enhanced Backbones . . . 53
4.2.4 Discussion: Relevance Transfer and Language Adaptation . . . 54
4.2.5 Knowledge Distillation . . . 55
4.3 Have Model and Data in Target Language: Multilingual BERT . . . 58
4.3.1 English/Multilingual Pre–Fine-Tuning Data . . . 58
4.3.2 Multilingual Fine-Tuning Data . . . 59
4.3.3 Enhanced Backbones and Hard Negatives . . . 60
4.4 Have Model and Data in Non-Target Language: Multilingual BERT . . . 62
4.5 Have Model and Data: Mono vs. Multilingual BERT . . . 66
4.6 Have Data, No Model . . . 68

5 Understanding **70**
5.1 Shared Tokens . . . 70
5.1.1 Shared Word Vocabularies . . . 71
5.1.2 Shared Subword Vocabularies . . . 74
5.2 Impact of Incidental Multilingual Tokens in Embedding Finetuning . . . 75
5.2.1 Experimental Design and Setup . . . 75
5.2.2 Results . . . 77
5.3 Token-Level Semantic Concepts . . . 77

5.3.1 Operations and Pipelines . . . 80
5.3.2 Evaluation Tasks and Datasets . . . 81
5.4 Token-Level Semantic Concepts Results . . . 83
5.4.1 Semantic Grouping . . . 83
5.4.2 Confounding Effect from Embedding Parameters . . . 86
5.4.3 Backbones . . . 87
5.4.4 Semantic Categories among Grouped Subwords . . . 87
5.4.5 Cross-lingual Transfer . . . 89
5.4.6 Ablation on Distance Metric . . . 90
5.5 Discussion on the Behaviors Difference across Tasks . . . 91
5.6 Conclusion . . . 91

6 Conclusion **95**

References **97**

List of Figures

3.1 Comparison between TYDI, XOR-TYDI, and Mr. TYDI with an example in Telugu. The green blocks indicate relevant passages and the red blocks indicate non-relevant passages. 13

3.2 MRR@100 of mDPR and the sparse–dense hybrid normalized with respect to BM25 for each language (left); corresponding pairs for each language plotted as a scatter plot (right). 21

3.3 Examples of missing relevant passages in TYDI (and thus Mr. TYDI) for the query "*Which is the largest marine animal?*" Since only relevant passages in the top Wikipedia article are included, other relevant passages are missed. 22

3.4 Diagram illustrating the MIRACL annotation workflow. 26

3.5 Examples of disagreements between annotators and reviewers. 30

4.1 The pipeline of (a) mDPR init. mBERT, (b) mDPR init. mContriever, and (c) mDDR and mDDR$^+$, where blocks with dashed outlines indicate modules that are frozen during training. (PT: pretrain; FT: fine-tune) 46

5.1 Words with similar meanings or inflectional changes fall under semantic concepts, as indicated by the colors. The sentence "They collected the rotted tomatoes." is from XNLI [25]. 78

5.2 Illustrations of the operations that modify the word embeddings and the pipelines composed of these operations. "Cont. Pt": Continual Pretraining, whose illustration is skipped due to its prevalence. The colored rows represent the embeddings of the subwords and the coordinates depict their spatial distances. The shapes indicate the underlying semantics and the colors indicate the languages of the subwords. Under the "PIPELINES", the solid boxes denote the required operations that are core to the pipeline or essential to the model functionality; the dashed boxes denote the optional operations that are only to provide additional improvement. Better viewed in color. 79

5.3 Results of mBERT with vocabulary and embeddings after semantic grouping (SG) or simply reduced size (First-k). **x-axis**: the grouping ratio r_G in log scale. The background colors indicate the relative performance to the oracle results, i.e., continual pretrained mBERT with full vocabulary, indicated by the green dashed lines on top. green: >90%, yellow: 70%–90%, red: 50%–70%. The scores of First-k at $r_G = \{5\%, 10\%\}$ are skipped in the figures as they greatly skew the y-axis. 83

5.4 Results of mBERT after semantic grouping (SG) with and without applying cross-lingual subword alignment (CLSA). 83

5.5 Results of semantic grouping with word embedding dimension $d \in \{768, 128, 32\}$, with CLSA and continual pretraining. Background colors design is identical to Figure 5.3. The relative performance of all classification tasks are higher than 90%, thus with full green background. 84

5.6 Comparison the effectiveness of Dimension Reduction (DR) and semantic Grouping (SG) at the same level of word embedding parameters size. **x-axis**: the ratio of the word embedding size after DR or SG in log scale. The brown lines: DR with dimension $d \in \{2, 8, 32, 128, 768\}$ and full vocabulary, corresponding to embedding parameters ratio of {0.3%, 1.0%, 4.2%, 17%, 100%}. The yellow lines: SG with dimension $d = 32$ and vocabulary grouping ratio $r_G \in \{5\%, 10\%, 20\%, 40\%\}$, corresponding to embedding parameters ratio of {0.21%, 0.42%, 0.84%, 1.7%, 4.2%}. Background colors design is identical to Figure 5.3. 85

5.7 Results of SG on multiple backbones, all with CLSA applied and no continual pretraining. All models show similar trend regarding semantically grouped subwords. 86

5.8 Eight semantic tokens formed by grouped subwords on mBERT word embedding with dimension $d = 768$ and CLSA applied. The labels outside the circles are either the keyword of the cluster, or "NUMBERS"/ "PUNCTUATION" if the cluster is a collection of numbers or punctuation. 88

5.9 Results when transferring from English to other languages on four benchmarks, with CLSA and continual pretraining applied. MasakhaNER is skipped as English training data is not available. Background colors design is identical to Figure 5.3. The dashed green line denotes the zero-shot performance without semantic grouping. 89

5.10 Comparison of L2 versus cosine distance when using K-Means to group the subwords. 90

5.11 More examples of the grouped subwords on mBERT with CLSA 93

List of Tables

3.1 Comparison of select multilingual retrieval datasets. **Natural Queries** and **Natural Passages** indicate whether the queries and passages are "natural", i.e., generated by native speakers (vs. human- or machine-translated), and for queries, in natural language (vs. keywords or entities); **Human Labels** indicates human-generated labels (vs. synthetically generated labels); # **Lang** is the number of languages supported; **Avg # Q** is the average number of queries for each language; **Avg # Labels/Q** is the average number of labels provided per query; **Total # Labels** is the total number of human labels (both positive and negative) across all languages (including synthetic labels in CLIRMatrix). **Training?** indicates whether the dataset provides sufficient data for training neural models. 12

3.2 Descriptive statistics for Mr. TYDI: the number of questions (# Q), judgments (# J), and the number of passages (Corpus Size) in each language. . 18

3.3 Results of BM25 (with default and tuned parameters), mDPR, and the sparse–dense hybrid on the test set of Mr. TYDI. The symbol † indicates significant improvements over BM25 (tuned) (paired t-test, $p < 0.01$). . . . 19

3.4 Descriptive statistics for all languages in MIRACL, organized by split. # **Q**: number of queries; # **J**: number of labels (relevant and non-relevant); # **Passages**: number of passages; # **Articles**: number of Wikipedia articles from which the passages were drawn; # **Avg. Q Len**: average number of tokens per query; # **Avg. P Len**: average number of tokens per passage. Tokens are based on characters for `th`, `ja`, and `zh`, otherwise delimited by whitespace. Underlined are the new languages in MIRACL. 25

3.5 Query distribution of each split in English MIRACL, compared to SQuAD. 32

3.6 Characteristics of languages in MIRACL, including language (sub-)family, script, linguistic typologies (synthesis, word order, and gender), number of speakers (L1 and L2), and number of Wikipedia articles. Under **White Space**, ✓ indicates the language uses white space as the token delimiter; under **Gender**, ✓ indicates that gender is evident in the language. Under # **Speakers** and **Wikipedia Size**, cells are highlighted column-wise with the color gradient ranging from **green** (high values) to **red** (low values), where the **Wikipedia Size** column is identical to the # **Articles** column in Table 3.4. Underlined are the new languages not included in Mr. TYDI. 35

3.7 Baseline results on the MIRACL dev set, where "K. Avg." and "S. Avg" indicate the average scores over the known (`ar`–`zh`) and surprise languages (`de`–`yo`). (**Hyb.**: Hybrid results of BM25 and mDPR; **mCol.**: mColBERT; **mCon.**: mContriever; **in-L**: in-language fine-tuned mDPR) Best results are in **bold**. 37

4.1 HuggingFace models used in our experiments. 44

4.2 "Have model, no data": choice of pre–fine-tuning data. MRR@100 and Recall@100 across all 11 languages and an overall average on the Mr. TYDI test set. All conditions are zero-shot; the models have not seen any training data from Mr. TYDI. The first two rows are baselines using BM25. The remaining rows are zero-shot results based on mDPR (DPR initialized with mBERT) pre–fine-tuned (pFT) on different datasets. NQ: Natural Questions; MS: the original MS MARCO dataset in English; $\mathcal{L}$-mMARCO: MS MARCO automatically translated to language $\mathcal{L}$, where $\mathcal{L} \in \{\texttt{ar}, \texttt{id}, \texttt{ja}, \texttt{ru}\}$; all-mMARCO: the entire mMARCO dataset, which contains 14 languages (including English). We underline the entries where the pre–fine-tuning dataset is in (and only in) the target language, which are also the best scores in the corresponding columns (with the exception of `ru` in terms of Recall@100). 48

4.3 "Have model, no data": enhanced backbones. MRR@100 and Recall@100 of mDPR initialized from different backbones, evaluated across all 11 languages and an overall average on the Mr. TYDI test set. The backbones include mBERT, XLM-R_{Large}, mContriever, and additional MLM pretraining (mDDR and mDDR$^+$). Under the "MS pFT" block, models are pre–fine-tuned on only English MS MARCO; Under the "all-mMARCO pFT" block, models are pre–fine-tuned on all of mMARCO (14 languages). Rows (T4.2–4, T4.2–9) are identical to the rows 4 and 9 in Table 4.2. The underlined entries are the best results under the "MS pFT" block, whereas the **bold** entries are the best results under both blocks. 49

4.4 "Have model, no data": knowledge distillation. MRR@100 and Recall@100 of the cross-encoder along with mDPR that is pre–fine-tuned or distilled from the cross-encoder, evaluated across all 11 languages and an overall average on the Mr. TYDI test set. Models are pre–fine-tuned on English MS MARCO in all experiments. The first row denotes the monoXLM-R_{Large} cross-encoder and the second block captures baselines without knowledge distillation, copied from previous tables. Best *distilled* results are in **bold**. 56

4.5 "Have model and data": multilingual pre–fine-tuning and fine-tuning. MRR@100 and Recall@100 of mDPR evaluated across all 11 languages and an overall average on the Mr. TYDI test set. "MS pFT": pre–fine-tuning on English MS MARCO; "all-mMARCO pFT": pre–fine-tuning on all of mMARCO including `en`; "in-lang FT": fine-tuning on the Mr. TYDI training set from only the target language; "all FT": fine-tuning on Mr. TYDI training data from all languages; "in/out-script": fine-tuning on training data from the target language *and* languages that are in the same/different scripts; for example, if the target language is `en`, the "in-script" languages would be the target language (`en`) and all the other Latin-script languages (`fi`, `id`, `sw`) and the "out-script" languages would be the target language (`en`) and all the non-Latin-script languages (`ar`, `bn`, `ja`, `ko`, `ru`, `te`, `th`). Best results are in **bold**. 57

4.6 "Have model and data": enhanced backbone and hard negatives. MRR@100 and Recall@100 of mDPR initialized from mBERT and mContriever, evaluated across all 11 languages and an overall average on the Mr. TYDI test set. All experiments are first pre–fine-tuned on the entire mMARCO dataset, then on all Mr. TYDI training data. Rows marked with HN use the hard negative training data on Mr. TYDI prepared from the model of row (5) in Table 4.3 (mDPR init. mContriever + all-mMARCO pFT). Best results are in **bold**. 61

4.7 Matrix experiments training mDPR on the language denoted in the row, evaluating on the language in the column, reporting MRR@100. Fine-tuning is performed on the mBERT backbone directly (top) and after pre–fine-tuning on MS MARCO (bottom). For `bn`, `ko`, and `sw`, we use all available training data; for the remaining languages, we downsample to 3,300 training queries. Cell values are highlighted *column-wise*: the color gradient ranges from **blue** (high values) to **orange** (low values). Note that due to data downsampling, the diagonal results are not identical to results in Table 4.5. 63

4.8 Element-wise difference between MS pFT + FT and MS pFT (zero-shot), answering the question: "Should we always fine-tune on non-target language data?" Answer is *yes*, with the exception that further fine-tuning on English data for retrieval in non-English languages might result in overfitting. Positive values are highlighted in **blue** and negative values in **orange**, with saturation proportional to magnitude. 64

4.9 Element-wise differences between MS pFT + FT vs. FT directly, answering the question: "Does pre–fine-tuning always improve effectiveness?" For in-language training, *yes*, as the diagonal entries are always positive (in **blue**). For cross-lingual zero-shot transfer, *not necessarily so*, as can be seen by the negative (in **orange**) cases off-diagonal. However, we do observe more positive than negative cases, which indicates pre–fine-tuning may help in general. We thus believe that pre–fine-tuning is a "safer" choice. Color saturation is proportional to the magnitude of values. 65

4.10 Rows (a–d): "Have model and data": mono vs. multilingual BERT. Rows (e–h): "Have model, no data" (Best results are in **bold**). MRR@100 and Recall@100 of mDPR evaluated across all 11 languages and an overall average on the Mr. TYDI test set. "MS pFT": pre–fine-tuning on MS MARCO; "in-lang FT": fine-tuning on the Mr. TYDI training set from only the target language; "all FT": fine-tuning on the Mr. TYDI training data from all languages; "in/out-script": fine-tuning on training data from the target language *and* languages that are in the same/different scripts. Rows (T4.5–5, T4.5–6) are identical to rows (5, 6) in Table 4.5. 67

5.1 Analysis of unique whitespace-delimited tokens in Mr. TYDI. Each row represents a different corpus; we omit Japanese here because it is written without spaces. Each column indicates the percentage of words belonging to that language, based on an off-the-shelf language id package. The total number of unique words in each language is reported in the rightmost column. The values are highlighted *row-wise*: the color gradient ranges from **blue** (high values) to **orange** (low values) 71

5.2 Analysis of unique tokens produced by pretrained transformer tokenizers in Mr. TYDI. Each language cell in a column (L_c) indicates the percentage of overlapping tokens between itself and the language cell in a row (L_r) over the total number of unique tokens in L_c. For example, the `bn`–`ar` entry in Table 5.2a is 57.3%, indicating that the number of overlapping subwords produced by the mBERT tokenizer for the `ar` and `bn` corpora accounts for 57.3% of the total unique subwords in the `bn` corpus. The values are highlighted *column-wise*: the color gradient ranges from **blue** (high values) to **orange** (low values) 72

5.3 nDCG@10 on the development set of Persian (`fa`), Hindi (`hi`), Bengali (`bn`) from MIRACL. **Column**: source languages L_s; **Rows**: target languages L_t. Under each target language, the first two rows are trained on official MIRACL training data (*w/ IMT*) and data with non-self script tokens removed (*w/o IMT*). The third row (%) is the relative effectiveness of *w/o IMT* compared to *w/ IMT*, highlighted in , where higher saturation indicates higher performance. Significantly different results are marked with † ($p < 0.01$ with paired t-tests). 76

5.4 Downstream Tasks and Datasets. Granularity suggests the information level required from input data to perform the task. "P Retrieval": Passage Retrieval; "P Reranking": Passage Reranking. 82

5.5 Numerical results of experiments, where each is an averaged result of all languages per benchmark. The colors indicate the relative performance to the oracle results: green: >90%, yellow: 70%–90%, red: 50%–70% and the dark red color means <50%. 94

Chapter 1

Introduction

Information access systems are increasingly built around learned representations of text. Modern search engines, question answering systems, and retrieval-augmented generation pipelines all rely on the ability to map a user's information need and a large collection of documents into representations that make relevant information easy to find. Embedding models, also known in retrieval as dense retrievers or bi-encoders, are central to this shift: they encode queries and documents as vectors, turning retrieval into nearest neighbor search in a shared representation space. Compared with traditional lexical matching methods such as BM25, embedding models promise better semantic matching, robustness to vocabulary mismatch, and stronger integration with downstream language models.

However, the progress of embedding-based information access has been uneven across languages. Much of the early success of dense retrieval was demonstrated in English, where large training collections, pretrained models, and standard benchmarks were readily available. For many other languages, especially low- and medium-resource languages, the research infrastructure has been far more limited. High-quality retrieval datasets are scarce, training practices are less well understood, and model behavior is often evaluated only through aggregate scores rather than through analysis of how multilingual representations are formed and transferred. This gap matters because search is a basic component of knowledge access. If multilingual retrieval models are weak, poorly evaluated, or poorly understood, then speakers of many languages receive less benefit from advances in information access and from systems that depend on retrieval.

This thesis studies multilingual embedding models for retrieval from three connected perspectives. First, it asks what data and benchmarks are needed to train and evaluate retrieval models in diverse languages. Second, it asks how multilingual embedding mod-

els should be trained under realistic resource conditions, including settings where target-language data or target-language pretrained models are limited. Third, it asks why cross-lingual transfer happens at all: what aspects of token overlap and shared semantic structure allow a model trained in one language to work in another? Together, these questions move from infrastructure, to modeling practice, to model understanding.

1.1 Overview

The first part of this thesis focuses on data. Dense retrieval requires supervision in the form of query-passage relevance judgments, but creating such data in many languages is expensive and methodologically delicate. The datasets must contain natural queries, natural documents, reliable human labels, enough examples to train neural models, and enough language diversity to support meaningful cross-lingual analysis. This thesis presents two multilingual retrieval resources, Mr. TYDI and MIRACL. Mr. TYDI establishes a large-scale monolingual retrieval benchmark over eleven typologically diverse languages. MIRACL expands this foundation to eighteen languages and substantially richer human annotations, providing a stronger basis for both model training and evaluation.

The second part of this thesis focuses on training. Once multilingual retrieval data are available, the design space for training embedding models becomes much larger than in the monolingual English setting. A practitioner may have a pretrained multilingual model but no target-language training data, target-language data but no strong pretrained model, or data and models in several related and unrelated languages. This thesis studies these scenarios systematically. It compares model backbones, training stages, translated and in-language data, cross-lingual transfer, knowledge distillation, and the role of monolingual versus multilingual pretrained transformers. The goal is not only to improve effectiveness, but also to identify practical training strategies for building multilingual dense retrievers under different resource conditions.

The third part of this thesis focuses on understanding. The training results show that multilingual embedding models can transfer relevance matching behavior across languages, sometimes even between languages with different scripts. This raises a deeper question: what information is being shared? This thesis investigates three possible sources of transfer. It first analyzes shared words and subwords that naturally occur across multilingual corpora and tokenizer vocabularies, then examines whether multilingual language models can preserve much of their downstream effectiveness when many subwords are grouped into shared token-level semantic concepts. These analyses suggest that multilingual transfer is

supported not only by model scale or training data size, but also by concrete lexical and semantic structures inside multilingual vocabularies and embeddings.

1.2 Organization

Chapter 2 provides background on the retrieval and representation-learning methods used throughout the thesis. It introduces dense retrieval with bi-encoder models, contrastive learning objectives, multilingual pretrained transformers, and the terminology used to describe model backbones, training stages, and cross-lingual transfer. This background establishes the technical setting for the dataset, training, and analysis chapters that follow.

Chapter 3 presents two multilingual datasets, Mr. TYDI and MIRACL, designed for monolingual retrieval beyond English. The chapter explains why existing resources were insufficient for training and evaluating dense retrievers, how the datasets were constructed, and how their language coverage and annotation density support both benchmarking and further research on training and understanding models' multilingual capabilities.

Chapter 4 studies training practices for multilingual embedding models. It begins from the observation that zero-shot multilingual dense retrievers can underperform strong lexical baselines, then examines how to improve them across several resource scenarios. The chapter compares choices of pretrained backbones, training data, training stages, and transfer settings, with the aim of deriving practical guidance for building effective multilingual dense retrieval models.

Chapter 5 turns to understanding multilingual representations through controlled modifications of multilingual data, token embeddings and vocabularies. The chapter connects empirical transfer behavior to properties of multilingual corpora, tokenizer vocabularies, and embedding spaces, offering a more detailed view of how multilingual language models may represent and share meaning across languages. Although the modifications on data and models are primarily used as analytical interventions, the results also reveal practical opportunities for vocabulary compression and post-hoc representation enhancement.

Chapter 2

Background

This chapter provides background on the fundamental concepts and backgrounds for embedding models, the task of ad hoc retrieval, which would be the main focus of this thesis. We will also briefly introduce the other embedding-based NLP tasks that we evaluated as part in Chapter 5.

2.1 Ad-hoc Retrieval

Ad hoc retrieval is a task that returns a ranked list of documents from a collection of documents according to their relevance to a user-given query. More formally, given a corpus C and a query q, the retrieval model returns an ordered list of the top-k documents from C ($k \ll |C|$), with the goal of maximizing retrieval quality as reflected by metrics such as MRR, nDCG, or recall. Following the Cranfield evaluation paradigm, a standard retrieval benchmark consists of a document collection, a set of queries, and relevance judgments that assess whether each judged query-document pair is relevant, either with binary or graded labels [21, 128]. The ranked list produced by a system is then compared against these judgments to compute downstream retrieval effectiveness.

2.1.1 Bag-of-Words Algorithms

Prior to embedding models, ad hoc retrieval was typically done using bag-of-words retrieval methods that score documents based on the statistics of lexical overlap between the query

and the document. Okapi BM25 [110] is a classic example of such methods, which still served as a strong baseline nowadays for monolingual retrieval:

$$\text{score}(q,d) = \sum_{i=1}^{|q|} idf(q_i) \cdot \frac{tf(q_i,d) \cdot (k_1+1)}{tf(q_i,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{2.1}$$

$$idf(q_i) = \ln(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1) \tag{2.2}$$

where $tf(q_i, d)$ denote the term frequency of q_i in d, $idf(q_i)$ denotes the inverse document frequency of q_i with smoothing and non-negative normalization, k_1 and b are configurable parameters,[1] $|d|$ denotes the length of document d, $avgdl$ denotes the average document length in the collection, $n(q_i)$ denotes the number of documents containing q_i, and N denotes the total number of documents in the collection.

2.1.2 Embedding Models

Embedding models learn to represent text (or inputs in other modalities) as embedding vectors in a continuous space such that the similarity between the vectors reflects the similarity between the inputs. In standard settings, the embedding models are trained under InfoNCE loss [97], a variant of the contrastive learning objective that learn representations by predicting which future segment x_{t+k} belongs to given context c_t:

$$L_N = -E_X[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}] \tag{2.3}$$

where $X = \{\text{one positive sample from } p(x_{t+k}|c_t), N-1 \text{ negative samples from the negative sampling distribution } p(x_{t+k})\}$. It has been proven that minimizing the InfoNCE loss is equivalent to maximizing the mutual information between c_t and x_{t+k} [97].

The loss function was firstly introduced to accomplish information retrieval by DPR [58], with the goal to identify the relevant documents from the candidate negative documents:

$$L_N = -E_{q,d} \log(\frac{\exp(s_{q,d^+})}{\exp(s_{q,d^+}) + \sum_{d_j \in \mathcal{D}} \exp(s_{q,d_j^-})}) \tag{2.4}$$

where q is the question representation, d^+ is the representation of a positive (relevant) document, d_j^- is the representation of a negative (non-relevant) document, $s_{q,d}$ is the similarity

[1] In this work, $k_1 = 0.9$ and $b = 0.4$ if not otherwise specified.

score computed based on the query and document representations, (e.g., cosine similarity or inner product), and $\mathcal{D} = \{$one positive sample from $p(d_j|q)$, $N-1$ negative samples from the negative sampling distribution $p(d_j)\}$.

DPR encoders were originally designed to be initialized with English BERT [58]. for retrieval in multiple languages and to exploit cross-lingual transfer, we instead initialize the model with multilingual pretrained transformers (e.g., mBERT, XLM-R_{Large}, mContriever, etc.), keeping all other aspects of the training identical. This would be referred to as "mDPR" in the rest of this thesis.

Note that throughout this paper we refer to mDPR generically as a shorthand for multilingual dense retrieval models based on a bi-encoder architecture, even though we explore additional techniques that were not present in the original paper, e.g., hard negative mining. Following standard parlance, we refer to the model used to initialize the training as the "backbone" (typically from a publicly available model checkpoint). For example, mDPR can be built on an mBERT backbone, but other alternatives include XLM-R_{Large} and mContriever. Indeed, we experiment with a number of different backbones in this work to explore their impact on effectiveness.

2.1.3 Cross-Encoder Models

Cross-encoder models refer to language models that encode both the query and the document as a single input, and then compute the relevance score between them [93, 144]. Given concatenated query q and document d, the model computes relevance scores $s_{q,d}$ from the representation of a special token (e.g., [CLS]) on the final layer of a language model E, formulated as follows [144]:

$$s_m(q,d) = E(q,d)W + b, \tag{2.5}$$

where $E(q,d) \in \mathbf{R}^D$ is the concatenated representation of the query and the document, and $W \in \mathbf{R}^{D\times 1}$ and $b \in \mathbf{R}$ are the weight and bias for classification. Given its computational expense, cross-encoder models are typically used as a reranking step after the retrieval stage.

2.2 Multilingual Retrieval

The term "multilingual retrieval" refers to the task of retrieving relevant documents from a collection of documents in more than one language, while in practice, it can be broken down

into several sub-tasks based on the language(s) of the queries and documents: *monolingual retrieval*, *cross-lingual retrieval*, and *multilingual retrieval*. Given a query in the language (L_1), monolingual retrieval aims to find relevant information from the same language as the query (L_1), cross-lingual retrieval finds relevant information from a different language (L_2), and multilingual retrieval finds relevant information from multiple different languages ($L_i, i \in \{1, \ldots, n\}$). [50] The focus on this thesis is specifically on monolingual retrieval, where the query and the documents are in the same language.

2.2.1 Prior to Embedding-based Retrievers

Before embedding models, retrieval across languages was largely studied as cross-lingual information retrieval, where bag-of-words methods naturally struggled with vocabulary mismatch between query and document languages. Mainstream solutions tackle the challenge explicitly by translating queries or documents with bilingual dictionaries, parallel or comparable corpora, and machine translation systems, often combined with query expansion, term disambiguation, and language-specific indexing before applying standard ranking methods [11, 38, 92].

2.2.2 Embedding-based Retrievers

Multilingual Embedding Models Embedding-based multilingual retrievers were first adapted from English dense retrieval models by replacing the encoder backbone with multilingual pretrained language models such as mBERT and XLM-R_{Large}, while also evaluated on cross-lingual retrieval tasks [9, 78]. mContriever extends this line by improving the backbone with contrastive pretraining on unsupervised synthetic data, making it a stronger multilingual backbone for retrieval on both zero-shot and finetuned settings [51].

More recently, works start to consider multilingual support as a general feature of the embedding model itself. E5 uses weakly supervised contrastive pretraining over large text-pair collections, and multilingual E5 extends the same recipe with multilingual text pairs and supervised finetuning data, including retrieval data from MIRACL and Mr. TYDI [130]. Commercial embedding APIs also follow this direction: Cohere's multilingual embedding models support retrieval-style query and document encodings across more than 100 languages [22].

After large language models (LLMs) have shown their effectiveness on generative tasks, embedding-based retrievers increasingly use LLMs as backbones or teachers during the

embedding training process. One common design is to reuse large decoder-only models as embedding backbones and derive a fixed-length representation from the final token [86], and later with more finetuning on multilingual data: EmbeddingGemma adapts the Gemma family into an encoder-style embedding model, while Qwen3-Embedding trains embedding and reranking models on top of Qwen3 with multilingual synthetic data and multi-stage finetuning [115, 153]. Knowledge distillation from stronger embedding models is often applied: Stella and Jina Embeddings v5 both use Qwen-based backbones with distillation, while Jina further organizes the resulting model family around task-specific adapters [147, 2]. Other strategies includes task prompts, long-context training, Matryoshka representation learning, or deployment-oriented quantization. [148, 129, 99]

2.3 Beyond Retrieval

2.3.1 MLM pretraining

Chapters 4 and 5 involve experiments where encoder language models are additionally pretrained on the target language corpus. Masked language modeling (MLM) is the standard self-supervised pretraining objective for encoder-only models such as BERT [31]. Given an input sequence, a subset of tokens is replaced by a special [MASK] token or otherwise corrupted, and the model is trained to recover the original tokens from their surrounding context. Because the encoder can attend bidirectionally to both the left and right context, MLM encourages contextual token representations that are useful for downstream understanding tasks. In this thesis, additional MLM pretraining refers to continuing this objective on text from a target language before finetuning or evaluating the model on downstream tasks.

2.3.2 Natural Language Understanding Tasks

Traditionally, many natural language understanding tasks are based on semantic representation of the text, and typically perform training and evaluation on classification tasks following a similar general recipe: encode an input sentence or sentence pair, then attach a task-specific classification head; and update the encoder and classifier head through supervised finetuning.

Named entity recognition (NER) is a token-level sequence labeling task that identifies spans corresponding to entities such as people, locations, and organizations. For encoder

models, each input token receives a contextual representation, and a classification layer predicts an entity tag for each token. Multilingual NER datasets such as MasakhaNER test whether token-level representations transfer to languages with different scripts, morphology, and data availability [1].

Natural language inference (NLI) is a sentence-pair classification task: given a premise and a hypothesis, the model predicts whether the hypothesis is entailed by, contradicts, or is neutral with respect to the premise. Multilingual NLI benchmarks such as XNLI evaluate whether multilingual encoders learn sentence representations that transfer across languages [25]. The typical finetuning process encodes the premise and hypothesis together with a classifier head, together trained on the sequence-level representation.

Extractive question answering (QA) is a reading comprehension task that requires selecting a span of text from a provided context as the answer to a question. For encoder models, the question and context are encoded together, and the model predicts the start and end positions of the answer span over the context tokens. Multilingual QA benchmarks such as TYDI evaluate whether models can perform this span selection process across typologically diverse languages [20].

2.3.3 Impact of Cross-Lingual Shared Tokens

Chapter 5 examines the impact of *incidental* multilingual text (IMT) tokens on cross-lingual transfer in retrieval. Related to this topic, many works have studied the impact of shared tokens with inconsistent results. It has been reported that more shared tokens lead to better cross-lingual transfer results on the NER task [103] and other NLP tasks, including POS and dependency parsing [138]. On the other hand, it has also been found that shared tokens do not play a significant role in cross-lingual transfer in NLI and NER [57] and that multilingual language models could be pretrained without shared vocabulary [7]. All of the above works are conducted on multilingual pLM pretraining or NLP tasks. On retrieval, it has been found that code-switching data could assist in cross-lingual retrieval, yet it remains unclear how shared tokens might influence language transfer in monolingual retrieval. [77] Closely related to our work, Do et al. [32] propose to enhance cross-lingual transfer in monolingual retrieval by leveraging *manual* code-mixing, while we focus on examining the impact of *naturally occurring* IMT tokens in retrieval corpora.

2.3.4 Token-Level Semantic Concepts

Chapter 5 also examines language models' understanding of token-level semantic concepts, which is related to the following related topics:

Semantic Similarities in Language Models The semantics latent knowledge in word embeddings have been leveraged to adapt the pretrained multilingual LMs to unseen languages or scripts [102, 132, 80]. While it has been shown that the shared semantics assist the language transfer, it is unclear how much the semantics alone could achieve compared to the full-fledged models. Another line of works focuses on the isomorphic representations of *contextualized* word embeddings. It has been found that vision and language models shows isomorphic representations [66], and similarly, among the contextualized representations of multilingual large language models [98] Two additional works also study contextualized concepts encoded in language models. One analyzes how latent concepts are encoded in representations and how they align with human-defined concepts [113]. The other analyzes how large language models encode the TypeOf relations of concepts, and proposes a model-agnostic, proof-of-concept method to shift the model to a concept-level understanding [116].

Bilingual Lexicon Induction and Word Alignment The Cross-lingual Subword Alignment (CLSA) operation to be introduced in Chapter 5 is related to the task of *Bilingual Lexicon Induction (BLI)* and *Word Alignment*. BLI aims to induce the equivalent translation in language L_2 given a word in language L_1 [6, 24, 133, 118]. CLSA is similar to BLI in terms of focusing on uncontextualized cross-lingual words pairs, but different in that it aims to align the word pairs in the embedding space and targets on not only bilingual but also multilingual words. Word alignment aims to find bilingual word pairs in parallel sentences [17, 53]. While names are similar, the goals are different: In addition to above differences with BLI, CLSA focuses on uncontextualized cross-lingual word pairs and does not involve parallel sentences.

Word Sense Clustering The Semantic Grouping (SG) operation to be introduced in Chapter 5 is related to the task of word sense clustering. Although both share similar objectives, word sense clustering primarily relies on corpus statistics [120], whereas the SG operation is based on the similarity of word embeddings.

Chapter 3

Dataset and Benchmarks

Reliable evaluation is a prerequisite for progress in any empirical research area. For multilingual retrieval, this requirement is especially important as aggregate results can hide large differences across languages, scripts, domains, and annotation practices. Without carefully constructed benchmarks, it is difficult to know whether a model truly retrieves relevant information on a diverse set of languages or only to a narrow set of high-resource languages. The need for training data is even stronger in the era of transformer-based retrieval models. Dense retrievers are trained from query-passage relevance judgments, and their effectiveness depends heavily on both the quantity and quality of the labeled examples used for training. This created a bottleneck for non-English languages where large-scale high-quality retrieval annotations were not readily available.

This chapter presents two datasets constructed to support training and evaluation for monolingual retrieval beyond English. The first, Mr. TYDI, provides a multilingual retrieval benchmark over eleven typologically diverse languages and establishes a foundation for studying dense retrievers outside the English setting. The second, MIRACL, expands the language coverage to eighteen languages, provides substantially richer human relevance annotations and rectify the passage distribution in Mr. TYDI. Together, these resources supply both the supervision needed to train multilingual embedding models and the benchmarks needed to evaluate them reliably across diverse languages. Table 3.1 summarizes how these datasets compare with several existing multilingual retrieval and question answering resources.

Dataset Name	Natural Queries	Natural Passages	Human Labels	# Lang	Avg # Q	Avg # Labels/Q	Total # Labels	Training?
NeuCLIR [64]	✓	✓	✓	3	160	32.74	5.2k	×
MKQA [82]	×	✓	✓	26	10k	1.35	14k	×
mMARCO [13]	×	×	✓	13	808k	0.66	533k	✓
CLIRMatrix [121]	×	✓	×	139	352k	693	34B	✓
Mr. TYDI [149]	✓	✓	✓	11	6.3k	1.02	71k	✓
MIRACL	✓	✓	✓	18	4.3k	9.23	726k	✓

Table 3.1: Comparison of select multilingual retrieval datasets. **Natural Queries** and **Natural Passages** indicate whether the queries and passages are "natural", i.e., generated by native speakers (vs. human- or machine-translated), and for queries, in natural language (vs. keywords or entities); **Human Labels** indicates human-generated labels (vs. synthetically generated labels); # **Lang** is the number of languages supported; **Avg # Q** is the average number of queries for each language; **Avg # Labels/Q** is the average number of labels provided per query; **Total # Labels** is the total number of human labels (both positive and negative) across all languages (including synthetic labels in CLIRMatrix). **Training?** indicates whether the dataset provides sufficient data for training neural models.

3.1 Mr. TyDi

Retrieval approaches based on learned dense representations, typically derived from transformers, form an exciting new research direction that has received much attention of late. These dense retrieval techniques generally adopt a supervised approach to representation learning, where a labeled dataset is used to train two encoders—one for the queries and the other for texts from the corpus to be retrieved—whose output representation vectors are then compared with a simple comparison function such as inner product. Retrieval against a large text corpus is typically formulated as nearest neighbor search and efficiently executed using off-the-shelf libraries.

Like all methods based on supervised machine learning, the effectiveness of the trained models on "out of distribution" (OOD) samples is an important issue since it concerns model robustness and generalizability. For dense retrieval, training data typically comprise (query, relevant passage) pairs, and in this context, OOD could mean that (1) the passage encoder is fed text from a different domain, genre, register, etc. than the training data, (2) the query encoder is fed queries that are different from the training queries, (3) the relationship between the inputs at inference time is different from the training samples (e.g., task variations), or (4) a combination of all of the above.

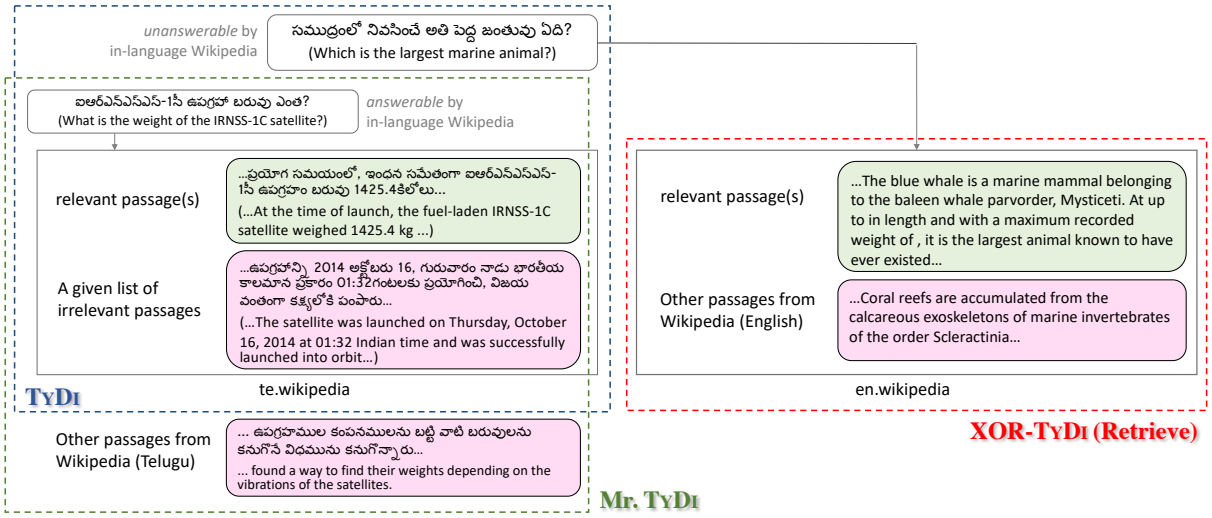


Figure 3.1: Comparison between TYDI, XOR-TYDI, and Mr. TYDI with an example in Telugu. The green blocks indicate relevant passages and the red blocks indicate non-relevant passages.

It is, in fact, already known that dense retrieval techniques generalize poorly across different corpora, queries, tasks, etc. In early 2021, BEIR [124] evaluate the zero-shot transfer capabilities of dense retrieval models by creating a framework that unifies over a dozen retrieval datasets spanning diverse domains. In a zero-shot setting, the authors found that BM25 remained the most effective overall. That is, dense retrieval techniques trained on one dataset can spectacularly fail on another dataset—exactly the out-of-distribution challenges we discussed above. In contrast, BM25, "just works" regardless of the corpus and queries, even though on "in distribution" samples, dense retrieval models are unequivocally more effective. Thus, learned dense representations are not as general and robust as BM25 representations.

In the work of Mr. TYDI, we focus on another aspect of the generalizability and robustness of learned representations for ranking: What if the encoders are applied to texts in languages different from the one they are trained in? Our focus is on *mono-lingual* retrieval in non-English languages (e.g., Bengali queries against Bengali documents) rather than *cross-lingual* retrieval, where documents and queries are in different languages (e.g., English queries against Arabic documents).

3.1.1 Background

In presenting a new benchmark dataset, one important question to answer is: Why is a new resource needed? We begin by addressing this question. The introduction already lays out the intellectual motivation for our work. Thus, we focus here on explaining why existing datasets are not sufficient. The answer is summarized in Figure 3.1.

Mr. TYDI is constructed from TYDI [20], a question answering dataset covering eleven typologically diverse languages. For each language, the creators provided annotators with a prompt (the first 100 characters of a Wikipedia article), who were asked to write a question that cannot answered by the snippet. Then, for each question, annotators were given the top Wikipedia article returned by Google search and asked to label the relevance of each passage in the article as well as to identify a minimal answer span (if possible). Given this procedure, the answer to the question may or may not be found in the passages from the selected article. The answer passages are always in the same language as the questions. Note that the questions in different languages are not comparable as they are created independently rather than through translation.

The weakness of TYDI from our perspective is that it is essentially a machine reading comprehension task like SQuAD [109] because candidate passages are all included as part of the dataset (i.e., the passages are from the top Wikipedia article returned by Google search). Instead, we need a resource akin to what QA researchers call the "open-domain" or "open-retrieval" task, where the problem involves retrieval from a much larger corpus (e.g., all of Wikipedia) [19]. Thus, at a high level, Mr. TYDI can be viewed as an open-retrieval extension to TYDI.

Related to this work, XOR-TYDI [8] is a cross-lingual QA dataset built on TYDI by annotating answers in English Wikipedia for questions TYDI considered unanswerable in the original source (non-English) language. This was accomplished by randomly sampling 5,000 unanswerable (non-English) questions from TYDI, and then searching English Wikipedia articles for answers. Specifically, each non-English question was first translated into English; then, annotators were given the top-ranked English Wikipedia articles and asked to label passages containing the answer.

The XOR-TYDI dataset contains three overlapping tasks, but all of them are focused on the cross-lingual condition. Among the three tasks (XOR-ENGLISHSPAN, XOR-RETRIEVE, and XOR-FULL), XOR-RETRIEVE is most comparable to our work, but the retrieval target for the task is explicitly English Wikipedia articles rather than Wikipedia articles in the question's language. While the XOR-FULL task requires systems to select answer spans from both English and target-language Wikipedia articles, the dataset does not provide ground truth for the intermediate retrieval step, thus it cannot be used for our evaluations.

The annotations of XOR-TYDI do not allow us to examine mono-lingual retrieval in non-English languages because the creators started with "unanswerable" non-English TYDI questions. Furthermore, since all answer passage annotations were performed on English Wikipedia, this doesn't help if we are interested in, for example, searching Finnish Wikipedia with Finnish questions.

Another point of comparison worth discussing is MKQA [82], which comprises 10k question–answer pairs aligned across 26 typologically diverse languages. Questions are paired with exact answers in the different languages, and evaluation is conducted in the open-retrieval setting by matching those answers in retrieved text. There are two main differences between our work and MKQA: First, MKQA was created via translation in order to achieve cross-lingual alignment. In addition to the possible translation artifacts that such a process might introduce, which TYDI [20] discussed at length, we argue that forced alignment creates non-natural questions, for the simple reason that speakers of different languages are likely to be interested in different topics. This is different from the "geographically dependent" questions that MKQA tries to avoid. Take the question "*who starred in the movie bridge over the river kwai*" as an example: While it does not involve any geographical preference, the question is probably less likely to be asked in, say, Swahili, compared to in English or Thai. Second, the builders of MKQA explicitly made the decision to create "retrieval-independent answer annotations" that are linked to Wikidata entities and a few other value types. This decision, we feel, restricts the range of natural language questions that are covered. The cross-lingual aspect of the dataset appears to be primarily limited to entity translations, which likely do not cover a wide range of linguistic phenomena (which is the reason that we are interested in typologically diverse languages to begin with). Thus, we believe that Mr. TYDI fills a gap in the evaluation space that is currently not occupied.

Many multi-lingual (both mono-lingual and cross-lingual) information retrieval and question answering datasets have been constructed over the past decades, via community-wide evaluations at TREC, FIRE, CLEF, and NCTIR. These test collections are typically built on newswire articles, although some evaluations use Wikipedia and scientific texts. While no doubt useful for evaluation, these test collections usually comprise only a small number of queries (at most a few dozen) with relevance judgments, which are insufficient to fine-tune dense retrieval models. Furthermore, whereas TYDI at least draws from comparable corpora (i.e., Wikipedia articles), these test collections are built on corpora from much more diverse sources. This makes it difficult to generalize across different languages. For these reasons, the above-mentioned IR and QA test collections are not suitable for tackling the research questions we are interested in.

3.1.2 Dataset Construction

Having justified the need for a new benchmark dataset, this section describes the construction of Mr. TYDI, which can be best described as an open-retrieval extension to TYDI.

Corpus The formulation of any text ranking problem begins with a corpus $C = \{d_i\}$ comprising the units of text to be retrieved. As the starting point, we used exactly the same raw Wikipedia dumps as TYDI.

Relevance annotations in TYDI are provided at the passage level (in the passage selection task), and thus we kept the same level of granularity in our corpus preparation. For articles covered by TYDI (identified by the article titles), we retained the original passages. For articles that are not covered by TYDI, we prepared passages using WikiExtractor[1] based on natural discourse units (e.g., two consecutive newlines in the wiki markup). Unfortunately, TYDI [20] did not precisely document their passage segmentation method, but based on manual examination of our results, the generated passages appear qualitatively similar to the TYDI passages.

The result of the corpus preparation process is, for each language, a collection of passages from the Wikipedia articles in that language. To form the final passages that comprise the basic unit of retrieval, we prepend the title of the Wikipedia article to each passage. This creates retrieval units that can be more readily understood in isolation.[2]

Task While Mr. TYDI is adapted from a QA dataset, our task is mono-lingual *ad hoc* retrieval. That is, given a question in language L, the task is to retrieve a ranked list of passages from C_L, the Wikipedia collection in the same language (prepared in the manner described above), where the retrieved passages are ranked according to their relevance to the given question.

Our assumption here is a standard "retriever–reader" framework [19] or a multi-stage ranking architecture [73], where we focus on the retriever (what IR researchers call candidate generation or first-stage retrieval). For end-to-end question answering, the output of the retriever would be fed to a reader for answer extraction. This focus on retrieval allows us to explore the research questions outlined in the introduction, and this formulation is consistent with previous work in dense retrieval [58].

[1] https://github.com/attardi/wikiextractor

[2] Mr. TyDi v1.1 contains these article titles, whereas v1.0 did not. Results reported in this paper are with v1.1; for differences, please refer to the earlier version of our paper posted on arXiv.

Questions and Judgments To prepare the questions, we started with all questions provided by TYDI and removed those without any answer passages or whose answer passages are all empty. We consider all non-empty annotated answer passages from TYDI as relevant to the corresponding question in Mr. TYDI. We adopt the development set of TYDI as our test set, since the original test data are not public. A new development set was created by randomly sampling 20% of questions from the original training set. We observed that some of the questions in TYDI are shared between the training and development set (but labeled with different answer passages). In these cases, we retained the duplicate questions only in the training set.

Descriptive statistics for Mr. TYDI are shown in Table 3.2, where languages are identified by their two letter ISO-639 language codes.

In summary, relevant passages in Mr. TYDI are imputed from TYDI. Since TYDI [20] only asked annotators to assess the top-ranked article for each question, there are likely relevant passages that have not been identified. Following standard assumptions in information retrieval, unjudged passages are considered non-relevant. Thus, it is likely that ranking models will retrieve false negatives, i.e., passages that are relevant, but would not be properly rewarded.

In other words, our judgments are far from exhaustive. This might be a cause for concern, but is a generally accepted practice in IR research due to the challenges of gathering complete judgments. The widely used MS MARCO datasets [10], for example, share this characteristic of having "sparse judgments". No claim is made about the exhaustiveness of the annotations, as both Mr. TYDI and MS MARCO provide only about one good answer per question. From a methodological perspective, findings based on MS MARCO "sparse judgments" are largely consistent with results from more expensive evaluation efforts (to gather more complete judgments), such as the TREC Deep Learning Tracks [29, 28]. We expect a similar parallel here: more exhaustive judgments will change the absolute scores, but will likely not affect the findings qualitatively.

3.1.3 Baselines

We provide a few baselines on Mr. TYDI as a starting point: BM25, mDPR, and the sparse–dense hybrid.

BM25 We report results with bag-of-words BM25 (Eq. 2.1) with the implementation provided by Pyserini [72], which is built on the open-source Lucene search library. Lucene

		Train		Dev		Test		Corpus Size
		# Q	# J	# Q	# J	# Q	# J	
Arabic	(ar)	12,377	12,377	3,115	3,115	1,081	1,257	2,106,586
Bengali	(bn)	1,713	1,719	440	443	111	130	304,059
English	(en)	3,547	3,547	878	878	744	935	32,907,100
Finnish	(fi)	6,561	6,561	1,738	1,738	1,254	1,451	1,908,757
Indonesian	(id)	4,902	4,902	1,224	1,224	829	961	1,469,399
Japanese	(ja)	3,697	3,697	928	928	720	923	7,000,027
Korean	(ko)	1,295	1,317	303	307	421	492	1,496,126
Russian	(ru)	5,366	5,366	1,375	1,375	995	1,168	9,597,504
Swahili	(sw)	2,072	2,401	526	623	670	743	136,689
Telugu	(te)	3,880	3,880	983	983	646	677	548,224
Thai	(th)	3,319	3,360	807	817	1,190	1,368	568,855
Total		48,729	49,127	12,317	12,431	8,661	10,092	58,043,326

Table 3.2: Descriptive statistics for Mr. TYDI: the number of questions (# Q), judgments (# J), and the number of passages (Corpus Size) in each language.

provides language-specific analyzers for nine of the eleven languages in Mr. TYDI; for these languages, we used the Lucene implementations. For Telugu (te) and Swahili (sw), since Lucene does not provide any language-specific implementations, we simply used its whitespace analyzer. We report BM25 scores on two conditions, with default and tuned k_1 and b parameters; the default settings are $k_1 = 0.9, b = 0.4$. Tuning was performed on the development set, on a per-language basis, via grid search on $k_1 \in [0.1, 1.6]$ and $b \in [0.1, 1.0]$, with step size 0.1, optimizing MRR@100.

mDPR Details of mDPR has been introduced in Section 2.1.2 and 2.2.2. Specifically in this work, we used mDPR based on mBERT [31] as the backbone, indexed using FlatIP indexes from Faiss [54] library for nearest neighbor search. Experiments were conducted using the same codebase as previous DPR replication experiments [85] on Pyserini toolkit [72]. We choice of zero-shot mDPR as a baseline to evaluate its generalizability to dense retrieval techniques in out-of-distribution settings (in this case, primarily different languages), which also makes the comparison "fair" to BM25, which is similarly not provided any labeled data from the target task.

Sparse–Dense Hybrid The hybrid baseline combines the scores of sparse (BM25) and dense (mDPR) retrieval results. The final fusion score of each document is calculated by

	Ar	Bn	En	Fi	Id	Ja	Ko	Ru	Sw	Te	Th	Avg
BM25 (default)	0.368	0.418	0.140	0.284	0.376	0.211	0.285	0.313	0.389	0.343	0.401	0.321
BM25 (tuned)	0.367	0.413	0.151	0.288	0.382	0.217	0.281	0.329	0.396	0.424	0.417	0.333
mDPR	0.291	0.291	0.291	0.206	0.271	0.213	0.235	0.283	0.189	0.111	0.172	0.226
hybrid	0.500†	0.555†	0.328†	0.377†	0.481†	0.360†	0.361†	0.455†	0.415	0.418	0.507†	**0.426**

(a) MRR@100

	Ar	Bn	En	Fi	Id	Ja	Ko	Ru	Sw	Te	Th	Avg
BM25 (default)	0.793	0.869	0.537	0.719	0.843	0.645	0.619	0.648	0.764	0.758	0.853	0.732
BM25 (tuned)	0.800	0.874	0.551	0.725	0.846	0.656	0.797	0.660	0.764	0.813	0.853	0.758
mDPR	0.650	0.779	0.678	0.568	0.685	0.584	0.533	0.647	0.528	0.366	0.515	0.594
hybrid	0.871†	0.946†	0.793†	0.827†	0.900†	0.794†	0.718†	0.815†	0.808†	0.823	0.883†	**0.834**

(b) Recall@100

Table 3.3: Results of BM25 (with default and tuned parameters), mDPR, and the sparse–dense hybrid on the test set of Mr. TYDI. The symbol † indicates significant improvements over BM25 (tuned) (paired t-test, $p < 0.01$).

$s_{\text{sparse}} + \alpha \cdot s_{\text{dense}}$, where s_{sparse} and s_{dense} represent the scores from sparse and dense retrieval, respectively. [85] We take 1000 hits from mDPR and 1000 hits from BM25 and normalize the scores from each into $[0, 1]$ since the range of the two types of scores otherwise are quite different. If one hit isn't found in the other, the normalized score for that hit is set to zero. The weight α was tuned in $[0, 1]$ with a simple line search on the development set by optimizing MRR@100 with step size 0.01.

3.1.4 Experiment Results

We performed experiments on Mr. TYDI v1.1, where each passage contains the title of the Wikipedia article and the passage text. Table 3.3 reports results on the test set across all eleven languages; mean reciprocal rank (MRR) in the top table and recall in the bottom table, both at a cutoff of 100 hits; the final column reports the average across all languages. The rows report BM25 results (default and tuned), followed by results of mDPR and the sparse–dense hybrid. For the hybrid method, statistically significant improvements over tuned BM25 are denoted with the symbol † based on paired t-tests ($p < 0.01$).

By comparing scores in each column, we observe that the absolute effectiveness of the techniques varies greatly across languages. Absolute scores are difficult to compare as both questions and underlying corpora are different. However, three high-level findings emerge:

First, we find that tuning BM25 parameters yields at most minor improvements for most languages, both in terms of MRR@100 and recall except for Telugu (cases where scores decrease slightly can be explained by noise in the training/test splits). This is a bit of a surprise, as parameter tuning usually yields larger overall gains, e.g., in the MS MARCO collections [10]. Regardless, tuned BM25 serves as a competitive baseline for the remainder of our experiments.

Second, we notice that mDPR underperforms BM25 across all languages except for English. That is, in a zero-shot setting, retrieval using learned dense representations from mDPR (fine-tuned with NQ) is a lot worse than just retrieval using BM25-based representations. Clearly, mDPR is far less robust in cross-lingual generalizations. Even within the same language, mDPR seems to be sensitive to characteristics of the training data. Effectiveness on the English portion of Mr. TYDI is only slightly better than BM25, likely arising from the fact that we are applying an NQ-trained model on "out-of-distribution" questions.

Based on the original DPR results [58], we would have expected mDPR to beat BM25 for in-distribution training and inference. Since NQ is also based on Wikipedia, corpus differences are less likely an issue; these results suggest that questions in TYDI and NQ are qualitatively different.

Third, despite the fact that mDPR effectiveness is quite a bit worse than BM25, the MRR@100 of the sparse–dense hybrid is significantly higher than tuned BM25 for nine of the eleven languages (the exceptions are Swahili and Telugu). Rephrased differently, this means that although mDPR by itself is a poor dense retrieval model in a zero-shot setting, it nevertheless contributes valuable relevance signals that are able to improve over tuned BM25. On average, the hybrid results are around eight and five points absolute higher than tuned BM25 in terms of MRR@100 and recall, respectively.

Because absolute scores vary widely across languages, it is helpful to normalize the effectiveness of tuned BM25 to 1.0 and scale the effectiveness of mDPR and the hybrid approach appropriately; this is shown in Figure 3.2 (left) for MRR@100. As an example, from the leftmost bars, we see that the MRR@100 of mDPR in Arabic is 71% of BM25 but the sparse–dense hybrid improves over BM25 by 34% (stat. sig). We additionally plot the relation between the normalized effectiveness of mDPR and the hybrid approach in Figure 3.2 (right).

This plot shows a clear positive (linear) correlation, that is, better mDPR (relative) effectiveness translates into bigger improvements over BM25 in the sparse–dense hybrid. What is surprising, though, is that this relationship seems to hold even if the mDPR results are poor. For example, in Thai, the MRR@100 of mDPR is only 32% of BM25 (tuned),

yet the hybrid yields a statistically significant 18% relative gain in the hybrid approach. However, there appear to be limits to our simple linear combination of relevance signals: for both Swahili and Telugu, the hybrid approach does not outperform tuned BM25, likely because mDPR effectiveness is too poor.

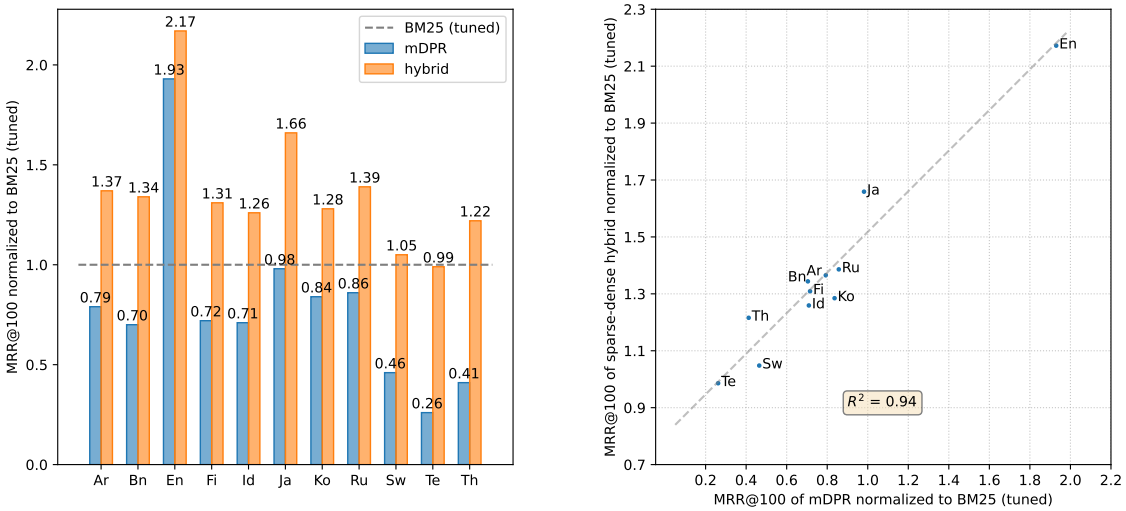


Figure 3.2: MRR@100 of mDPR and the sparse–dense hybrid normalized with respect to BM25 for each language (left); corresponding pairs for each language plotted as a scatter plot (right).

3.2 MIRACL

Built on top Mr. TYDI, MIRACL is a multilingual dataset for *ad hoc* retrieval across 18 languages that collectively encompass over three billion native speakers around the world. In total, we have gathered over 726k high-quality relevance judgments for 78k queries over Wikipedia in these languages, where all annotations have been performed by native speakers hired by our team. MIRACL covers languages that are both typologically close as well as distant from 10 language families and 13 sub-families, associated with varying amounts of publicly available resources. Extensive automatic heuristic verification and manual assessments were performed during the annotation process to control data quality. In total, MIRACL represents an investment of around five person years of human annotator effort. Our goal is to spur research on improving retrieval across a continuum of languages,

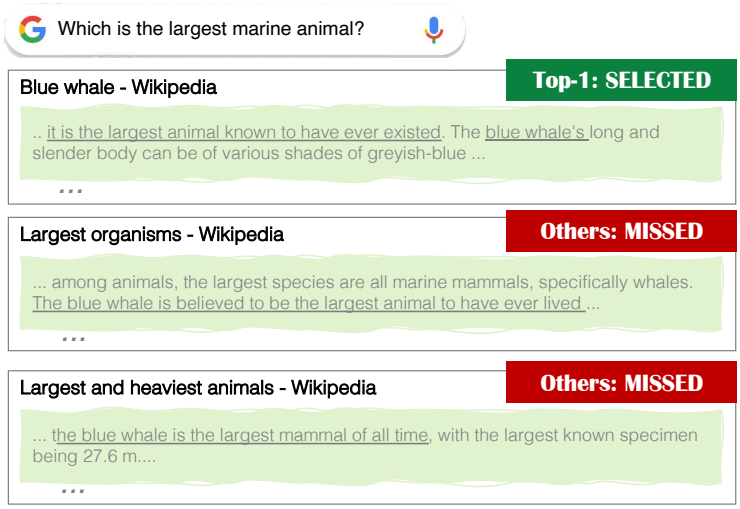


Figure 3.3: Examples of missing relevant passages in TYDI (and thus Mr. TYDI) for the query "*Which is the largest marine animal?*" Since only relevant passages in the top Wikipedia article are included, other relevant passages are missed.

thus enhancing information access capabilities for diverse populations around the world, particularly those that have traditionally been underserved. MIRACL is available at `http://miracl.ai/`. Up to the time of Dec. 2023, the released dataset has been downloaded over 10,863 times (within 1 year).

3.2.1 Comparison to Mr. TyDi

Since MIRACL inherits from Mr. TYDI, it makes sense to discuss important differences between the two. While Mr. TYDI enables the training and evaluation of monolingual retrieval, it has three shortcomings we aimed to address in MIRACL.

Limited Positive Passages. In TYDI, and thus Mr. TYDI, candidate passages for annotation are selected only from the top-ranked Wikipedia article based on a Google search. Consequently, a considerable number of relevant passages that exist in other Wikipedia articles are ignored; Figure 3.3 illustrates an instance of this limitation. In contrast, MIRACL addresses this issue by sourcing candidate passages from all of Wikipedia, ensuring that our relevant passages are diverse.

Moreover, in MIRACL, we went a step further and asked annotators to assess the top-10 candidate passages from an ensemble model per query, resulting in richer annotations compared to those of Mr. TYDI, which were mechanistically generated from TYDI as no new annotations were performed. Furthermore, we believe that explicitly judged negative examples are quite valuable, compared to, for example, implicit negatives in MS MARCO sampled from BM25 results, as recent work has demonstrated the importance of so-called "hard negative" mining [58, 140, 114, 35]. As the candidate passages come from diverse models, they are particularly suitable for feeding various contrastive techniques.

Inconsistent Passages. Since passage-level relevance annotations in Mr. TYDI were derived from TYDI, it retained exactly those same passages. However, as TYDI was not originally designed for retrieval, it did not provide consistent passage segmentation for all Wikipedia articles. Thus, the Mr. TYDI corpora comprised a mix of TYDI passages and custom segments that were heuristically adjusted to "cover" the entire raw Wikipedia dumps. This inconsistent segmentation is a weakness of Mr. TYDI that we rectified in MIRACL by re-segmenting the Wikipedia articles provided by TYDI and re-building the relevance mapping for these passages (more details in Section 3.2.3).

No Typologically Similar Languages. The 11 languages included in TYDI encompass a broad range of linguistic typologies by design, belonging to 11 different sub-families from 9 different families. However, this means that they are all quite distant from one another. In contrast, new languages added to MIRACL include typologically similar languages. The inclusion of these languages is crucial because their presence can better foster research on cross-lingual comparisons. For example, cross-lingual transfer can be more effectively studied when more similar languages are present. We explore this issue further in Section 3.2.5.

Summary. Overall, not only is MIRACL an order of magnitude larger than Mr. TYDI, as shown in Table 3.1, but MIRACL goes beyond simply scaling up the dataset to correcting many shortcomings observed in Mr. TYDI. The result is a larger, higher-quality, and richer dataset to support monolingual retrieval in diverse languages.

3.2.2 Overview

MIRACL is a multilingual retrieval dataset that spans 18 different languages, focusing on the monolingual retrieval task. In total, we have gathered over 726k manual relevance

judgments (i.e., query–passage pairs) for 78k queries across Wikipedia in these languages, where all assessments have been performed by native speakers. There are sufficient examples in MIRACL to train and evaluate neural retrieval models. Detailed statistics for MIRACL are shown in Table 3.4.

Among the 18 languages in MIRACL, 11 are existing languages that are originally covered by Mr. TYDI, where we take advantage of the Mr. TYDI queries as a starting point. Resources for the other 7 new languages are created from scratch. We generated new queries for all languages, but since we inherited Mr. TYDI queries, fewer new queries were generated for the 11 existing languages as these languages already had sizable training and development sets. For all languages, we provide far richer annotations with respect to Wikipedia passages. That is, compared to Mr. TYDI, where each query has on average only a single positive (relevant) passage [149], MIRACL provides far more positive as well as negative labels for all queries. Here, we provide an overview of the data splits; details of fold construction will be introduced in Section 3.2.3.

- **Training and development sets:** For the 11 existing Mr. TYDI languages, the training (development) sets comprise subsets of the queries from the Mr. TYDI training (development) sets. The main difference is that MIRACL provides richer annotations for more passages. For the new languages, the training (development) data consist of entirely of newly generated queries.
- **Test-A sets:** Similar to the training and development sets, the test-A sets align with the test sets in Mr. TYDI. However, the test-A sets exist *only* for the existing languages.
- **Test-B sets:** For all languages, the test-B sets are comprised entirely of new queries that have never been released before (compared to test-A sets, whose queries ultimately draw from TYDI and thus have been publicly available for quite some time now). These queries can be viewed as a *true* held-out test set.

Although the training, development, and test-A sets of MIRACL languages that overlap with Mr. TYDI align with Mr. TYDI, in some cases there are fewer queries in MIRACL than in their corresponding Mr. TYDI splits. This is because our annotators were unable to find relevant passages for some queries from Mr. TYDI; we call these "invalid" queries and removed them from the corresponding MIRACL splits (details in Section 3.2.5).

To evaluate the quality of system outputs, we use standard information retrieval metrics, nDCG@10 and Recall@100. These serve as the official metrics of MIRACL. In addition to releasing the evaluation resources, we organized a WSDM Cup challenge at the

Lang	ISO	Train		Dev		Test-A		Test-B		# Passages	# Articles	Avg. Q Len	Avg. P Len
		# Q	# J	# Q	# J	# Q	# J	# Q	# J				
Arabic	ar	3,495	25,382	2,896	29,197	936	9,325	1,405	14,036	2,061,414	656,982	6	54
Bengali	bn	1,631	16,754	411	4,206	102	1,037	1,130	11,286	297,265	63,762	7	56
English	en	2,863	29,416	799	8,350	734	5,617	1,790	18,241	32,893,221	5,758,285	7	65
Finnish	fi	2,897	20,350	1,271	12,008	1,060	10,586	711	7,100	1,883,509	447,815	5	41
Indonesian	id	4,071	41,358	960	9,668	731	7,430	611	6,098	1,446,315	446,330	5	49
Japanese	ja	3,477	34,387	860	8,354	650	6,922	1,141	11,410	6,953,614	1,133,444	17	147
Korean	ko	868	12,767	213	3,057	263	3,855	1,417	14,161	1,486,752	437,373	4	38
Russian	ru	4,683	33,921	1,252	13,100	911	8,777	718	7,174	9,543,918	1,476,045	6	46
Swahili	sw	1,901	9,359	482	5,092	638	6,615	465	4,620	131,924	47,793	7	36
Telugu	te	3,452	18,608	828	1,606	594	5,948	793	7,920	518,079	66,353	5	51
Thai	th	2,972	21,293	733	7,573	992	10,432	650	6,493	542,166	128,179	42	358
Spanish	es	2,162	21,531	648	6,443	–	–	1,515	15,074	10,373,953	1,669,181	8	66
Persian	fa	2,107	21,844	632	6,571	–	–	1,476	15,313	2,207,172	857,827	8	49
French	fr	1,143	11,426	343	3,429	–	–	801	8,008	14,636,953	2,325,608	7	55
Hindi	hi	1,169	11,668	350	3,494	–	–	819	8,169	506,264	148,107	10	69
Chinese	zh	1,312	13,113	393	3,928	–	–	920	9,196	4,934,368	1,246,389	11	121
German	de	–	–	305	3,144	–	–	712	7,317	15,866,222	2,651,352	7	58
Yoruba	yo	–	–	119	1,188	–	–	288	2,880	49,043	33,094	8	28
Total		40,203	343,177	13,495	130,408	7,611	76,544	17,362	174,496	106,332,152	19,593,919	10	77

Table 3.4: Descriptive statistics for all languages in MIRACL, organized by split. # **Q**: number of queries; # **J**: number of labels (relevant and non-relevant); # **Passages**: number of passages; # **Articles**: number of Wikipedia articles from which the passages were drawn; # **Avg. Q Len**: average number of tokens per query; # **Avg. P Len**: average number of tokens per passage. Tokens are based on characters for `th`, `ja`, and `zh`, otherwise delimited by whitespace. Underlined are the new languages in MIRACL.

WSDM 2023 conference to encourage participation from the community, where the test-B sets were used for final evaluation. Among the 18 languages, German (`de`) and Yoruba (`yo`), the bottom block in Table 3.4, were selected as *surprise languages*, while the rest are *known languages*. This distinction was created for the official WSDM competition. Whereas data from the known languages were released in October 2022, the identity of the surprise languages was concealed until two weeks before the competition deadline in January 2023. For the known languages, participants were given ample data and time to train language-specific models. On the other hand, for the surprise languages, no training splits were provided to specifically evaluate retrieval under a limited data and time condition. However, since the WSDM Cup challenge has concluded, the distinction between surprise and known languages is no longer relevant.

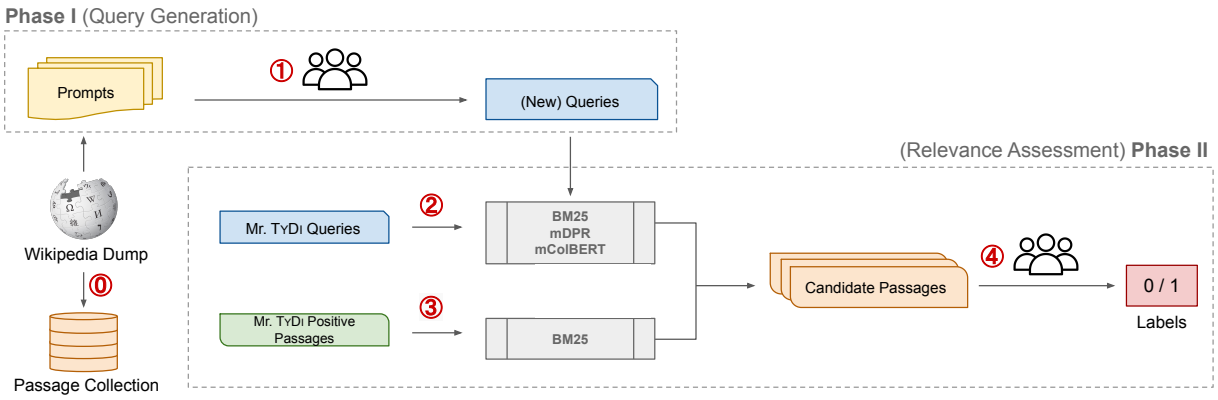


Figure 3.4: Diagram illustrating the MIRACL annotation workflow.

3.2.3 Dataset Construction

To build MIRACL, we hired native speakers as annotators to provide high-quality queries and relevance judgments. At a high level, our workflow comprised two phases: First, the annotators were asked to generate well-formed queries based on "prompts" [20]. Then, they were asked to assess the relevance of the top-k query–passage pairs produced by an ensemble baseline retrieval system.

An important feature of MIRACL is that our dataset was *not* constructed via crowd-sourced workers, unlike other previous efforts such as SQuAD [109]. Instead, we hired 31 annotators (both part-time and full-time) across all languages. Each annotator was interviewed prior to being hired and was verified to be a native speaker of the language they were working in. Our team created a consistent onboarding process that included carefully crafted training sessions. The annotation process lasted for 5 months (from late April 2022 to the end of September 2022).

Throughout the annotation process, we constantly checked randomly sampled data to monitor annotation quality (see Section 3.2.4). Whenever issues were detected, we promptly communicated with the annotators to clarify the problems so that they were resolved expeditiously. Constant interactions with the annotators helped minimize errors and misunderstandings. We believe that this design yielded higher-quality annotations than what could have been obtained by crowd-sourcing.

In total, MIRACL represents over 10k hours of assessor effort, or around five person years. We offered annotators the hourly rate of $18.50 per hour (converted into USD). For reference, the local minimum wage is $11.50 USD/hr.

Corpora Preparation For each MIRACL language, we prepared a pre-segmented passage corpus from a raw Wikipedia dump. For the existing languages in Mr. TYDI, we used exactly the same raw Wikipedia dump as Mr. TYDI and TYDI from early 2019. For the new languages, we used releases from March 2022. We parsed the Wikipedia articles using WikiExtractor[3] and segmented them into passages based on natural discourse units using two consecutive newlines in the wiki markup as the delimiter. Query–passage pairs formed the basic annotation unit (Step ⓪ in Figure 3.4).

Annotation Workflow MIRACL was created using a two-phase annotation process adapted from TYDI, which was in turn built on best practices derived from Clark et al. [20]. The two phases are *query generation* and *relevance assessment*.

Query Generation. In Phase I, annotators were shown "prompts", comprising the first 100 words of randomly selected Wikipedia articles that provide contexts to elicit queries. Following Clark et al. [20], the prompts are designed to help annotators write queries for which they "seek an answer."

To generate high-quality queries, we asked annotators to avoid generating queries that are directly answerable by the prompts themselves. They were asked to generate well-formed natural language queries likely (in their opinion) to have precise, unambiguous answers. Using prompts also alleviates the issue where the queries may be overly biased towards their personal experiences. We also gave the annotators the option of skipping any prompt that did not "inspire" any queries. (step ① in Figure 3.4)

Note that in this phase, annotators were asked to generate queries *in batch* based on the prompts, and did not proceed to the next phase before finishing tasks in this phase. Thus, the annotators had not yet examined any retrieval results (i.e., Wikipedia passages) at this point. Moreover, we did not suggest to annotators during the entire process that the queries should be related to Wikipedia in order to prevent the annotators from writing oversimplified or consciously biased queries. Therefore, it could be the case that their queries cannot be readily answered by the information contained in the corpora.

Relevance Assessment. In the second phase, for each query from the previous phase, we asked the annotators to judge the binary relevance of the top-k candidate passages ($k = 10$) from an ensemble retrieval system that combines three separate models. (Step ② in Figure 3.4):

- **BM25** [110], a traditional retrieval algorithm based on lexical matching, which has been shown to be a robust baseline when evaluated zero-shot across domains and

[3] https://github.com/attardi/wikiextractor

languages [124, 149]. We used the implementation in Anserini [143], using default parameters and the corresponding language-specific analyzer in Lucene if it exists. If not, we simply used the white space tokenizer.

- **mDPR** [58, 149], a single-vector dense retrieval method that has proven to be effective for many retrieval tasks. Model detailed are mentioned in Section 2.1.2. This model was trained using the Tevatron toolkit [39] starting from an mBERT checkpoint and then fine-tuned using the training set of MS MARCO Passage. Retrieval was performed in a zero-shot manner.

- **mColBERT** [59, 13], a multi-vector dense retrieval model that has been shown to be effective in both in-domain and out-of-domain contexts [124, 114]. This model was trained using the authors' official repository.[4] Same as mDPR, the model was initialized from mBERT and fine-tuned on MS MARCO Passage; retrieval was also performed zero shot.

For each query, we retrieved the top 1000 passages using each model, then performed ensemble fusion by first normalizing all retrieval scores to the range $[0, 1]$ and then averaging the scores. A final ranked list was then generated from these new scores. Based on initial experiments, we found that annotating the top 10 passages per query yielded a good balance in terms of obtaining diverse passages and utilizing annotator effort.

As mentioned in Section 3.2.1, we rectified the inconsistent passage segmentation in Mr. TYDI by re-segmenting the Wikipedia articles of TYDI. The downside, unfortunately, is that annotated passages from Mr. TYDI may no longer exist in the MIRACL corpora. Thus, to take advantage of existing annotations, for queries from the 11 existing languages in Mr. TYDI, we augmented the set of passages to be assessed with "projected" relevant passages. This allowed us to take advantage of existing annotations transparently in our workflow. To accomplish this, we used relevant passages from Mr. TYDI as queries to search the corresponding MIRACL corpus using BM25. As the passages in both corpora differ in terms of segmentation but not content, the top retrieved passages are likely to have substantial overlap with the annotated passage in Mr. TYDI (and hence are also likely to be relevant). This is shown as Step ③ in Figure 3.4.

We used a simple heuristic to determine how many of these results to re-assess. If the score of the top retrieved passage from MIRACL is 50% higher than the score of the passage ranked second, we have reasonable confidence (based on initial explorations) that the top passage is a good match for the original relevant passage. In this case, we only add

[4] https://github.com/stanford-futuredata/ColBERT#colbertv1

the top passage to the set of candidates that the assessor considers. Otherwise, we add the top 5 passages.

Once the candidate passages are prepared, the annotators are asked to provide a binary label for each query–passage pair (1 = "relevant" and 0 = "not relevant"). This is shown in Step ④ in Figure 3.4. Note that the "projected" passages prepared from Step ③ are not identified to the annotators. In all cases, they simply receive a set of passages to label per query, without any explicit knowledge of where the passages came from.

In our design, each query–passage pair was only labeled by one annotator, which we believe is a better use of limited resources, compared to the alternative of multiple judgments over fewer queries. The manual assessment proceeded in batches of approximately 1000 query–passage pairs. Our annotators differed greatly in speed, but averaged over all languages and individuals, each batch took roughly 4.25 hours to complete.

Fold Creation and Data Release. At a high-level, MIRACL contains two classes of queries: those inherited from Mr. TYDI and those that were created from scratch, from which four splits were prepared. During the annotation process, all queries followed the same workflow. After the annotation process concluded, we divided MIRACL into training sets, development sets, test-A sets, and test-B sets. For existing languages, the training, development, and test-A sets align with the training, development, and test sets in Mr. TYDI, and the test-B sets are formed by the newly generated queries. For the new (known) languages, all generated queries were split into training, development, and test-B sets with a split ratio of 50%, 15%, and 35%. Note that there are no test-A sets for these languages.

3.2.4 Quality Control

To ensure data quality, we implemented two processes: automatic heuristic verification that includes simple daily checks and manual periodic assessment executed by human reviewers on a random subset of annotations.

Automatic Verification Automatic heuristic verification was applied to both phases daily, using language-specific heuristics to flag "obviously bad" annotations. In Phase I, we flagged the generated query if it was: (1) left empty; (2) similar to previous queries

Query:
ইন্দোনেশিয়ার রাজধানী কোথায় ?

Where is the capital of Indonesia?

Candidate Passage:
সুরিনামের রাজধানী পারামারিবোতে ইন্দোg gনেশিয়ার একটি দূতাবাস রয়েছে। এই দূতাবাসের মাধ্যমে, ইন্দোনেশিয়ার সাথে গায়ানার বিভিন্ন বিষয়াবলী এবং কার্যক্রম সম্পন্ন করা হয়। অপরদিকে, ইন্দোনেশিয়ার রাজধানী জাকার্তায় সুরিনামের একটি দূতাবাস রয়েছে। ইন্দোনেশিয়া এবং সুরিনাম উভয় দেশই বিশ্ব বাণিজ্য সংস্থা (ডব্লিউটিও) এবং ফোরাম অব ইস্ট এশিয়া-লাতিন আমেরিকা কো-অপারেশন এর সদস্য।.

Indonesia has an embassy in Paramaribo, the capital of Suriname. Through this embassy, Guyana's various affairs and activities with Indonesia are carried out. On the other hand, Suriname has an embassy in **Jakarta, the capital of Indonesia**. Both Indonesia and Suriname are members of the World Trade Organization (WTO) and the Forum of East Asia-Latin America Cooperation.

(a) Example of an annotation mistake: the relevant information (highlighted in red) appears in the middle of a passage containing mostly non-relevant material.

Query:
هل إسرائيل هي دولة دينية ؟
Is Israel a religious state?

Candidate Passage:
أصدرت إسرائيل منذ تأسيسها العديد من القوانين التي تعكس الهوية اليهودية وقيم الأغلبية)حوالي 75٪ في عام 2016 (من مواطنيها .ومع ذلك فإن النقاش الدائر في إسرائيل إزاء كون الدولة علمانية أم دينية جعل النقاش يتركز حول طبيعة الدولة اليهودية على وجه التحديد . كما ركز جانب آخر من النقاش على وضع الأقليات في إسرائيل وعلى وجه الخصوص السكان العرب الإسرائيليين.

Since its founding, Israel has passed many laws that reflect the Jewish identity and the values of the majority (about 75% in 2016) of its citizens. However, the debate in Israel about whether the state is secular or religious has made the debate focus on the nature of the Jewish state specifically. Another aspect of the discussion focused on the situation of minorities in Israel and in particular the Israeli Arab population.

(b) Example of a partially relevant query–passage pair.

Figure 3.5: Examples of disagreements between annotators and reviewers.

in the same language;[5] (3) missing interrogative indicators;[6] (4) overly short or long.[7] In Phase II, we flagged the label if it was left empty or an invalid value (i.e., values that are not 0 or 1).

The most extreme cases of bad annotations (e.g., empty or duplicate queries) were removed from the dataset, while minor issues were flagged but the data were retained. For both cases, whenever the metrics dropped below a reasonable threshold,[8] we scheduled a meeting with the annotator in question to discuss. This allowed us to minimize the number of obvious errors while giving and receiving constructive feedback.

Manual Assessment Manual assessment was also applied to both phases, where we hired another group of native speakers of each language as reviewers (with minor overlap). Similarly, we provided consistent onboarding training to each reviewer.

In **Phase I**, reviewers were given both the prompts and the generated queries, and asked to apply a checklist to determine whether the queries met our requirements. Criteria include the examination of the query itself (e.g., spelling or syntax errors, fluency, etc.) and whether the query could be answered directly by the prompt, which we wished to avoid (see Section 3.2.3).

How fluent are the queries? Review results showed that approximately 12% of the questions had spelling or syntax errors, or "sound artificial". However, almost all these flaws (over 99%) did not affect the understanding of the question itself. We thus retained these "flawed" queries to reflect real-world distributions.

How related are the queries to the prompts? We measured the lexical overlap between the queries and their corresponding prompts to understand their connections. Our analysis shows approximately two words of overlap on average, which is consistent with statistics reported in TYDI [20]. Overlaps primarily occur in entities or stopwords. We thus conclude that the generated queries are reasonably different from the given prompts.

In **Phase II**, reviewers were provided the same guidance as annotators performing the relevance assessment. They were asked to (independently) label a randomly sampled

[5] We measured similarity using Levenshtein distance. A query was flagged if its similarity score to any previous query was greater than 75%.

[6] For example, writing system- and language-specific interrogative punctuation and particles.

[7] For `zh`, `ja` and `th`, length is measured in characters, where the expected range is $[6, 40]$ for `zh` and `ja`, and $[6, 120]$ for `th`. Otherwise, length is measured in tokens delimited by white space, with expected range $[3, 15]$.

[8] In practice, we set this threshold to 0.8 in terms of the percentage of questionable annotations in each submission.

	MIRACL				SQuAD
	Train	**Dev**	**Test-A**	**Test-B**	
HOW	15%	17%	19%	6%	12%
WHAT	26%	27%	31%	34%	51%
WHEN	26%	25%	20%	6%	8%
WHERE	6%	7%	4%	6%	5%
WHICH	1%	1%	2%	25%	5%
WHO	13%	11%	9%	19%	11%
WHY	1%	1%	1%	1%	2%
YES/NO	6%	7%	4%	6%	<1%

Table 3.5: Query distribution of each split in English MIRACL, compared to SQuAD.

subset of the query–passage pairs. The degree of agreement on the overlapping pairs is used to quantify the quality of the relevance labels. As a high-level summary, we observe average agreements of over 80% on query–passage relevance.

Why are there disagreements? As is well-known from the IR literature dating back many decades [128], real-world annotators disagree due to a number of "natural" reasons. To further understand why, we randomly sampled five query–passage pairs per language for closer examination. After some iterative sense-making, we arrived at the conclusion that disagreements come from both mislabeling and partially relevant pairs. We show examples of the two cases in Figure 3.5a and Figure 3.5b.

Figure 3.5a depicts a common mislabeling case, where a passage starting with misleading information is labeled as non-relevant. We observe more mislabeling errors from the reviewers compared to the annotators, which may be attributed to the annotators' greater commitment to the task and more experience in performing it. That is, in these cases, the reviewers are more often "wrong" than the original assessors.

Partially relevant annotations can take on various forms. The most frequent case is exemplified by Figure 3.5b, where a passage is related to the query but does not provide a precise answer. Differences in "thresholds" for relevance are inevitable under a binary relevance system. Furthermore, disagreements often arise when an annotator labels a passage as non-relevant, but a reviewer labels it as relevant. These cases suggest that annotators exercise greater caution and are more "strict" when they are uncertain. Overall, these analyses convinced us that the generated queries and annotations are of high quality.

3.2.5 Analysis

Question Words We first analyze question types, following [20]. Table 3.5 shows the distribution of the question word in the English dataset of MIRACL, along with SQuAD [109] as a point of reference. The MIRACL statistics are grouped by splits in Table 3.5. The test-B column shows the distribution of the new queries generated with the workflow described in Section 3.2.3, whereas the other columns correspond to queries derived from Mr. TYDI.

We observe a more balanced distribution of question words in MIRACL compared to SQuAD. In addition, the question words highlight a distribution shift between Mr. TYDI and the new queries. More specifically, the test-B split contains more WHICH and WHO queries, while containing fewer HOW and WHEN queries, compared to the other splits inherited from Mr. TYDI. This also shows that exactly replicating a query generation process is challenging, even when closely following steps outlined in prior work.

Queries with No Relevant Passages Recall that since the query generation and relevance assessment phases were decoupled, for some fraction of queries in each language, annotators were not able to identify any relevant passage in the pairs provided to them. Because we aimed to have at least one relevant passage per query, these queries—referred to as "invalid" for convenience—were discarded from the final dataset.

For each language, we randomly sampled five invalid queries and spot-checked them using a variety of tools, including interactive searching on the web. From this, we identified a few common reasons for the lack of relevant passages in Wikipedia: (1) query not asking for factual information, for example, "*Kuinka nuoret voivat osallistua suomalaisessa yhteiskunnassa?*" (Finnish, "*How can young people participate in Finnish society?*"); (2) query being too specific, for example, " 华盛顿2021年新增人口多少? "(Chinese, "*What is the new population of Washington in 2021?*"); (3) inadequate Wikipedia content, where the query seems reasonable, but the relevant information could not be found in Wikipedia due to the low-resource nature of the language (even after interactive search with Google), for example, "*Bawo ni aaye Tennis se tobi to?*" (Yoruba, "*How big is the tennis court?*").

We note that just because no relevant passage was found in the candidates presented to the annotators, it is not necessarily the case that no relevant passage exists in the corpus. However, short of exhaustively assessing the corpus (obviously impractical), we cannot conclude this with certainty. Nevertheless, for dataset consistency we decided to discard "invalid" queries from MIRACL.

Discussion of MIRACL Languages One important consideration in the selection of languages is **linguistic diversity**, a motivating design principle in TYDI, which we built on in MIRACL. Our additional languages introduced *similar* pairs as well, opening up new research opportunities. A summary can be found in Table 3.6.

Families and sub-families provide a natural approach to group languages based on historical origin. Lineage in this respect explains a large amount of resemblance between languages, for example, *house* in English and *haus* in German [44]. The 18 languages in MIRACL are from 10 language families and 13 sub-families, shown in the first two columns of Table 3.6.

The notion of *synthesis* represents an important morphological typology, which categorizes languages based on how words are formed from *morphemes*, the smallest unit of meaning. The languages in MIRACL are balanced across the three synthesis types (*analytic*, *agglutinative*, and *fusional*), which form a spectrum with increasing complexity in word formation when moving from analytic to fusional languages. [104, 30, 95]. Additional features of the languages are summarized in Table 3.6, including the written scripts, word order, use of white space for delimiting tokens, and grammatical gender.

In our context, linguistic diversity is critical to answering important questions about the transfer capabilities of multilingual language models [103], including whether certain typological characteristics are intrinsically challenging for neural language models [42] and how to incorporate typologies to improve the effectiveness of NLP tasks [105, 56]. While researchers have examined these research questions, most studies have not been in the context of retrieval specifically.

In this respect, MIRACL can help advance the state of the art. Consider language family, for example: our resource can be used to compare the transfer capacity between languages at different "distances" in terms of language kinship. While Mr. TYDI provides some opportunities to explore this question, the additional languages in MIRACL enrich the types of studies that are possible. For example, we might consider both contrastive pairs (i.e., those that are very typologically different) as well as similar pairs (i.e., those that are closer) in the context of multilingual language models.

Similar questions abound for synthesis characteristics, written script, word order, etc. For example, we have three exemplars in the Indo-Iranian sub-family (Bengali, Hindi, and Persian): despite lineal similarities, these languages use different scripts. How does "cross-script" relevance transfer work in multilingual language models? We will explore these questions in Chapter 5 based on MIRACL.

The other important consideration in our choice of languages is the demographic distribution of language speakers. As noted in the introduction, information access is a fun-

Language Family	Language Sub-Family	Language	Script	Synthesis	Word Order	White Space	Gender	# Speakers	Wikipedia Size
Dravidian	South-Central	Telugu	Telugu	agglutinative	SOV	✓	✓	96M	66,353
Koreanic	–	Korean	Hangul	agglutinative	SOV	✓	×	80M	437,373
Japonic	–	Japanese	Japanese	agglutinative	SOV	×	×	128M	1,133,444
Austronesian	Malayo-Polynesian	Indonesian	Latin	agglutinative	SVO	✓	×	300M	446,330
Uralic	Finno-Ugric	Finnish	Latin	agglutinative	SVO	✓	×	6M	447,815
Niger–Congo	Atlantic-Congo	Swahili	Latin	agglutinative	SVO	✓	×	83M	47,793
		Yoruba	Latin	analytic	SVO	✓	×	52M	33,094
Sino-Tibetan	Sinitic	Chinese	Chinese	analytic	SVO	×	✓	1350M	1,246,389
Kra–Dai	Tai	Thai	Thai	analytic	SVO	×	✓	72M	128,179
Indo-European	Germanic	English	Latin	analytic	SVO	✓	✓	1130M	5,758,285
		German	Latin	fusional	SVO, SOV	✓	✓	178M	2,651,352
	Italic	French	Latin	fusional	SVO	✓	✓	398M	2,325,608
		Spanish	Latin	fusional	SVO	✓	✓	592M	1,669,181
	Balto-Slavic	Russian	Cyrillic	fusional	SVO	✓	✓	260M	1,476,045
	Indo-Iranian	Bengali	Bengali	fusional	SOV	✓	✓	337M	63,762
		Hindi	Devanagari	fusional	SOV	✓	✓	592M	148,107
		Persian	Arabic	agglutinative	SOV	✓	×	110M	857,827
Afro-Asiatic	Semitic	Arabic	Arabic	fusional	VSO	✓	✓	630M	656,982

Table 3.6: Characteristics of languages in MIRACL, including language (sub-)family, script, linguistic typologies (synthesis, word order, and gender), number of speakers (L1 and L2), and number of Wikipedia articles. Under **White Space**, ✓ indicates the language uses white space as the token delimiter; under **Gender**, ✓ indicates that gender is evident in the language. Under # **Speakers** and **Wikipedia Size**, cells are highlighted column-wise with the color gradient ranging from **green** (high values) to **red** (low values), where the **Wikipedia Size** column is identical to the # **Articles** column in Table 3.4. Underlined are the new languages not included in Mr. TYDI.

damental human right that ought to extend to *every* inhabitant of our planet, regardless of the languages they speak.

We attempt to quantify this objective in the final two columns of Table 3.6, which presents statistics of language speakers and Wikipedia articles. We count both L1 and L2 speakers.[9] Both columns are highlighted column-wise based on the value, from **green** (high values) to **red** (low values). We can use the size of Wikipedia in that language as a proxy for the amount of language resources that are available. We see that in many cases, there are languages with many speakers, but are poor in resources. Particularly noteworthy examples include Telugu, Indonesian, Swahili, Yoruba, Thai, Bengali, and Hindi. We hope that the inclusion of these languages will catalyze interest in multilingual retrieval and in turn benefit large populations for whom the languages have been historically overlooked

[9]L1 are the native speakers; L2 includes other speakers who learned the language later in life.

by the mainstream IR research community.

3.2.6 Experiments

Baseline scores for these retrieval models are shown in Table 3.7 in terms of the two official retrieval metrics of MIRACL, reproducible on Pyserini IR toolkit [72]. The baselines include the three methods used in the ensemble system introduced in Section 3.2.3 (BM25, mDPR, mColBERT) plus the following approaches:

- **Hybrid** combines the scores of BM25 and mDPR results. For each query–passage pair, the hybrid score is computed as $s_{\text{Hybrid}} = \alpha \cdot s_{\text{BM25}} + (1 - \alpha) \cdot s_{\text{mDPR}}$, where we set $\alpha = 0.5$ without tuning. Scores of BM25 and mDPR (s_{BM25} and s_{mDPR}) are first normalized to $[0, 1]$.
- **mContriever** [51] adopts additional pretraining with contrastive loss based on unsupervised data prepared from CCNet [134], which demonstrates improved effectiveness in downstream IR tasks. We used the authors' released multilingual checkpoint, where the model was further fine-tuned on English MS MARCO after additional pretraining.[10]
- **In-language fine-tuned mDPR** follows the same model configuration as the mDPR baseline, but we fine-tuned each model on the MIRACL training set of the target language rather than MS MARCO. Here, the negative examples are sampled from the labeled negatives and the unlabeled top-30 candidates from BM25.

Note that all languages in MIRACL are included in the pretraining corpus of mBERT, which provides the backbone of the mDPR and mColBERT models. However, three languages (`fa`, `hi`, `yo`) are not included in CCNet [134], the dataset used by mContriever for additional pretraining. Code, documentation, and instructions for reproducing these baselines have been released together with the MIRACL dataset and can be found on the MIRACL website.

These results provide a snapshot of the current state of research. We see that across these diverse languages, mDPR does not substantially outperform decades-old BM25 technology, although BM25 exhibits much lower effectiveness in French and Chinese. Nevertheless, the BM25–mDPR hybrid provides a strong zero-shot baseline that outperforms all the other individual models on average nDCG@10. Interestingly, the BM25–mDPR hybrid

[10] https://huggingface.co/facebook/mcontriever-msmarco

ISO	nDCG@10						Recall@100					
	BM25	**mDPR**	**Hyb.**	**mCol.**	**mCon.**	**in-L.**	**BM25**	**mDPR**	**Hyb.**	**mCol.**	**mCon.**	**in-L.**
ar	0.481	0.499	0.673	0.571	0.525	0.649	0.889	0.841	0.941	0.908	0.925	0.904
bn	0.508	0.443	0.654	0.546	0.501	0.593	0.909	0.819	0.932	0.913	0.921	0.917
en	0.351	0.394	0.549	0.388	0.364	0.413	0.819	0.768	0.882	0.801	0.797	0.751
fi	0.551	0.472	0.672	0.465	0.602	0.649	0.891	0.788	0.895	0.832	0.953	0.907
id	0.449	0.272	0.443	0.298	0.392	0.414	0.904	0.573	0.768	0.669	0.802	0.823
ja	0.369	0.439	0.576	0.496	0.424	0.570	0.805	0.825	0.904	0.895	0.878	0.880
ko	0.419	0.419	0.609	0.487	0.483	0.472	0.783	0.737	0.900	0.722	0.875	0.807
ru	0.334	0.407	0.532	0.477	0.391	0.521	0.661	0.797	0.874	0.866	0.850	0.850
sw	0.383	0.299	0.446	0.358	0.560	0.644	0.701	0.616	0.725	0.692	0.911	0.909
te	0.494	0.356	0.602	0.462	0.528	0.781	0.831	0.762	0.857	0.830	0.961	0.957
th	0.484	0.358	0.599	0.481	0.517	0.628	0.887	0.678	0.823	0.845	0.936	0.902
es	0.319	0.478	0.641	0.426	0.418	0.409	0.702	0.864	0.948	0.842	0.841	0.783
fa	0.333	0.480	0.594	0.460	0.215	0.469	0.731	0.898	0.937	0.910	0.654	0.821
fr	0.183	0.435	0.523	0.267	0.314	0.376	0.653	0.915	0.965	0.730	0.824	0.823
hi	0.458	0.383	0.616	0.470	0.286	0.458	0.868	0.776	0.912	0.884	0.646	0.777
zh	0.180	0.512	0.526	0.398	0.410	0.515	0.560	0.944	0.959	0.908	0.903	0.883
de	0.226	0.490	0.565	0.334	0.408	–	0.572	0.898	0.898	0.803	0.841	–
yo	0.406	0.396	0.374	0.561	0.415	–	0.733	0.715	0.715	0.917	0.770	–
K. Avg	0.394	0.415	0.578	0.441	0.433	0.535	0.787	0.788	0.889	0.828	0.855	0.856
S. Avg	0.316	0.443	0.470	0.448	0.412	–	0.653	0.807	0.807	0.860	0.806	–
Avg	0.385	0.418	**0.566**	0.441	0.431	–	0.772	0.790	**0.880**	0.832	0.849	–

Table 3.7: Baseline results on the MIRACL dev set, where "K. Avg." and "S. Avg" indicate the average scores over the known (ar–zh) and surprise languages (de–yo). (**Hyb.**: Hybrid results of BM25 and mDPR; **mCol.**: mColBERT; **mCon.**: mContriever; **in-L**: in-language fine-tuned mDPR) Best results are in **bold**.

even outperforms in-language fine-tuned mDPR on most of the languages, except for the ones that are comparatively under-represented in mBERT pretraining (e.g., sw, th, hi). Overall, these results show that plenty of work remains to be done to advance multilingual retrieval.

3.3 Discussion

This chapter has presented Mr. TYDI and MIRACL, two datasets designed to provide a reliable foundation for multilingual retrieval research. Together, they address two needs: enough labeled query-passage pairs to train neural retrieval models, and sufficiently careful human annotation to evaluate those models across diverse languages. Mr. TYDI takes the first step by adapting existing question answering datasets into a multilingual retrieval

benchmark, while MIRACL expands both the language coverage and the density of relevance judgments.

These resources also make it possible to ask questions on training and understanding multilingual embedding models. With sufficient training instances and more reliable evaluation, we can systematically study how multilingual embedding models should be trained under different resource conditions. At the same time, the language diversity in these datasets provides an empirical basis for examining how models transfer across languages, and how factors such as shared tokens may shape that transfer. The next chapter (Chapter 4) uses these datasets to study training practices for multilingual embedding models; the following chapter (Chapter 5) turns to understanding what these results reveal about how multilingual language models process and connect text across languages.

Chapter 4

Training

The results in Sections 3.1.4 and 3.2.6 show that, under unsupervised conditions, plain mDPR does not perform much better than BM25, despite its strong in-language results. This gap motivates the central question of this chapter: how can we train embedding models that provide better support for non-English languages?

This question is complex in reality given the extra language dimension. Suppose we are interested in building a dense retrieval model for target language $\mathcal{L}$. What should we do if we don't have any in-domain training data in the target language (by "in-domain", we mean training data that is drawn from the same distribution as the evaluation data)? Should we rely entirely on cross-lingual transfer from another out-of-domain dataset such as MS MARCO in English (by "out-of-domain", we mean training data that is *not* drawn from the same distribution as the evaluation data)? How can we exploit versions of MS MARCO that have been automatically translated into different languages, including the target language? Alternatively, if translations are not available, can we exploit unlabeled data in the target language, and how does that compare to using machine translated training data? What if we *do* have (perhaps, limited) in-domain training data in the target language? Is a large out-of-domain dataset such as MS MARCO (in English or automatically translated into other languages) still beneficial?

More open questions: Transformers can be fine-tuned in different "stages" [41, 139, 152], so it is certainly possible to exploit datasets in different languages at different stages when building from a multilingual transformer. Should we? Does the answer change if the datasets draw from linguistically unrelated families? Or are written in entirely different scripts? Are the cross-lingual transfer effects more prominent at one stage compared to the others? What about using different pretrained transformers as starting points? And to

consider an entirely different approach, should we prefer monolingual transformers if one is available?

The goal of our work is to answer these myriad questions and to begin to develop a set of best practices for training multilingual dense retrieval models. Our exploration is organized into three main scenarios, as follows:

1. **Have model, no data** — scenario (1): The target language $\mathcal{L}$ is covered by a pretrained multilingual transformer, but unfortunately we do not have any in-domain training data in the target language.
2. **Have model and data** — scenario (2): The target language $\mathcal{L}$ is covered by a pretrained multilingual transformer and we have in-domain training data (i.e., manually labeled relevance judgments). This breaks down into two sub-cases, one where we have in-domain training data in the target language $\mathcal{L}$ and the other where we have in-domain training data in some other (non-target) language $\mathcal{K}$. We also consider the case where a *monolingual* transformer for language $\mathcal{L}$ is available.
3. **Have data, no model** — scenario (3): No pretrained transformer is available for the target language $\mathcal{L}$, but we do have in-domain training data in the target language.

4.1 Experimental Design

4.1.1 Design Space

The application of transformers to tackle modern NLP problems follows an approach that begins with pretraining followed by fine-tuning. In many cases, the pretraining step can be omitted altogether by using an existing model checkpoint as a starting point; such models are widely shared, for example, on the HuggingFace model hub [136]. IR researchers have adopted the same high-level approach for tackling retrieval tasks [74], and training a DPR model is no exception. Typically, researchers begin with a pretrained model checkpoint as the backbone and further fine-tune the model with relevance judgments (e.g., from the MS MARCO passage ranking dataset) in a bi-encoder architecture as discussed above.

However, beyond this general "pretrain then fine-tune" recipe, there are several more detailed steps that deserve consideration. Together, they circumscribe the design space of our experiments. A more detailed articulation of the steps is shown in Figure 3.4 and discussed below:

- **Pretrain (PT).** In the pretraining step, a transformer model with randomly initialized weights is trained "from scratch" on a large amount of unlabeled text in a *self-supervised* manner. The goal of pretraining is to imbue the model with "general-purpose" knowledge of language and the world that can be shared across downstream tasks. This is the most computationally expensive step, involving a massive amount of resources. The outputs of this step are usually shared with the community, for example, mBERT and XLM-R. In our work, we do not pretrain any models from scratch, and instead start from these existing model checkpoints.

- **Additional pretrain (Additional PT).**

  Researchers have found it helpful to include additional self-supervised pretraining on top of a pretrained checkpoint, typically using a "general purpose" objective such as MLM [111] or some IR-specific pretraining regime [51, 84, 83]. In this work, mContriever is selected as a representative model that exploits IR-specific additional pretraining; our experiments build on publicly available model checkpoints. Separately, we perform additional pretraining ourselves in the context of DDR [145] and our improvements to that model.

- **Pre–fine-tune (pFT)**. Similar to additional pretraining, pre–fine-tuning is usually adopted to establish a better starting point for fine-tuning, or alternatively, the model can be used in a zero-shot setting. This step typically uses manually labeled (i.e., *supervised*) datasets, which lead to better task performance than the self-supervised approaches described in the previous two steps. However, pre–fine-tuning data usually comes from a different distribution (e.g., language or domain) than the final evaluation data. In a retrieval task, an example is using the MS MARCO passage ranking dataset for pre–fine-tuning, which has been shown to be effective [152]. In this work, we pre–fine-tune mDPR models ourselves based on the model checkpoints (i.e., backbones) described above.

- **Fine-tune (FT).** In the final step of building a transformer-based application, the model is fine-tuned on manually labeled data that draws from the same distribution as the final evaluation data. That is, fine-tuning typically takes advantage of "in-domain" data, as opposed to pre–fine-tuning, which usually refers to the use of "out-of-domain" data. In this work, we perform all model fine-tuning ourselves.

As mentioned in the introduction, our explorations are grouped into three scenarios: (1) have model, no data; (2) have model and data; (3) have data, no model.

The majority of our experiments are based on mBERT, for multiple reasons: First, keeping the backbone and training regime identical allows us to compare results across the different scenarios. Second, while there are indeed many multilingual (and monolingual) backbones that can serve as starting points for training dense retrieval models, all of them follow essentially the same training steps. As it is impractical to run all experimental conditions with all backbones, we argue that it suffices to experiment with a representative example. We selected mBERT as the representative because it takes "reasonable" time and resources to train, allowing us to run many experimental conditions. In contrast, the training time of an XLM-R_{Large} backbone is around 5× that of mBERT when consuming the same amount of data. In Sections 4.2 and 4.3, we additionally study the effects of different backbones and additional pretraining techniques, but it is impractical to exhaustively explore these approaches under all settings.

4.1.2 Datasets

All experiments and recommendations in this work are based on Mr. TYDI [149], a multilingual retrieval benchmark built on the TYDI [20] dataset (originally developed for question answering). Mr. TYDI covers 11 languages from different language families that are typologically diverse. Among the languages, English, Finnish, Indonesian, and Swahili are written in Latin script, whereas no other language pair shares the same script. The corpus for each language is drawn from Wikipedia, and the query and relevance judgments are prepared by native speakers of that language. Table 3.2 presents statistics of the Mr. TYDI dataset, copied from the original paper [149]. At the time we began our work, this was the only available dataset for evaluating monolingual retrieval in diverse languages; other datasets, for example, XOR-TYDI [8] and CLIRMatrix [121], focus on cross-lingual retrieval. We have since built MIRACL [151], a new multilingual retrieval dataset that spans 18 different languages, but this work does not take advantage of MIRACL.

The test set of Mr. TYDI is used for the evaluation of our dense retrieval models, and thus the training and development sets of Mr. TYDI serve as the in-domain data for fine-tuning. In this work, another three datasets serve as pre–fine-tuning data, as they provide query–passage relevance labels, but not exactly in the same domain (and language) as Mr. TYDI:

- **Natural Questions (NQ).** NQ is a question answering dataset [61], which provides over 300k questions for training. The questions are taken from queries submitted to Google and the passages are curated from Wikipedia. This dataset is widely used by QA researchers.

- **MS MARCO.** The MS MARCO passage ranking dataset grew out of an earlier QA dataset [10], where the queries draw from Bing and the passages comprise snippets of webpages. This dataset provides over 500k manually labeled query–passage pairs and is widely used by IR researchers.

- **mMARCO.** The mMARCO dataset [13] is a multilingual passage ranking dataset automatically translated from the MS MARCO passage ranking dataset above. We use the version that is translated with the Google Translate API.[1] The dataset contains 14 languages in total (including the original English), five of which overlap with languages in Mr. TYDI (`ar`, `en`, `id`, `ja`, and `ru`).

These pre–fine-tuning datasets allow us to explore the design space described in Section 4.1.1, particularly for the scenario where we do not have in-domain data. Specifically, pre–fine-tuned models can be applied in a zero-shot setting.

4.1.3 Data Preparation

For the three pre–fine-tuning datasets, we use the Tevatron [39] versions for NQ[2] and MS MARCO,[3] and the official release for mMARCO.[4] We use version 1.1 of the Mr. TYDI dataset released on HuggingFace.[5] In the training sets used to train our dense retrieval models, positive examples are from the labeled positive passages provided by Mr. TYDI. Unless otherwise specified, the negative examples are prepared from the top-30 results retrieved using the tuned BM25 baseline, following standard practice [58]. In Section 4.2, some experiments take advantage of hard negative mining [140]. In these experiments, the negative examples are mined from the top-30 results retrieved by an initial mDPR model that is initialized from mContriever and then pre–fine-tuned on the entire mMARCO dataset, i.e., the model from row (5) in Table 4.3.

In our experiments, we train models with data from a single language and with data combined from multiple languages. In the multiple languages case, data in the same batch are always drawn from the same language for both the pre–fine-tuning and fine-tuning steps. Pilot experiments indicate that this practice is helpful because it prevents in-batch negatives from degenerating into a language detection task, where the model

[1] https://cloud.google.com/translate
[2] https://huggingface.co/datasets/Tevatron/wikipedia-nq
[3] https://huggingface.co/datasets/Tevatron/msmarco-passage
[4] https://huggingface.co/datasets/unicamp-dl/mmarco
[5] https://huggingface.co/datasets/castorini/mr-tydi

Language	HuggingFace Model
multilingual pretrained transformers	
mBERT	`bert-base-multilingual-cased`
XLM-R_{Large}	`xlm-roberta-large`
mContriever	`facebook/mcontriever`
AfriBERTa	`castorini/afriberta_large`
monolingual pretrained transformers	
Arabic	`asafaya/bert-base-arabic`
English	`bert-base-uncased`
Finnish	`TurkuNLP/bert-base-finnish-cased-v1`
Indonesian	`cahya/bert-base-indonesian-522M`
Korean	`kykim/bert-kor-base`

Table 4.1: HuggingFace models used in our experiments.

tries to distinguish positive passages (in one language) from negative passages (in another language).

For experiments in Section 4.4, where we fine-tune in language $\mathcal{K}$ and evaluate on language $\mathcal{L}$ in a large matrix experiment, we wish to alleviate the effects of training dataset size, since different languages have different numbers of relevance judgments. However, based on pilot experiments, simply downsampling all datasets to the smallest language results in under-fitting for most of the languages. Instead, we randomly sample 3300 training queries from each language, which is close to the size of training queries in Thai. For Bengali, Korean, and Swahili, which have fewer relevance judgments, we retain all available data. It is worth emphasizing that downsampling is applied *only* for the matrix experiments in Section 4.4; all other experiments in this work use all available data.

4.1.4 Experimental Setup

Dense retrieval model training Our models are trained using Tevatron [39], a more efficient and flexible implementation of the original DPR code open-sourced by DPR [58]. Following the standard practices of the community today, we train different dense retrieval models with shared parameters between the query and passage encoders, whereas in the original Mr. TYDI work [149], separate query and passage encoders are trained. In practice, a shared encoder yields comparable effectiveness, but has the advantage of simplified model

management. We initialize the model with different pretrained model checkpoints, all of which are available on HuggingFace [136] (see Table 4.1).

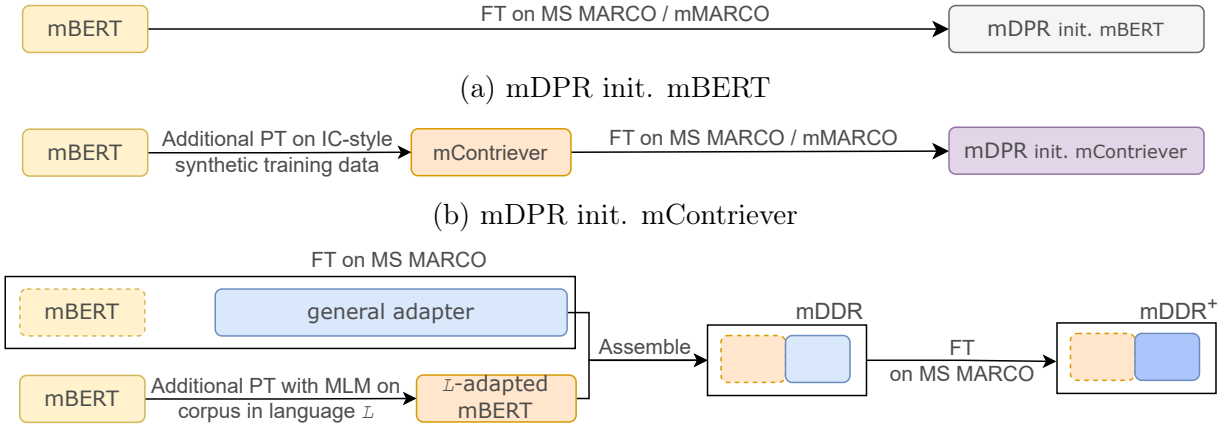
Unfortunately, not all languages have publicly available monolingual pretrained BERT models. Furthermore, we only considered monolingual models that were trained from scratch (not initialized with existing models). In the end, we experimented with monolingual BERT models for five languages in Mr. TYDI: Arabic, English, Finnish, Indonesian, and Korean. These were the only models that we were aware of when we first began our work.

In all experiments, models are trained 40 epochs with the corresponding training data (single or multiple languages) with 128 batch size. As mentioned above, when the model is trained on data in multiple languages, data in the same batch are always drawn from the same language in both pre–fine-tuning and fine-tuning steps. We use the Adam optimizer, setting the learning rate to 4e-5 for experiments trained on a single language and 1e-5 for the others. The maximum lengths of queries and passages are set to 64 and 256, respectively. We tuned the learning rate and the number of epochs on the Mr. TYDI development sets, considering {5e-6, 1e-5, 2e-5, 4e-5, 8e-5} and {10, 20, 30, 40, 50}, respectively, to arrive at settings that appear to work well across all languages.

Additional pretraining with MLM and adapters (mDDR and mDDR$^+$). In both NLP and IR, additional MLM pretraining on the target corpus has been known to be effective for adapting pretrained transformers to a different domain or language [45, 94, 62, 145]. We experiment with DDR [145], a specific technique for efficiently applying additional MLM pretraining to dense retrieval.

DDR [145] was proposed in the context of out-of-domain English retrieval based on "adapter" modules, which allow efficient parameter sharing among models for different languages and domains [145, 101, 12]. The approach disentangles dense retrieval models into a Relevance Estimation Module (REM) and Domain Adaption Modules (DAMs), where the REM is delegated to the adapter and the DAMs are delegated to the backbones. Specifically, DDR additionally pretrains a backbone from BERT on *each* target domain corpus, obtaining a new domain-adapted backbone. In parallel to the additional pretraining, an adapter is fine-tuned with the original backbone (e.g., BERT), where the backbone is frozen and only the adapter is trained. The trained adapter hopefully captures general-purpose relevance estimation signals. Finally, at inference time, DDR is assembled from the adapter and the domain-adapted backbone.

We refer to our approach as mDDR in the rest of this paper, which applies DDR in the multilingual scenario: the adapter is trained on MS MARCO along with mBERT, and the



(a) mDPR init. mBERT

(b) mDPR init. mContriever

(c) mDDR and mDDR$^+$

Figure 4.1: The pipeline of (a) mDPR init. mBERT, (b) mDPR init. mContriever, and (c) mDDR and mDDR$^+$, where blocks with dashed outlines indicate modules that are frozen during training. (PT: pretrain; FT: fine-tune)

language-adapted backbones are additionally pretrained on the Mr. TYDI corpus (used in our experiments, see Section 4.1.2) in *each* language, starting from mBERT as well. That is, applying mDDR on Mr. TYDI yields 11 backbones (one per language) and one general-purpose adapter.

Note that the training of the backbones and the adapters proceed in parallel in the original DDR pipeline, thus there is no guarantee that the weights of the two align after assembly. To address this issue, we further fine-tune the adapter with the domain-adapted backbone rather than the original one, referred to as mDDR$^+$. At this stage, the 11 backbones are identical to the ones in mDDR, but now we have 11 adapters, each aligned to a different language backbone.

The pipelines of mDDR and mDDR$^+$ are illustrated in Figure 4.1, along with the pipelines of mDPR with different backbones for comparison. The implementation of mDDR and mDDR$^+$ are based on Tevatron [39] and AdapterHub [100]. Adapter fine-tuning shares the same configuration as dense retrieval training above. In MLM pretraining, the models are trained for 50k steps, with batch size 2048 and learning rate 1e-4. We set linear warm-up in the first 10k steps. The additional pretraining is conducted on a TPU v3-8 using a gradient accumulation of 16.

Cross-encoder knowledge distillation. Other than achieving multilingual transfer capabilities via directly fine-tuning the dense retrieval models themselves, previous research has shown that dense retrieval models distilled from more powerful teacher models (usually cross-encoders) yield better in-domain and out-of-domain effectiveness with the same training data. These findings mostly lie in the context of English retrieval [47, 48], which does not only benefit DPR, but also applies to other bi-encoder designs [35, 34, 114, 67]. For non-English retrieval, it has been demonstrated that SPLADE++ [34], a sparse bi-encoder model augmented with knowledge distillation, achieves competitive scores on retrieval in Persian and Russian [70].

The implementation of cross-encoder knowledge distillation is extended from Tevatron, sharing the same data pipeline as the dense retrieval model experiments. To train the cross-encoder based on XLM-R_{Large} for knowledge distillation, we use 32 batch size with learning rate 5e-6. We distill the cross-encoder into mDPR using KL-divergence [46] with batch size 512. These reported settings are based on pilot experiments on a combination of a cross-encoder selected from {XLM-R_{Large}, mT5}, a distillation method selected from {KL-divergence Loss [46], MarginMSE [47], TASB [48]}, and tested on 8 languages (all but {`en`, `ja`, `ru`}, due to computation costs). While results on each language vary, we find that KL-divergence loss achieves the highest average MRR@100 over the tested languages.

Hybrid fusion. For experiments that involve combining results from BM25 and mDPR, we follow exactly the sparse–dense hybrid approach in Mr. TYDI [149]. The final retrieval score s_{hybrid} is computed as a linear combination of the BM25 score s_{sparse} and the mDPR score s_{dense}, with α as a weighting parameter tuned on the development set.

Evaluation We report MRR@100 and Recall@100 on the test sets of each language. In our analyses, we report statistically significant differences detected using paired t-tests ($p < 0.01$). Wary of the dangers of multiple hypothesis testing, we do not apply tests indiscriminately, but rather only to answer specific research questions.

4.2 Have Model, No Data

Definition In Scenario (1), the target language $\mathcal{L} \in$ mBERT, but we have no in-domain data in the target language $\mathcal{L}$. Note that here, as well as in the following sections, we use mBERT as a convenient shorthand to refer to a multilingual pretrained transformer. Later

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
(1)	BM25 (default)	0.368	0.418	0.140	0.284	0.376	0.211	0.285	0.313	0.389	0.343	0.401	0.321
(2)	BM25 (tuned)	0.367	0.413	0.151	0.288	0.382	0.217	0.281	0.329	0.396	0.424	0.417	0.333
(3)	mDPR (NQ pFT)	0.184	0.279	0.315	0.240	0.273	0.194	0.219	0.292	0.143	0.075	0.147	0.215
(4)	mDPR (MS pFT)	0.441	0.397	0.327	0.275	0.352	0.311	0.282	0.356	0.342	0.310	0.269	0.333
(5)	mDPR (ar-mMARCO pFT)	0.499	–	0.285	–	0.388	0.297	–	0.297	–	–	–	–
(6)	mDPR (id-mMARCO pFT)	0.466	–	0.300	–	0.500	0.334	–	0.334	–	–	–	–
(7)	mDPR (ja-mMARCO pFT)	0.422	–	0.290	–	0.341	0.365	–	0.365	–	–	–	–
(8)	mDPR (ru-mMARCO pFT)	0.415	–	0.265	–	0.350	0.280	–	0.367	–	–	–	–
(9)	mDPR (all-mMARCO pFT)	0.466	0.401	0.282	0.346	0.458	0.323	0.327	0.366	0.381	0.560	0.371	**0.389**

(a) MRR@100

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
(1)	BM25 (default)	0.793	0.869	0.537	0.719	0.843	0.645	0.619	0.648	0.764	0.758	0.853	0.732
(2)	BM25 (tuned)	0.800	0.874	0.551	0.725	0.846	0.656	0.797	0.660	0.764	0.813	0.853	**0.758**
(3)	mDPR (NQ pFT)	0.493	0.703	0.757	0.601	0.653	0.546	0.545	0.667	0.414	0.190	0.409	0.548
(4)	mDPR (MS pFT)	0.797	0.784	0.754	0.647	0.736	0.732	0.617	0.743	0.634	0.782	0.595	0.711
(5)	mDPR (ar-mMARCO pFT)	0.856	–	0.718	–	0.798	0.710	–	0.710	–	–	–	–
(6)	mDPR (id-mMARCO pFT)	0.822	–	0.752	–	0.867	0.751	–	0.751	–	–	–	–
(7)	mDPR (ja-mMARCO pFT)	0.800	–	0.737	–	0.770	0.765	–	0.765	–	–	–	–
(8)	mDPR (ru-mMARCO pFT)	0.768	–	0.681	–	0.715	0.684	–	0.749	–	–	–	–
(9)	mDPR (all-mMARCO pFT)	0.803	0.806	0.677	0.722	0.803	0.688	0.616	0.722	0.665	0.893	0.740	0.740

(b) Recall@100

Table 4.2: "Have model, no data": choice of pre–fine-tuning data. MRR@100 and Recall@100 across all 11 languages and an overall average on the Mr. TYDI test set. All conditions are zero-shot; the models have not seen any training data from Mr. TYDI. The first two rows are baselines using BM25. The remaining rows are zero-shot results based on mDPR (DPR initialized with mBERT) pre–fine-tuned (pFT) on different datasets. NQ: Natural Questions; MS: the original MS MARCO dataset in English; $\mathcal{L}$-mMARCO: MS MARCO automatically translated to language $\mathcal{L}$, where $\mathcal{L} \in \{\texttt{ar}, \texttt{id}, \texttt{ja}, \texttt{ru}\}$; all-mMARCO: the entire mMARCO dataset, which contains 14 languages (including English). We underline the entries where the pre–fine-tuning dataset is in (and only in) the target language, which are also the best scores in the corresponding columns (with the exception of `ru` in terms of Recall@100).

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
MS pFT													
(T4.2–4)	mDPR init. mBERT	0.441	0.397	**0.327**	0.275	0.352	0.311	0.282	0.356	0.342	0.310	0.269	0.333
(1)	mDPR init. XLM-R_{Large}	0.393	0.429	0.303	0.298	0.413	0.336	**0.375**	0.333	0.151	0.401	0.460	0.354
(2)	mDPR init. mContriever	0.500	0.461	0.301	0.369	0.461	0.310	0.349	0.357	0.461	0.450	**0.483**	0.409
(3)	mDDR	0.369	0.371	0.285	0.334	0.440	0.280	0.333	0.265	0.484	0.186	0.361	0.337
(3′)	mDDR$^+$	0.451	0.401	0.302	0.386	**0.463**	0.319	0.360	0.346	**0.533**	0.251	0.394	0.382
all-mMARCO pFT													
(T4.2–9)	mDPR init. mBERT	0.466	0.401	0.282	0.346	0.458	0.323	0.327	0.366	0.381	**0.560**	0.371	0.389
(4)	mDPR init. XLM-R_{Large}	0.435	0.391	0.243	0.338	0.413	0.308	0.298	0.325	0.396	0.425	0.476	0.368
(5)	mDPR init. mContriever	**0.505**	**0.467**	0.275	**0.393**	0.448	**0.344**	0.349	**0.369**	0.529	0.486	0.464	**0.421**

(a) MRR@100

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
MS pFT													
(T4.2–4)	mDPR init. mBERT	0.797	0.784	**0.754**	0.647	0.736	0.732	0.617	0.743	0.634	0.782	0.595	0.711
(1)	mDPR init. XLM-R_{Large}	0.758	0.824	0.720	0.670	0.812	**0.758**	0.676	0.761	0.405	0.676	**0.875**	0.721
(2)	mDPR init. mContriever	**0.836**	**0.878**	0.727	0.785	0.821	0.713	0.706	0.745	0.766	0.868	0.859	0.791
(3)	mDDR	0.759	0.770	0.725	0.763	0.825	0.639	0.701	0.639	0.837	0.460	0.758	0.716
(3′)	mDDR$^+$	0.826	0.847	0.745	**0.804**	**0.842**	0.696	**0.708**	0.738	**0.875**	0.601	0.784	0.769
all-mMARCO pFT													
(T4.2–9)	mDPR init. mBERT	0.803	0.806	0.677	0.722	0.803	0.688	0.616	0.722	0.665	0.893	0.740	0.740
(4)	mDPR init. XLM-R_{Large}	0.757	0.806	0.616	0.721	0.777	0.673	0.590	0.682	0.706	0.846	0.862	0.731
(5)	mDPR init. mContriever	0.814	0.874	0.686	0.801	0.837	0.715	0.687	**0.764**	0.855	**0.928**	0.874	**0.803**

(b) Recall@100

Table 4.3: "Have model, no data": enhanced backbones. MRR@100 and Recall@100 of mDPR initialized from different backbones, evaluated across all 11 languages and an overall average on the Mr. TYDI test set. The backbones include mBERT, XLM-R_{Large}, mContriever, and additional MLM pretraining (mDDR and mDDR$^+$). Under the "MS pFT" block, models are pre–fine-tuned on only English MS MARCO; Under the "all-mMARCO pFT" block, models are pre–fine-tuned on all of mMARCO (14 languages). Rows (T4.2–4, T4.2–9) are identical to the rows 4 and 9 in Table 4.2. The underlined entries are the best results under the "MS pFT" block, whereas the **bold** entries are the best results under both blocks.

in this section, we describe experiments that replace mBERT with enhanced backbones; while effectiveness improves, our core findings and recommendations do not.

Recommendations Pre–fine-tune mDPR starting with a multilingual backbone (from a public checkpoint) on the mMARCO dataset. Use training data only in the target language $\mathcal{L}$ if they are available (i.e., MS MARCO that has been translated into language $\mathcal{L}$), otherwise use all available translated variants of MS MARCO, including English. Additional pretraining in the target language with MLM can be helpful in a more resource-constrained setting, but this approach forgoes the opportunity to exploit transfer effects from multiple languages in mMARCO. Independently, apply knowledge distillation from cross-encoders when possible. For retrieval, apply inference directly in the target language $\mathcal{L}$ in a zero-shot manner. □

Explanation In this section, we provide supporting experimental evidence for the above recommendations, drawn from Tables 4.2, 4.3, and 4.4. In all these tables, each row represents an experimental condition and each column shows the effectiveness (MRR@100 on top, Recall@100 on bottom) for a particular language. At the top of Table 4.2, rows (1) and (2) capture BM25 with default and tuned parameters, copied directly from the original Mr. TYDI paper [149], which are used as baselines. We present explorations from three perspectives: choice of pre–fine-tuning data (Sections 4.2.1 and 4.2.2), enhanced backbones (Section 4.2.3), and knowledge distillation from cross-encoders (Section 4.2.5).

4.2.1 English Pre–Fine-Tuning Data

In Table 4.2, row (3) follows the experimental setup of mDPR in Mr. TYDI [149], where the model is fine-tuned on Natural Questions (NQ) [61]. However, as mentioned in Section 4.1.4, we share the parameters of the query and passage encoders rather than train two separate encoders, as in the original DPR paper [58]. Row (3) represents a zero-shot baseline using a multilingual dense retrieval model that has not seen any annotated examples from Mr. TYDI.

Building on this, the obvious improvement is to follow the same procedure as row (3), but using a different dataset. In row (4), we fine-tune mDPR on the MS MARCO passage ranking dataset [10], which is larger than NQ [61], and then apply inference for all languages in a zero-shot manner. As discussed in Section 4.1.1, we refer to the approach in rows (3) and (4) as pre–fine-tuning (or pFT for short) [152], to distinguish additional

fine-tuning that we can further perform downstream using labeled data drawn from the same distribution as the evaluation data (discussed later).

We clearly see that pre–fine-tuning on MS MARCO, row (4), consistently beats results on NQ [61], row (3). These differences are statistically significant across all languages and support our recommendation that if one doesn't have data in the target language $\mathcal{L}$, then pre–fine-tuning on a large existing "out-of-domain" dataset is the best course of action. This strategy takes advantage of cross-lingual transfer effects, and is consistent with previous work. The dataset used for pre–fine-tuning *does* matter, and here we find that MS MARCO is better than NQ.

These results also show that the domain of a dataset may not be the only or even the most important factor in its transfer ability: While results with NQ pre–fine-tuning are worse than with MS MARCO, its domain is "closer" to Mr. TYDI—both contain relatively well-formed questions posed against Wikipedia—whereas MS MARCO contains noisier and (sometimes) ill-formed questions, annotated against snippets extracted from webpages. Other factors, for example, the dataset size, the query length, the document length, etc., may also play important roles.

4.2.2 Multilingual Pre–Fine-Tuning Data

While we observe clear transfer effects from English pre–fine-tuning to other languages in downstream retrieval tasks, it is unclear whether using machine-translated training data in the target language achieves better zero-shot effectiveness. We explore this question using mMARCO [13], where 5 out of the 14 languages overlap with Mr. TYDI (`ar`, `en`, `id`, `ja`, and `ru`). Obviously, it would be ideal to have large manually annotated datasets like MS MARCO in multiple non-English languages, but this is obviously impractical due to the substantial costs of assembling such a dataset. The point of these experiments is to determine whether automatically machine-translated versions of MS MARCO can nevertheless be beneficial.

Single-language pre–fine-tuning data In Table 4.2, rows (5–8), we train the model using mMARCO in the languages {`ar`, `id`, `ja`, `ru`} with exactly the same pre–fine-tuning regime as row (4). That is, rows (4–8) can be viewed as a group where they only differ in the pre–fine-tuning language. These languages are chosen because they are the overlapping languages between Mr. TYDI and mMARCO. We then evaluate the models on the same five languages, focusing on the effectiveness differences when the pre–fine-tuning data are in the target language versus when it is not.

Comparing the results across rows (4–8), we see that on all five languages, the zero-shot MRR@100 scores are the highest when the model is pre–fine-tuned on the target language (the underlined entries in Table 4.2). Additionally, the scores vary to a large extent when pre–fine-tuning in different languages, showing that the zero-shot effectiveness is sensitive to the dataset language. These results affirm the findings of mMARCO [13] for dense retrieval, demonstrating that pre–fine-tuning a dense retrieval model on target-language training data can improve zero-shot effectiveness even when the data are *automatically* translated, despite the fact that the data may be of lower quality due to translation errors (holding size and domain constant).

Multi-language pre–fine-tuning data Knowing that pre–fine-tuning on training data in the target language is consistently better (even if automatically translated), the next question is: How would additional non-target languages in the pre–fine-tuning data affect the results?

This is investigated in row (9), where we pre–fine-tune mDPR on all 14 languages provided by mMARCO (including English). Interestingly, based on observations on the five languages where Mr. TYDI and mMARCO overlap (`ar`, `en`, `id`, `ja`, `ru`), pre–fine-tuning in *only* the target language itself shows the best results. That is, pre–fine-tuning in more than just the target language *hurts* zero-shot effectiveness, which echoes the "curse of multilinguality" [23] even though we are only using 14 languages. This is shown by comparing the underlined scores in rows (4–8) to the score in the same column in row (9). Take Arabic as an example: while mDPR pre–fine-tuned on the `ar`-mMARCO training data achieves MRR@100 0.499, shown in row (5), column `ar`, it only achieves MRR@100 0.466 when pre–fine-tuned on all languages, shown in row (9), column `ar`. The difference is significant for 4 out of 5 languages (`ar`, `en`, `id`, `ja`) and the scores are only on par for `ru` (0.367 vs. 0.366).

However, experimental results suggest that it is still better to use multiple languages for pre–fine-tuning, compared to just using a single *non-target* language (including pre–fine-tuning *only* on the original English version of MS MARCO). This conclusion is based on the observation that each entry in row (9) is higher than most of the non-underlined corresponding entries in rows (4-8). For example, the MRR@100 of `ar` in row (9) is 0.466, which is higher than or similar to the corresponding scores in rows (4, 6, 7, 8). The MRR@100 of `fi` in row (9) is 0.346, which is higher than the score in row (4).

Synthesizing these results, we arrive at the following recommendations: For searching in target language $\mathcal{L}$, pre–fine-tune only on MS MARCO translated into $\mathcal{L}$ if it exists. Otherwise, pre–fine-tuning on all available translated versions of MS MARCO (including the

original English version) appears to beat just pre–fine-tuning on MS MARCO translated into a single non-target language.

4.2.3 Enhanced Backbones

Another approach to achieve better generalization capacity is to start from better backbones. These results are shown in Table 4.3, where models are pre–fine-tuned under two conditions: on English MS MARCO only in the first block (MS pFT) and on all languages in mMARCO in the second block (all-mMARCO pFT). We compare three approaches to backbone enhancement:

General multilingual backbone This is represented by XLM-R_{Large} [23], with results reported in rows (1, 4). Comparing row (1) to row (T4.2–4), XLM-R_{Large} shows better overall zero-shot transfer ability than mBERT (average MRR@100 0.354 vs. 0.333), although at the cost of twice the model size and 5× training time. Additionally, we find its advantage lost when pre–fine-tuning using multilingual data: When pre–fine-tuning the model on all-mMARCO rather than MS MARCO, although the average score improves, XLM-R_{Large} is now surpassed by mBERT, as shown in rows (4, T4.2–9), average MRR@100 0.368 vs. 0.389. When comparing improvements per language, row (1) vs. row (4), we see that 5 (`bn`, `id`, `ja`, `ko`, `ru`) out of the 10 languages (excluding `en`) show worse or similar zero-shot MRR@100. This can be explained by the fact that XLM-R_{Large} has already "absorbed" a large amount of multilingual knowledge (more so than mBERT), and thus benefits less from additional machine-translated data in the pre–fine-tuning step.

Additional pretraining on the target task This is represented by mContriever [51], which started from mBERT and was then additionally pretrained on unsupervised synthetic data for information retrieval tasks using contrastive learning. Results with mContriever are reported in rows (2, 5). Under both MS pFT and all-mMARCO pFT conditions, mContriever outperforms mBERT and XLM-R_{Large} by a large margin, which affirms its zero-shot transfer ability as reported in the original paper [51]. Similar to observations with XLM-R_{Large}, its improvement over mBERT is more prominent when fine-tuned on MS MARCO (0.409 vs. 0.333, improvement of 23%) compared to all-mMARCO (0.421 vs. 0.389, improvement of only 8%).

Based on these experiments, our recommendation that pre–fine-tuning in multiple languages is better than a single non-target language holds as the backbone changes. This is supported by the observation that MRR@100 in rows (4, 5) are generally higher than

the corresponding conditions in rows (1, 2) for the non-English entries. We set aside `en` as a special case since rows (1, 2) are pre–fine-tuned on English MS MARCO. That is, the models are pre–fine-tuned in the target language when evaluated on English.

Additional pretraining in the target language This is represented by mDDR and mDDR$^+$ [145], where the models also start from mBERT, but with additional pretraining on target language corpora with MLM loss (see Section 4.1.4 for details). As mDDR adapts mBERT to each language, we do not further pre–fine-tune the model with mMARCO, which "dilutes" the model for non-target languages. Results of mDDR are reported in row (3), where we observe that the strategy of continuing to pretrain on target corpora does not consistently improve the out-of-language generalization capacity: Compared to row (T4.2–4), we observe improvements with mDDR, row (3), on some of the languages (`fi`, `id`, `ko`, `sw`, `th`), but effectiveness drops for the other languages, some by a large margin. Overall, mDDR achieves similar effectiveness as mDPR initialized from mBERT, differing only by 0.004 in average MRR@100.

However, effectiveness is greatly improved after additional pre–fine-tuning on the adapter. Comparing row (3′) to row (3), we see that MRR@100 improves for all languages and the average increases from 0.337 to 0.382. These results affirm our hypothesis on the misaligned weights between the language-adapted backbone and the adapter. Additionally, average MRR@100 is now on par with mDPR pFTed on all mMARCO data, initialized from mBERT, i.e., row (3′) vs. row (4); the scores are close for most of the languages, except `sw` and `te`. These results suggest that additional MLM pretraining and pFT on data in multiple languages serve similar functions; see discussions in the next section.

4.2.4 Discussion: Relevance Transfer and Language Adaptation

Taking a step back and reviewing the above zero-shot approaches, we can understand them as attempts to "approximate" in-language, in-domain fine-tuning data in a cheap manner from two perspectives: (1) query–document relevance, e.g., from human-labeled datasets such as MS MARCO and from synthetic relevance data in mContriever, and (2) target-language knowledge, e.g., from machine translated data and from additional MLM pretraining on the target-language corpus. Based on this perspective, the above empirical results suggest that:

1. Regarding query–document relevance: IC-style additional pretraining on synthetic relevance data provides gains beyond English MS MARCO, since we observe that mContriever + MS MARCO > mBERT + MS MARCO.

2. Regarding target-language knowledge: additional MLM pretraining yields similar results as pre–fine-tuning on a human-labeled retrieval dataset that has been translated into multiple languages, but both appear to be worse than target-language translations of the same dataset (in all cases, we consider imperfect machine translations). This is seen in mBERT + target-$\mathcal{L}$-mMARCO > mBERT + all-mMARCO $\approx$ mDDR$^+$. In other words, machine translation should be used in a *targeted* manner.

While this summary provides a better understanding of the comparative effectiveness under each experimental configuration, it is also worth discussing the cost of each approach, especially with respect to point (2) above: While mMARCO provides out-of-the-box translations for 13 languages, it is an existing resource from our perspective and hence "cheap" for us to exploit. However, in the general case, for a new target language $\mathcal{L}$, performing machine translation on a similar scale remains costly and challenging (not to mention that a good translation system for $\mathcal{L}$ might not even be available). Similarly, the pretraining of mContriever depends on CCNet [134] and requires careful preprocessing of web data. Additionally, based on our experience, effective IC-based pretraining is non-trivial in terms of properly setting hyperparameters and other experimental configurations to achieve the desired gains. On the contrary, while additional pretraining with MLM does not yield the best results, it requires only an *unlabeled* corpus in the target language. Thus, we suggest that additional MLM pretraining in the target language may be an effective substitute when the translation of pre–fine-tuning data is not practical.

4.2.5 Knowledge Distillation

The above experiments compare different approaches to improving out-of-language generalization using only the mDPR model (including different backbones). We now discuss the case where we have access to cross-encoders, which have been shown to be more robust in out-of-domain tasks [47].

Results are shown in Table 4.4. The top row presents the scores of monoXLM-R$_{\text{Large}}$, a variant of the monoBERT cross-encoder [93] initialized from XLM-R$_{\text{Large}}$. The model reranks the top-100 candidates returned from BM25, i.e., row (2) in Table 4.2. The second block, rows (T4.2–4, T4.3–2), shows the baselines of mDPR initialized from different backbones (mBERT and mContriever) and then pre–fine-tuned on English MS MARCO, copied from row (4) in Table 4.2 and row (2) in Table 4.3, respectively. Comparing row (A) to the two baselines, we see that monoXLM-R$_{\text{Large}}$ MRR@100 exceeds mDPR by a large margin, which confirms the effectiveness of cross-encoders in the out-of-language scenario.

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
(A)	monoXLM-R_{Large}	0.651	0.595	0.366	0.536	0.635	0.476	0.451	0.532	0.632	0.787	0.686	0.577
(T4.2–4)	mDPR init. mBERT	0.441	0.397	0.327	0.275	0.352	0.311	0.282	0.356	0.342	0.310	0.269	0.333
(T4.3–2)	mDPR init. mContriever	0.500	0.461	0.301	0.369	0.461	0.310	0.349	0.357	0.461	0.450	0.483	0.409
(1)	Distilled mDPR init. mBERT	0.503	0.432	0.304	0.399	0.463	0.338	0.339	0.398	0.436	0.389	0.340	0.395
(2)	Distilled mDPR init. mContriever	0.554	0.488	0.292	0.413	0.517	0.275	0.405	0.333	0.506	0.639	0.505	**0.448**

(a) MRR@100

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
(A)	monoXLM-R_{Large}	0.793	0.870	0.537	0.720	0.843	0.643	0.619	0.654	0.764	0.897	0.853	0.745
(T4.2–4)	mDPR init. mBERT	0.797	0.784	0.754	0.647	0.736	0.732	0.617	0.743	0.634	0.782	0.595	0.711
(T4.3–2)	mDPR init. mContriever	0.836	0.878	0.727	0.785	0.821	0.713	0.706	0.745	0.766	0.868	0.859	0.791
(1)	Distilled mDPR init. mBERT	0.830	0.860	0.655	0.770	0.815	0.733	0.687	0.765	0.737	0.798	0.700	0.759
(2)	Distilled mDPR init. mContriever	0.864	0.896	0.705	0.840	0.864	0.665	0.741	0.695	0.874	0.961	0.896	**0.818**

(b) Recall@100

Table 4.4: "Have model, no data": knowledge distillation. MRR@100 and Recall@100 of the cross-encoder along with mDPR that is pre–fine-tuned or distilled from the cross-encoder, evaluated across all 11 languages and an overall average on the Mr. TYDI test set. Models are pre–fine-tuned on English MS MARCO in all experiments. The first row denotes the monoXLM-R_{Large} cross-encoder and the second block captures baselines without knowledge distillation, copied from previous tables. Best *distilled* results are in **bold**.

The third block of the table presents our knowledge distillation results: rows (1, 2) report mDPR initialized from mBERT or mContriever, distilled from the cross-encoder in row (A). Compared to the two baselines above, these models consume the same amount of resources during fine-tuning (i.e., backbone and pre–fine-tuning data), and only differ in whether the model is fine-tuned on the data directly or distilled from the cross-encoder.

The results are positive. When initialized from either backbone, knowledge distillation yields big improvements: average MRR@100 from 0.333 to 0.395 with mBERT, row (T4.2–4) vs. row (1), and average MRR@100 from 0.409 to 0.448 with mContriever, row (T4.3–2) vs. row (2). Furthermore, a better initialization checkpoint boosts the knowledge distillation results: Starting from mContriever achieves 0.053 points higher average MRR@100 compared to mBERT, shown in row (2) vs. row (1), yielding the best zero-shot average MRR@100 on Mr. TYDI among the settings we have explored so far. Nonetheless, a large disparity in effectiveness persists between the distilled mDPR models and their cross-encoder teachers, suggesting the need for additional explorations in knowledge distillation techniques under multilingual scenarios.

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
(T4.2–4)	mDPR (MS pFT)	0.441	0.397	0.327	0.275	0.352	0.311	0.282	0.356	0.342	0.310	0.269	0.333
(T4.2–9)	mDPR (all-mMARCO pFT)	0.466	0.401	0.282	0.346	0.458	0.323	0.327	0.366	0.381	0.560	0.371	0.389
(1)	mDPR (in-lang FT)	0.678	0.638	0.418	0.516	0.544	0.447	0.383	0.448	0.580	0.860	0.597	0.555
(2)	mDPR (all FT)	0.695	0.659	0.476	0.550	0.565	0.496	0.453	0.515	0.633	0.891	0.607	0.594
(3)	mDPR (in-script FT)	–	–	0.444	0.535	0.560	–	–	–	0.622	–	–	–
(4)	mDPR (out-script FT)	–	–	0.457	0.543	0.573	–	–	–	0.624	–	–	
(5)	mDPR (MS pFT + in-lang FT)	0.691	0.651	0.489	0.551	0.562	0.488	0.453	0.485	0.640	0.876	0.619	0.591
(6)	mDPR (MS pFT + all FT)	0.695	0.623	0.492	0.560	0.579	0.501	0.487	0.517	0.644	0.891	0.617	0.600
(7)	mDPR (MS pFT + in-script FT)	–	–	0.473	0.555	0.563	–	–	–	0.635	–	–	–
(8)	mDPR (MS pFT + out-script FT)	–	–	0.476	0.563	0.565	–	–	–	0.644	–	–	–
(9)	mDPR (mMARCO pFT + in-lang FT)	0.669	0.648	0.441	0.524	0.535	0.438	0.421	0.458	0.584	0.860	0.599	0.562
(10)	mDPR (mMARCO pFT + all FT)	0.660	0.671	0.456	0.514	0.548	0.460	0.466	0.488	0.604	0.865	0.581	0.574
(11)	mDDR$^+$ (in-lang FT)	0.677	0.585	0.395	0.537	0.538	0.411	0.375	0.432	0.621	0.824	0.567	0.542
(12)	BM25 + row (6)	0.714	0.702	0.520	0.590	0.634	0.558	0.523	0.590	0.623	0.845	0.697	**0.636**

(a) MRR@100

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
(T4.2–4)	mDPR (MS pFT)	0.797	0.784	0.754	0.647	0.736	0.732	0.617	0.743	0.634	0.782	0.595	0.711
(T4.2–9)	mDPR (all-mMARCO pFT)	0.803	0.806	0.677	0.722	0.803	0.688	0.616	0.722	0.665	0.893	0.740	0.740
(1)	mDPR (in-lang FT)	0.889	0.896	0.797	0.837	0.873	0.791	0.749	0.800	0.864	0.954	0.866	0.847
(2)	mDPR (all FT)	0.894	0.937	0.839	0.846	0.867	0.811	0.771	0.819	0.893	0.969	0.866	0.865
(3)	mDPR (in-script FT)	–	–	0.812	0.842	0.855	–	—	–	0.877	–	–	–
(4)	mDPR (out-script FT)	–	–	0.817	0.543	0.862	–	–	–	0.876	–	–	
(5)	mDPR (MS pFT + in-lang FT)	0.891	0.914	0.837	0.852	0.863	0.828	0.799	0.819	0.890	0.967	0.886	0.868
(6)	mDPR (MS pFT + all FT)	0.900	0.955	0.841	0.856	0.860	0.813	0.785	0.843	0.876	0.966	0.883	0.871
(7)	mDPR (MS pFT + in-script FT)	0.847	0.865	0.837	0.849	0.874	–	0.740	–	0.876	0.876	0.629	–
(8)	mDPR (MS pFT + out-script FT)	–	–	0.824	0.857	0.869	–	–	–	0.873	–	–	–
(9)	mDPR (mMARCO pFT + in-lang FT)	0.890	0.941	0.810	0.823	0.846	0.785	0.749	0.809	0.856	0.962	0.870	0.849
(10)	mDPR (mMARCO pFT + all FT)	0.878	0.914	0.815	0.827	0.842	0.783	0.770	0.804	0.872	0.958	0.843	0.846
(11)	mDDR$^+$ (in-lang FT)	0.904	0.937	0.830	0.884	0.881	0.806	0.742	0.842	0.919	0.964	0.895	0.873
(12)	BM25 + row (6)	0.932	0.946	0.857	0.909	0.948	0.883	0.853	0.898	0.903	0.982	0.946	**0.916**

(b) Recall@100

Table 4.5: "Have model and data": multilingual pre–fine-tuning and fine-tuning. MRR@100 and Recall@100 of mDPR evaluated across all 11 languages and an overall average on the Mr. TYDI test set. "MS pFT": pre–fine-tuning on English MS MARCO; "all-mMARCO pFT": pre–fine-tuning on all of mMARCO including `en`; "in-lang FT": fine-tuning on the Mr. TYDI training set from only the target language; "all FT": fine-tuning on Mr. TYDI training data from all languages; "in/out-script": fine-tuning on training data from the target language *and* languages that are in the same/different scripts; for example, if the target language is `en`, the "in-script" languages would be the target language (`en`) and all the other Latin-script languages (`fi`, `id`, `sw`) and the "out-script" languages would be the target language (`en`) and all the non-Latin-script languages (`ar`, `bn`, `ja`, `ko`, `ru`, `te`, `th`). Best results are in **bold**.

4.3 Have Model and Data in Target Language: Multilingual BERT

Definition In Scenario (2a), the target language $\mathcal{L} \in$ mBERT and we have in-domain training data (manually labeled relevance judgments) in $\mathcal{L}$.

Recommendations Start with pre–fine-tuning mDPR using English MS MARCO data, then continue fine-tuning on relevance judgments in $\mathcal{L}$. Augmenting the fine-tuning data with relevance judgments from other languages generally helps as well, although in some cases the results are mixed and the overall differences are small. Somewhat surprisingly, given in-domain training data in the target language, pre–fine-tuning on (only) English MS MARCO beats pre–fine-tuning on all languages in mMARCO. Additionally, training the model with in-language hard negative examples provides consistent gains. □

4.3.1 English/Multilingual Pre–Fine-Tuning Data

Table 4.5 focuses on scenario (2a) specifically: rows (T4.2–4, T4.2–9) are zero-shot results copied from rows (4, 9) in Table 4.2. Rows (1–4) start with the raw mBERT checkpoint without any pre–fine-tuning, and then fine-tune on different groups of Mr. TYDI training data. This is contrasted against rows (5–8), which start from the checkpoint pre–fine-tuned on English MS MARCO, row (T4.2–4). Two groups, rows (1, 2) and rows (5, 6), are additionally contrasted against rows (9, 10), which start from the checkpoint pre–fine-tuned on all-mMARCO, row (T4.2–9). Specifically, rows (1, 5, 9), tagged with "in-lang", represent models fine-tuned using the Mr. TYDI training data only in the target language $\mathcal{L}$. These results answer the following questions:

(i) Do we still need pre–fine-tuning if we have training data in the target language $\mathcal{L}$?

(ii) If so, does the model still benefit from additional multilingual pretraining and pre–fine-tuning as in the "have model, no data" scenario?

The answer to the first question appears to be, unequivocally, *yes*, as row (5) beats row (1) across all languages; these differences are statistically significant in 6 out of the 11 languages. This means that, no matter the target language, models benefit from cross-lingual transfer from English in the pre–fine-tuning step. Retrieval in Arabic, Bengali, etc. still

benefits from cross-lingual transfer with English data—despite the fact that the languages are so different (down to the written script).

However, the answer to the second question appears to be *no*. Comparing row (9) to row (5), pre–fine-tuning on all of mMARCO produces worse results than pre–fine-tuning on (only) English MS MARCO, across all languages. These results seem to indicate that the multilingual information learned from the pre–fine-tuning step is secondary when downstream in-language data are available, and that multilingual pre–fine-tuning does not provide a better initialization point for subsequent fine-tuning, even though it shows better zero-shot effectiveness. Another way to explain this finding is that if "high quality" in-domain, in-language training data are available, adding lower quality data (e.g., automatically translated, out-of-domain) might actually reduce effectiveness.

Similar results are observed when the model is additionally pretrained using MLM on the target language corpus. As presented in row (11), starting from the mDDR$^+$ checkpoint, row (3′) in Table 4.3, we continue to fine-tune the adapters on the in-language Mr. TYDI training data. We see that the advantage of language adaptation from mDDR$^+$ appears to completely vanish after in-language fine-tuning: In row (11), starting from mDDR$^+$ yields average MRR@100 0.542, which is lower than the score of directly fine-tuning mDPR from mBERT, average MRR@100 0.555, shown in row (1).

We thus conclude that knowledge gained from multiple languages prior to the fine-tuning stage, under the framework in Section 4.2.4, does not appear to provide additional benefits when in-language, in-domain fine-tuning data are available.

4.3.2 Multilingual Fine-Tuning Data

A closely related question: Can we further leverage cross-lingual transfer effects at the *fine-tuning* step? In Mr. TYDI, we have data for all languages, and so we can experiment with MS MARCO or mMARCO pFT and further fine-tune on *all* available data, in all languages. This answers the following research question: If we have in-domain relevance judgments in languages other than the target language, are such data helpful?

The outcome of these experiments is presented in the remaining rows of Table 4.5. Results starting from the MS MARCO pFT checkpoint are shown in row (6). We see that using data from all languages beats or performs similarly to using data only from the target language, row (5), in all cases except Bengali; the gains are only statistically significant for Russian, though. Results starting from mMARCO pFT are similar. We observe that row (10), where the model is fine-tuned on all languages, outperforms row (9) on 8 out of 11 languages. These results suggest that we can benefit from cross-lingual transfer in the

fine-tuning step as well as the pre–fine-tuning step. However, the conclusions are equivocal: Augmenting in-domain fine-tuning data in target language $\mathcal{L}$ with in-domain data from other non-target languages generally helps, but not always. Nevertheless, the magnitudes of the observed differences are generally small.

Next, we attempt to better understand cross-lingual transfer effects from different non-target languages. For example, would models benefit more from languages sharing the same script with the target language? Rows (3, 4, 7, 8) answer this question. Row (3) reports the results of the variant where we only fine-tune in languages written in the Latin script (there are no other language pairs that share the same script), and row (4) reports results that only fine-tune on the (Latin-script) target language itself and all other non-Latin-script languages (for example, the fine-tuning languages are all but `fi`, `id`, `sw` when the target language is `en`). For rows (7–8), we apply the same fine-tuning treatment as rows (3–4), but start from the checkpoint with pre–fine-tuning rather than the raw mBERT checkpoint.

We observe that scores in rows (3–4) are always higher than in row (1), which indicates that additional training data in both in-script and out-script languages are generally beneficial when not pre–fine-tuning. Comparing those two rows to row (2), we observe that using all available data additionally helps. However, this trend does not hold when pre–fine-tuning is added: the scores in rows (7–8) are quite close to the scores in row (5).

The summary here is that the models all appear to benefit from fine-tuning on all available data, although the effects are stronger without pre–fine-tuning. This does make sense since with pre–fine-tuning, we are already obtaining maximal cross-language transfer effects from a large dataset (albeit from English only). We see that in terms of MRR@100, row (6) beats row (5) for 9 of the 11 languages and has a higher overall average. This supports our overall recommendation of pre–fine-tuning on English MS MARCO and then further fine-tuning on all available data.

Putting everything together, we combine the results of the best overall multilingual dense retrieval model (pFT on English MS MARCO, further fine-tuning on all data) with bag-of-words BM25; these results are shown in Table 4.5, row (12). We can see, consistent with previous work, that dense and sparse retrieval models provide complementary signals, and that their combination yields effectiveness that is better than each model alone.

4.3.3 Enhanced Backbones and Hard Negatives

Results in Section 4.2 show that mDPR initialized from mContriever performs the best in the zero-shot condition. We thus take it as a starting point and compare the approach

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
(1)	mDPR init. mBERT	0.660	0.671	0.456	0.514	0.548	0.460	0.466	0.488	0.604	0.865	0.581	0.574
(2)	mDPR init. mBERT (HN)	0.670	0.616	0.529	0.519	0.547	0.515	0.465	0.545	0.607	0.883	0.595	0.590
(3)	mDPR init. mContriever	0.663	0.663	0.426	0.516	0.538	0.451	0.482	0.468	0.666	0.884	0.633	0.581
(4)	mDPR init. mContriever (HN)	0.692	0.677	0.518	0.534	0.551	0.516	0.512	0.545	0.661	0.878	0.659	**0.613**

(a) MRR@100

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Avg**
(1)	mDPR init. mBERT	0.878	0.914	0.815	0.827	0.842	0.783	0.770	0.804	0.872	0.958	0.843	0.846
(2)	mDPR init. mBERT (HN)	0.879	0.896	0.881	0.827	0.845	0.819	0.786	0.839	0.866	0.972	0.855	0.861
(3)	mDPR init. mContriever	0.884	0.928	0.800	0.846	0.844	0.774	0.794	0.806	0.897	0.961	0.889	0.857
(4)	mDPR init. mContriever (HN)	0.886	0.955	0.878	0.850	0.880	0.845	0.830	0.869	0.912	0.975	0.914	**0.890**

(b) Recall@100

Table 4.6: "Have model and data": enhanced backbone and hard negatives. MRR@100 and Recall@100 of mDPR initialized from mBERT and mContriever, evaluated across all 11 languages and an overall average on the Mr. TYDI test set. All experiments are first pre–fine-tuned on the entire mMARCO dataset, then on all Mr. TYDI training data. Rows marked with HN use the hard negative training data on Mr. TYDI prepared from the model of row (5) in Table 4.3 (mDPR init. mContriever + all-mMARCO pFT). Best results are in **bold**.

with mBERT in the current scenario, where in-language, in-domain data are available. Results are presented in Table 4.6. All these experiments start from the corresponding all-mMARCO pFT checkpoint in Table 4.3, and then are fine-tuned on training data in all Mr. TYDI languages.

We additionally investigate the effects of fine-tuning with hard negatives, which has been found to be helpful on English datasets, but has not been explored in multilingual retrieval. In rows (2, 4), where the experiments are marked with HN, the negative training examples are prepared from the top-30 results using the model in row (5) in Table 4.3 (DPR init. mContriever + all-mMARCO pFT), rather than from BM25 (the rest of the experiments in this paper).

In terms of the average MRR@100 scores reported in rows (1, 3), which do not use hard negatives, mContriever continues to outperform mBERT. However, looking at the effectiveness in each language, mContriever is only noticeably better on 4 out of 11 languages (`ko`, `sw`, `te`, `th`) and this is only significant on `sw` and `th`, whereas mBERT appears to outperform mContriever on five (`bn`, `en`, `id`, `ja`, `ru`). The two models are on par with each other for the other two languages (`ar`, `fi`). Nevertheless, when fine-tuned on the hard negatives, as shown in rows (2, 4), mContriever equals or outperforms mBERT on

9 languages (all but `en` and `te`), among which three are significant (`ko`, `sw`, `th`). These results suggest that mContriever can better take advantage of the hard negative training data compared to mBERT. Hard negatives increase MRR@100 by 6% with mContriever, comparing row (3) to row (4), while only 3% with mBERT, comparing row (1) to row (2). Additionally, hard negatives improve MRR@100 significantly on five languages (`ar`, `en`, `ja`, `ru`, `th`) with mContriever, while the improvement is only significant on two languages (`en`, `ru`) with mBERT.

To conclude, we find that the advantage of mContriever over mBERT still persists when in-language, in-domain fine-tuning data are available, and that the advantage is more prominent when hard negative examples are exploited. This leads to our recommendation to use hard negative mining techniques if possible.

4.4 Have Model and Data in Non-Target Language: Multilingual BERT

Definition In Scenario (2b), the target language $\mathcal{L} \in$ mBERT but we have no data in language $\mathcal{L}$. However, we have data in another language $\mathcal{K}$ and $\mathcal{K} \in$ mBERT. Language $\mathcal{K}$ may not be linguistically related to language $\mathcal{L}$ and may not even be in the same script.

Recommendations Start with pre–fine-tuning mDPR using English MS MARCO data, and then continue fine-tuning on relevance judgments in language $\mathcal{K}$. That is, models may benefit from cross-lingual transfer, surprisingly, across different language families and even different scripts! □

Explanation The results of a matrix experiment where we consider all combinations of $\mathcal{L}$ and $\mathcal{K}$ for all languages in Mr. TYDI are shown in Table 4.7. On the top, we fine-tune mDPR directly from the mBERT checkpoint, and on the bottom, we fine-tune an mDPR model that has already been pre–fine-tuned on MS MARCO. We report MRR@100: each row represents the source language of the fine-tuning data and each column represents the target language. The diagonal corresponds to the "in-language" condition described in Section 4.3. However, as described in Section 4.1.3, we use the same amount of training data for all languages (exactly 3,300 queries), except for Bengali, Korean, and Swahili, which have fewer training queries and therefore we use all available data. Due to this difference, the results in the diagonals differ from the results in Table 4.5.

	ar	bn	en	fi	id	ja	ko	ru	sw	te	th
ar	0.629	0.560	0.373	0.455	0.483	0.367	0.384	0.451	0.460	0.506	0.387
bn	0.448	0.638	0.309	0.372	0.383	0.314	0.326	0.325	0.433	0.532	0.341
en	0.439	0.389	0.436	0.381	0.402	0.296	0.314	0.363	0.397	0.289	0.218
fi	0.524	0.436	0.369	0.484	0.471	0.325	0.358	0.420	0.470	0.619	0.295
id	0.486	0.490	0.375	0.415	0.524	0.326	0.353	0.372	0.442	0.366	0.221
ja	0.480	0.573	0.328	0.391	0.395	0.471	0.382	0.401	0.441	0.615	0.345
ko	0.421	0.464	0.315	0.373	0.377	0.309	0.383	0.330	0.421	0.300	0.181
ru	0.476	0.524	0.337	0.410	0.400	0.364	0.311	0.419	0.424	0.531	0.243
sw	0.422	0.397	0.251	0.324	0.367	0.258	0.304	0.317	0.580	0.276	0.228
te	0.512	0.587	0.331	0.399	0.404	0.398	0.368	0.380	0.421	0.861	0.422
th	0.505	0.554	0.319	0.389	0.410	0.361	0.323	0.416	0.411	0.648	0.588

(a) mDPR initialized from mBERT

	ar	bn	en	fi	id	ja	ko	ru	sw	te	th
ar	0.663	0.608	0.450	0.450	0.508	0.432	0.392	0.509	0.425	0.613	0.380
bn	0.549	0.645	0.432	0.427	0.468	0.432	0.401	0.435	0.438	0.644	0.417
en	0.469	0.412	0.480	0.292	0.319	0.368	0.344	0.376	0.302	0.283	0.245
fi	0.542	0.501	0.472	0.524	0.495	0.396	0.409	0.477	0.441	0.463	0.355
id	0.551	0.538	0.432	0.441	0.561	0.385	0.396	0.443	0.430	0.496	0.349
ja	0.531	0.500	0.374	0.367	0.405	0.483	0.396	0.453	0.385	0.429	0.375
ko	0.515	0.450	0.430	0.379	0.428	0.426	0.453	0.499	0.413	0.404	0.336
ru	0.570	0.521	0.448	0.437	0.488	0.427	0.400	0.499	0.421	0.510	0.331
sw	0.498	0.470	0.388	0.409	0.440	0.365	0.353	0.428	0.635	0.494	0.331
te	0.575	0.583	0.426	0.390	0.433	0.449	0.413	0.480	0.432	0.867	0.436
th	0.587	0.530	0.446	0.410	0.490	0.427	0.411	0.480	0.428	0.605	0.611

(b) mDPR pre–fine-tuned on MS MARCO

Table 4.7: Matrix experiments training mDPR on the language denoted in the row, evaluating on the language in the column, reporting MRR@100. Fine-tuning is performed on the mBERT backbone directly (top) and after pre–fine-tuning on MS MARCO (bottom). For `bn`, `ko`, and `sw`, we use all available training data; for the remaining languages, we downsample to 3,300 training queries. Cell values are highlighted *column-wise*: the color gradient ranges from **blue** (high values) to **orange** (low values). Note that due to data downsampling, the diagonal results are not identical to results in Table 4.5.

Let us consider the starting point of mDPR with MS MARCO pre–fine-tuning. The research question is: We have training data in language $\mathcal{K}$, but it is not in our target language $\mathcal{L}$. Should we fine-tune on it anyway? That answer, surprisingly, is *yes*. These results are presented in Table 4.8, which shows the element-wise difference between MS pFT + FT vs. MS pFT (zero-shot). For example, mDPR (MS pFT) obtains 0.269 on Thai in a zero-shot setting, as seen in Table 4.2, row (4). However, if we fine-tune further on

	ar	bn	en	fi	id	ja	ko	ru	sw	te	th
ar	0.221	0.211	0.123	0.175	0.156	0.122	0.110	0.152	0.083	0.303	0.111
bn	0.108	0.248	0.105	0.152	0.116	0.121	0.119	0.079	0.096	0.334	0.147
en	0.027	0.015	0.153	0.017	-0.033	0.058	0.062	0.020	-0.040	-0.027	-0.025
fi	0.101	0.105	0.145	0.249	0.143	0.085	0.127	0.121	0.099	0.153	0.086
id	0.110	0.141	0.105	0.166	0.209	0.074	0.114	0.086	0.088	0.186	0.080
ja	0.090	0.103	0.047	0.092	0.053	0.172	0.114	0.097	0.043	0.119	0.105
ko	0.074	0.053	0.103	0.104	0.076	0.115	0.171	0.143	0.071	0.094	0.067
ru	0.128	0.124	0.121	0.162	0.136	0.117	0.118	0.143	0.079	0.200	0.062
sw	0.057	0.073	0.061	0.134	0.088	0.054	0.071	0.072	0.293	0.183	0.061
te	0.133	0.187	0.099	0.115	0.081	0.139	0.131	0.124	0.090	0.557	0.167
th	0.145	0.133	0.119	0.135	0.138	0.116	0.129	0.124	0.086	0.295	0.342

Table 4.8: Element-wise difference between MS pFT + FT and MS pFT (zero-shot), answering the question: "Should we always fine-tune on non-target language data?" Answer is *yes*, with the exception that further fine-tuning on English data for retrieval in non-English languages might result in overfitting. Positive values are highlighted in **blue** and negative values in **orange**, with saturation proportional to magnitude.

Telugu data, we can achieve 0.436 on Thai (seen in Table 4.7, bottom, `te` row, `th` column). This represents an improvement of 0.167 (seen in Table 4.8, `te` row, `th` column). We obtain this gain even though Telugu and Thai have very little in common and are not even written in the same script!

Looking at Table 4.8, we see that all values are positive with the exception of a few in the `en` row; that is, when further fine-tuning on English data and evaluating on other languages. This tells us that, starting with the mDPR model pre–fine-tuned on MS MARCO, additional fine-tuning on language $\mathcal{K}$ improves effectiveness, even if the target language is $\mathcal{L}$, where $\mathcal{L} \neq \mathcal{K}$. The exception appears to be further fine-tuning in English, where we interpret the results to be a sign of overfitting, since the pre–fine-tuning data are already in English.

The above analysis uses pre–fine-tuning on MS MARCO as the starting point, and then further fine-tuning on language-specific data. However, does pre–fine-tuning always improve effectiveness? That is, could it be the case that for target language $\mathcal{L}$, if we have data in language $\mathcal{K}$, it is better to start from the raw mBERT checkpoint and skip pre–fine-tuning? This question is answered by Table 4.9, where we show the element-wise difference between MS pFT + FT vs. FT directly; this corresponds to the element-wise difference between the two matrices in Table 4.7. Along the *diagonals*, the answer is that pre–fine-tuning always helps, and this is consistent with the results in Section 4.3.1.

However, for scenario (2b), where $\mathcal{L} \neq \mathcal{K}$, the answer appears to be, not necessarily

	ar	bn	en	fi	id	ja	ko	ru	sw	te	th
ar	0.033	0.047	0.077	-0.005	0.025	0.065	0.008	0.058	-0.035	0.107	-0.007
bn	0.101	0.006	0.123	0.055	0.085	0.118	0.075	0.110	0.006	0.112	0.076
en	0.029	0.023	0.044	-0.089	-0.083	0.072	0.030	0.013	-0.095	-0.006	0.026
fi	0.018	0.066	0.103	0.040	0.024	0.071	0.052	0.057	-0.030	-0.156	0.060
id	0.066	0.048	0.057	0.026	0.036	0.059	0.043	0.071	-0.012	0.130	0.128
ja	0.051	-0.074	0.046	-0.024	0.010	0.012	0.015	0.052	-0.057	-0.186	0.030
ko	0.094	-0.014	0.115	0.006	0.051	0.117	0.070	0.169	-0.008	0.104	0.155
ru	0.094	-0.003	0.111	0.028	0.089	0.064	0.089	0.080	-0.003	-0.020	0.088
sw	0.076	0.072	0.137	0.085	0.073	0.107	0.048	0.112	0.055	0.218	0.103
te	0.063	-0.003	0.095	-0.009	0.029	0.052	0.045	0.100	0.011	0.006	0.014
th	0.082	-0.024	0.127	0.021	0.081	0.066	0.088	0.064	0.016	-0.043	0.023

Table 4.9: Element-wise differences between MS pFT + FT vs. FT directly, answering the question: "Does pre–fine-tuning always improve effectiveness?" For in-language training, *yes*, as the diagonal entries are always positive (in **blue**). For cross-lingual zero-shot transfer, *not necessarily so*, as can be seen by the negative (in **orange**) cases off-diagonal. However, we do observe more positive than negative cases, which indicates pre–fine-tuning may help in general. We thus believe that pre–fine-tuning is a "safer" choice. Color saturation is proportional to the magnitude of values.

so. For example, training on $\mathcal{K}$ = Telugu and evaluating on $\mathcal{L}$ = Japanese, starting with the MS pFT model confers an advantage of 0.052 over starting from the default mBERT checkpoint (seen in Table 4.9, `te` row, `ja` column); with pFT, we obtain 0.449, but without pFT, we only reach 0.398 (shown in Table 4.7). On the flip side, training on $\mathcal{K}$ = Japanese and evaluating on $\mathcal{L}$ = Telugu, starting from the raw mBERT checkpoint beats starting with MS pFT by 0.186; with pFT, we obtain 0.429, but without pFT, we reach 0.615. In other words, given target language $\mathcal{L}$ and training data in another language $\mathcal{K}$, it does not appear possible to determine, *a priori*, whether we should pre–fine-tune on MS MARCO first, or directly use the raw mBERT checkpoint as the starting point.

Phrased slightly differently and summarizing our recommendations for scenario (2b), for a target language $\mathcal{L}$, there does appear to exist a language $\mathcal{K}$ (and in many cases, multiple languages) that provides positive cross-lingual transfer, although we are not able to generalize what characteristics lead to these beneficial effects. With pre–fine-tuning, further fine-tuning on language $\mathcal{K}$ appears not to hurt (in general), and thus forms the basis of our recommendation. While it is true that in some cases one can achieve even better effectiveness *without* pre–fine-tuning (see Table 4.9), we struggle to provide a consistent, explainable pattern, and thus we believe that pre–fine-tuning is a "safer" choice.

4.5 Have Model and Data: Mono vs. Multilingual BERT

Definition In the previous exploration of scenario (2a), we assume the starting point of mBERT (i.e., using mBERT as the backbone). Here, in scenario (2a′), we consider the case where a *monolingual* BERT model is available for the target language $\mathcal{L}$. What should we do in this case?

Recommendations Using a monolingual BERT backbone *can* yield a model that is more effective than using mBERT, but the monolingual model is *not* consistently better. Thus, it seems "safer" to just use mBERT as the backbone. □

Explanation Before presenting experimental results that support our recommendations, it makes sense to discuss the possible tradeoffs at work here. Previous work has found that monolingual BERT models *can* be more effective than mBERT on a variety of natural language processing tasks [112, 88, 126, 5], but the gains are *not* consistent. Obviously, the relative effectiveness depends on the quality of the monolingual model, the corpora used for pretraining, and a number of other factors. Nevertheless, it is possible that there exist "better" monolingual BERT models for a target language, compared to mBERT. However, this must be balanced against the benefits of cross-lingual transfer obtained "for free" from using mBERT by leveraging datasets in other languages, for example, pre–fine-tuning on MS MARCO. Obviously, with a monolingual (non-English) BERT, it no longer makes sense to pre–fine-tune on MS MARCO. So, the research question boils down to this: What's more important, overall monolingual "in-language" quality or cross-lingual transfer effects from pre–fine-tuning using datasets in other languages?

The answer appears to be that monolingual models *can* be better, but not consistently so, and thus our recommendation is that using mBERT appears to be "safer". These results are shown in Table 4.10, rows (a–d), where we trained DPR in five different languages (including English) starting from a monolingual backbone (described in Section 4.1.4). We are quick to point out that these models are pretrained by different groups and may vary in quality, which is dependent on the pretraining corpus, model size, hyperparameters, and a myriad of other issues.

Starting with an available monolingual BERT (in each of the target languages), we can fine-tune on just the in-language training data, shown in row (a), all in-script training data, shown in row (b), in-language and out-script training data, row (c), and all available data, row (d). The results show, not surprisingly, that with monolingual BERT, fine-tuning on anything other than in-language data appears to be pointless; in most cases it makes little

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	Avg
(T4.5–5)	mDPR (MS pFT + in-lang FT)	0.691	0.651	0.489	0.551	0.562	0.488	0.453	0.485	0.640	0.876	0.619	0.591
(T4.5–6)	mDPR (MS pFT + all FT)	0.695	0.623	0.492	0.560	0.579	0.501	0.487	0.517	0.644	0.891	0.617	0.600
(a)	monolingual DPR (in-lang FT)	0.678	–	0.426	0.573	0.545	–	0.476	–	–	–	–	–
(b)	monolingual DPR (in-script FT)	–	–	0.412	0.540	0.488	–	–	–	–	–	–	–
(c)	monolingual DPR (out-script FT)	0.682	–	0.426	0.522	0.540	–	0.454	–	–	–	–	–
(d)	monolingual DPR (all FT)	0.682	–	0.448	0.540	0.533	–	0.454	–	–	–	–	–
(e)	English DPR (in-lang FT)	0.578	0.261	0.426	0.385	0.396	0.084	0.011	0.291	0.447	0.001	0.007	0.262
(f)	English DPR (MS pFT + in-lang FT)	0.592	0.318	0.497	0.423	0.439	0.218	0.182	0.298	0.499	0.001	0.030	0.318
(g)	AfriBERTa DPR (in-lang FT)	0.442	0.186	0.236	0.321	0.355	0.220	0.140	0.094	0.465	0.548	0.263	0.297
(h)	BM25 + row (f)	0.628	0.480	0.501	0.480	0.510	0.333	0.299	0.440	0.535	0.423	0.424	**0.459**

(a) MRR@100

		ar	bn	en	fi	id	ja	ko	ru	sw	te	th	Avg
(T4.5–5)	mDPR (MS pFT + in-lang FT)	0.891	0.914	0.837	0.852	0.863	0.828	0.799	0.819	0.890	0.967	0.886	0.868
(T4.5–6)	mDPR (MS pFT + all FT)	0.900	0.955	0.841	0.856	0.860	0.813	0.785	0.843	0.876	0.966	0.883	0.871
(a)	monolingual DPR (in-lang FT)	0.894	–	0.805	0.893	0.888	–	0.820	–	–	–	–	–
(b)	monolingual DPR (in-script FT)	–	–	0.793	0.881	0.869	–	–	–	–	–	–	–
(c)	monolingual DPR (out-script FT)	0.890	–	0.801	0.865	0.877	–	0.802	–	–	–	–	–
(d)	monolingual DPR (all FT)	0.890	–	0.814	0.867	0.890	–	0.802	–	–	–	–	–
(e)	English DPR (in-lang FT)	0.807	0.581	0.805	0.712	0.740	0.297	0.021	0.689	0.790	0.017	0.035	0.499
(f)	English DPR (MS pFT + in-lang FT)	0.816	0.676	0.879	0.753	0.788	0.509	0.423	0.676	0.827	0.016	0.095	0.587
(g)	AfriBERTa DPR (in-lang FT)	0.738	0.451	0.565	0.658	0.692	0.633	0.416	0.293	0.808	0.872	0.657	0.617
(h)	BM25 + row (f)	0.874	0.946	0.867	0.830	0.908	0.723	0.646	0.795	0.872	0.813	0.851	**0.829**

(b) Recall@100

Table 4.10: Rows (a–d): "Have model and data": mono vs. multilingual BERT. Rows (e–h): "Have model, no data" (Best results are in **bold**). MRR@100 and Recall@100 of mDPR evaluated across all 11 languages and an overall average on the Mr. TYDI test set. "MS pFT": pre–fine-tuning on MS MARCO; "in-lang FT": fine-tuning on the Mr. TYDI training set from only the target language; "all FT": fine-tuning on the Mr. TYDI training data from all languages; "in/out-script": fine-tuning on training data from the target language *and* languages that are in the same/different scripts. Rows (T4.5–5, T4.5–6) are identical to rows (5, 6) in Table 4.5.

difference, and in some cases it actually hurts. This makes sense, as a monolingual BERT model, by definition, is not designed to process input in other languages.

The monolingual BERT vs. mBERT comparison is shown by row (a) vs. row (T4.5–6), since with mBERT we can exploit pre–fine-tuning on MS MARCO and training data in other languages. We see that effectiveness is comparable, and Finnish is the only language where using a monolingual model yields higher MRR@100. In row (f), more fully discussed in Section 4.6, we start with English BERT, apply pre–fine-tuning on MS MARCO and additional fine-tuning on English data. For English retrieval, this yields the highest ef-

fectiveness, since we are leveraging both a high-quality monolingual model *and* exploiting (in-language) pre–fine-tuning. The difference between row (f) and row (T4.5–5) can be attributed to mBERT vs. English BERT, and we can see that even in the best possible case, monolingual BERT is only marginally better than mBERT.

These findings support our recommendation to start with mBERT, which we view as the "safer" option, and has the additional benefit of simplifying the operational aspects of model training (e.g., dealing with different tokenizers, different model sizes, etc.). Our results also show that for English retrieval, English BERT beats mBERT slightly, but this is not inconsistent with our recommendation since our focus is on multilinguality.

4.6 Have Data, No Model

Definition In Scenario (3), the target language $\mathcal{L} \notin$ mBERT (i.e., we do not have a multilingual model that has been pretrained with the target language), but we *do* have data in the target language $\mathcal{L}$.

Recommendations Training a dense retrieval model with an English backbone, even for a non-English target language, can be effective. Similarly, training a dense retrieval model using a multilingual BERT, even one that does not include the target language, can also be effective. Even if these dense models are not effective on their own, they appear to provide valuable relevance signals that can improve over bag-of-words BM25 results. □

Explanation The core of our recommendations here can be characterized as "something is better than nothing", and the results are quite surprising. Before proceeding, however, we should first defend this as a reasonable scenario: We *do* think that it is plausible to have training data in some low-resource language that is not covered by mBERT. In short, search existed before neural networks and pretrained transformers. It is certainly possible for some organization to build (or even have already built) a bag-of-words search engine in some low-resource language outside the set used to pretrain mBERT, and then to use this search engine to gather relevance judgments (perhaps to train a simple learning-to-rank model). After all, mBERT only covers around 100 languages, and there are thousands of languages in the world we would potentially like to search in. The amount of relevance judgments used in our experiments (as low as around 1000 examples) seems modest (see Table 3.2), and thus we argue that it is entirely plausible to have some training data in a target language not covered by mBERT.

What if we just start with *English* BERT and fine-tune on training data in the target language $\mathcal{L}$, even though at face value this seems non-sensical, since the language might bear no similarity to English? These results are shown in Table 4.10: row (e), without MS MARCO pFT, and row (f), with MS MARCO pFT. We see that, for example, training *English* BERT with *Indonesian* relevance judgments yields a model that is quite "reasonable" for retrieval in *Indonesian*. We achieve an MRR@100 0.439, row (f), compared to mDPR (MS pFT + all FT), which achieves 0.579. Note that Indonesian and English are both written in the Latin script. However, even *across scripts*, we can build dense retrieval models that perform much better than random, e.g., Arabic and Bengali, but in other cases, e.g., Telugu and Thai, the results are pretty close to garbage. Interestingly, pre–fine-tuning English DPR on MS MARCO improves effectiveness across all languages except Telugu. This seems to indicate that, with a high-quality English DPR model, we can obtain at least a "reasonable" dense retrieval model for many languages.

An alternative solution for this scenario is to use a multilingual BERT that does not contain the target language $\mathcal{L}$. Here, we obviously cannot use mBERT, since all the languages in Mr. TYDI are in mBERT's pretraining corpus. Instead, we use AfriBERTa [96], an mBERT-style multilingual language model pretrained from scratch on 11 African languages. Hence, all the languages in Mr. TYDI, except Swahili, are absent from its pretraining corpus. Results are shown in Table 4.10, row (g); note that pre–fine-tuning AfriBERTa on English MS MARCO does not make sense because English is not one of the languages in the pretraining corpus. Surprisingly, AfriBERTa obtains very "reasonable" results on most languages, despite never having been pretrained on them. However, it seems like English BERT performs on par or better on all languages except Telugu and Thai. From these results, we can see that a multilingual BERT model can be used to bootstrap "reasonable" dense retrieval models in this scenario.

Our recommendation for scenario (3) is that if a pretrained model does not exist for the target language, then simply using English BERT or another multilingual BERT can be helpful. However, the astute reader might point out that in this scenario, dense retrieval models perform worse than BM25, so what's the point of even using them? We address this potential objection by combining the results in row (f) with bag-of-words BM25 results, shown in Table 4.10 as row (h). Here, we see that the hybrid dense retrieval + BM25 scores are generally better than BM25 results alone. Even in the case of Telugu and Thai, the two cases where English BERT yields terrible low-quality output, the hybrid results are just a tiny bit worse (`te`) or marginally better (`th`). Thus, it appears that even if the models trained following our recommendations are not effective, they can still provide complementary relevance signals to improve bag-of-words BM25 in a hybrid fusion approach. Thus, "something is better than nothing".

Chapter 5

Understanding

In the previous chapter, we observed scenarios in which models trained on one language "magically" performed well on other languages, a phenomenon commonly referred to as *cross-lingual transfer*. This chapter uses controlled modifications of multilingual data, token embeddings, and vocabularies to understand how much superficial token-form overlap contributes to cross-lingual transfer, and whether multilingual models can rely on semantic concepts at the token level.

5.1 Shared Tokens

It is usually assumed that corpora from different languages have very different vocabularies, either in terms of naturally occurring words or algorithmically generated subwords from the tokenizers of pretrained multilingual transformers. We question this assumption and hypothesize that shared vocabularies (especially after tokenization) contribute to the cross-lingual transfer capabilities of multilingual models. In this section, we refer to tokens generated by transformer tokenizers as *subwords*, and tokens that naturally exist in a corpus simply as *words*. Our analysis of overlapping vocabularies across different languages shows that the proportions of common words and subwords are much higher than one might have expected. We speculate that this is naturally caused by code-switching in Wikipedia articles and further enriched by the (relatively) compact vocabulary space in current multilingual pretrained transformers.

	ar	bn	en	fi	id	ja	ko	ru	sw	te	th	**Total**
ar	94.1%	0.0%	2.5%	0.7%	1.0%	0.1%	0.5%	0.2%	0.9%	0.0%	0.0%	1,397,830
bn	2.2%	66.0%	14.2%	3.3%	5.7%	0.3%	3.8%	0.4%	3.9%	0.1%	0.1%	64,379
en	0.7%	0.1%	37.4%	13.7%	21.7%	0.8%	6.0%	1.4%	17.9%	0.0%	0.3%	1,802,836
fi	0.0%	0.0%	3.4%	90.7%	3.6%	0.1%	0.1%	0.2%	2.0%	0.0%	0.0%	2,337,303
id	1.0%	0.1%	16.6%	10.4%	51.3%	1.0%	3.9%	0.6%	15.1%	0.0%	0.2%	382,769
ja	–	–	–	–	–	–	–	–	–	–	–	–
ko	0.0%	0.0%	0.9%	0.2%	0.4%	0.3%	97.5%	0.1%	0.6%	0.0%	0.0%	4,192,486
ru	0.1%	0.0%	2.6%	1.0%	1.2%	0.1%	0.5%	93.4%	1.1%	0.0%	0.0%	2,994,515
sw	0.7%	0.0%	9.4%	6.6%	11.8%	0.0%	0.3%	0.3%	70.7%	0.0%	0.0%	119,526
te	0.7%	0.2%	16.2%	3.5%	7.7%	0.1%	0.3%	0.3%	6.8%	64.3%	0.0%	65,950
th	0.0%	0.0%	0.9%	0.2%	0.4%	0.2%	0.4%	0.0%	0.4%	0.0%	97.5%	2,470,418

Table 5.1: Analysis of unique whitespace-delimited tokens in Mr. TYDI. Each row represents a different corpus; we omit Japanese here because it is written without spaces. Each column indicates the percentage of words belonging to that language, based on an off-the-shelf language id package. The total number of unique words in each language is reported in the rightmost column. The values are highlighted *row-wise*: the color gradient ranges from **blue** (high values) to **orange** (low values)

5.1.1 Shared Word Vocabularies

In Wikipedia, named entities are additionally written in their "native" language when it is different from the language of the corpus. This is most common in English articles, but also appears frequently in pairs of non-English languages. For example, the Korean article about Saladin, the 12th century sultan of the Ayyubid dynasty, contains his Arabic name in the first sentence of the article. This means that naturally occurring corpora in one language contain tokens from other languages, and we observe this occurring across all the Mr. TYDI languages. Trivially, all languages share in the use of Arabic numerals as well.

We attempt to quantitatively characterize this phenomenon in the Mr. TYDI corpora. In Table 5.1, we report the percentage of shared vocabularies of naturally occurring words, which illustrates the prominence of code-switching, especially surrounding English. For each language, we begin by tokenizing its corpus using whitespace and punctuation; we omit Japanese from this analysis since it is written without spaces. We apply an off-the-shelf Python language id package called `langdetect`[1] to determine the language of each unique token. These results are shown in Table 5.1, where each row represents the corpus in a language and each column indicates the identified language. Considering the

[1] https://github.com/Mimino666/langdetect

	ar	bn	en	fi	id	ja	ko	ru	sw	te	th
ar	100.0%	92.3%	62.2%	82.3%	81.3%	78.2%	81.9%	68.6%	94.2%	92.2%	90.5%
bn	57.3%	100.0%	39.1%	54.8%	54.7%	52.1%	57.2%	43.8%	72.0%	80.3%	67.8%
en	98.1%	99.2%	100.0%	99.0%	98.8%	98.5%	97.7%	94.5%	99.7%	98.5%	99.6%
fi	82.0%	88.0%	62.6%	100.0%	81.3%	78.8%	81.5%	70.4%	94.9%	89.2%	88.6%
id	87.7%	95.1%	67.6%	88.0%	100.0%	84.9%	88.1%	73.6%	97.0%	94.1%	95.2%
ja	88.3%	94.8%	70.6%	89.3%	88.8%	100.0%	92.5%	76.5%	96.4%	94.1%	96.5%
ko	79.4%	89.4%	60.0%	79.2%	79.2%	79.4%	100.0%	65.9%	91.5%	88.3%	91.5%
ru	92.2%	94.9%	80.6%	95.0%	91.7%	91.1%	91.4%	100.0%	97.8%	94.2%	96.2%
sw	57.7%	71.1%	38.8%	58.4%	55.1%	52.3%	57.9%	44.6%	100.0%	75.2%	67.2%
te	51.0%	71.5%	34.6%	49.5%	48.2%	46.1%	50.4%	38.8%	67.9%	100.0%	61.0%
th	71.6%	86.5%	50.0%	70.4%	69.9%	67.6%	74.7%	56.6%	86.8%	87.3%	100.0%

(a) mBERT tokenizer

	ar	bn	en	fi	id	ja	ko	ru	sw	te	th
ar	100.0%	84.7%	43.7%	71.1%	70.1%	62.5%	67.2%	54.2%	87.7%	82.1%	78.8%
bn	47.0%	100.0%	25.0%	44.6%	44.0%	39.3%	44.6%	31.7%	64.9%	69.6%	56.7%
en	93.8%	96.7%	100.0%	96.4%	95.6%	90.4%	89.9%	87.3%	97.9%	94.1%	95.0%
fi	71.0%	80.3%	44.8%	100.0%	71.9%	63.4%	67.2%	57.4%	90.3%	80.4%	77.3%
id	77.7%	87.8%	49.3%	79.8%	100.0%	69.9%	74.4%	59.8%	92.9%	85.0%	84.9%
ja	76.3%	86.3%	51.4%	77.5%	77.0%	100.0%	83.4%	61.0%	89.6%	84.1%	87.2%
ko	66.8%	80.0%	41.7%	67.1%	66.8%	68.0%	100.0%	51.0%	83.4%	77.8%	80.2%
ru	82.8%	87.4%	62.2%	87.9%	82.4%	76.5%	78.3%	100.0%	93.0%	85.5%	86.2%
sw	47.6%	63.6%	24.8%	49.2%	45.6%	39.9%	45.6%	33.1%	100.0%	66.1%	56.4%
te	42.3%	64.6%	22.6%	41.5%	39.5%	35.5%	40.3%	28.8%	62.7%	100.0%	52.2%
th	59.4%	77.0%	33.4%	58.4%	57.7%	53.8%	60.7%	42.5%	78.2%	76.2%	100.0%

(b) XLM-R_{Large} tokenizer

Table 5.2: Analysis of unique tokens produced by pretrained transformer tokenizers in Mr. TYDI. Each language cell in a column (L_c) indicates the percentage of overlapping tokens between itself and the language cell in a row (L_r) over the total number of unique tokens in L_c. For example, the `bn`–`ar` entry in Table 5.2a is 57.3%, indicating that the number of overlapping subwords produced by the mBERT tokenizer for the `ar` and `bn` corpora accounts for 57.3% of the total unique subwords in the `bn` corpus. The values are highlighted *column-wise*: the color gradient ranges from **blue** (high values) to **orange** (low values)

first row, showing results for the Arabic corpus: We see that 94.1% of unique words are indeed identified as Arabic. However, there are non-trivial numbers of words that are identified as belonging to other languages as well. For example, the Arabic article on Isaac

Newton contains his name in English. In total, we observe that about 2.5% of the unique words in the Arabic corpus are actually identified as English. Although the results in Table 5.1 are a bit noisy because language id is imperfect and whitespace tokenization can be problematic for some cases,[2] we have manually verified these results as "reasonable" via spot checking. Regardless, this analysis conclusively shows that code-switching naturally occurs in Wikipedia.

Looking at the English row, we see that English Wikipedia exhibits the greatest diversity in code-switching "out of English" into other languages, e.g., Saladin to Arabic, Peter the Great to Russian, Tokugawa Ieyasu to Japanese. Looking at the English column, we see that English is the most common language that non-English corpora code-switch "into". For example, the Wikipedia pages on Albert Einstein in non-English languages all mention his name in English. While English-centric cross-lingual anchors dominate, we see many examples that do not involve English: for example, the Arabic article about Tokugawa Ieyasu contains his name in Japanese; the Russian article about the Chittoor district in India contains the native Telugu name. Furthermore, these pivots are, for the most part, named entities, which are often the topic of natural language questions!

We believe this provides at least an initial explanation of why pre–fine-tuning on MS MARCO data works so well, even in a zero-shot setting, e.g., Table 4.2, row (4). It seems reasonable to assume that mBERT has already "internalized" dictionaries of named entities across many languages as a natural byproduct of MLM pretraining,[3] given that the code-switching behavior described above in essence provides parallel data. Wikipedia is part of the pretraining corpus for most pretrained models, and it is likely that code-switching is common as well in other genres of text in the pretraining corpus. Thus, when we fine-tune in English, these cross-language anchors enable the model to transfer relevance matching capabilities across languages. Since we have also demonstrated the existence of non-English cross-lingual anchors, this might begin to explain why fine-tuning on language $\mathcal{K}$ helps language $\mathcal{L}$ (where both $\mathcal{K}$ and $\mathcal{L}$ are not English). We further note that in QA applications, getting the named entity right is "half the battle" (if not more). Consider a question about the birthdate of a person; if the model simply retrieves passages about the correct person, the results will already be "reasonable" (and may in fact contain the answer, just by sheer luck).

[2]For example, it can be difficult to identify the language of words in Latin-script languages without context.

[3]We confirmed via spot-checking that this is indeed the case by forcing mBERT to perform cross-lingual prediction of masked tokens, and indeed the model is able to generate correct named entity translations.

5.1.2 Shared Subword Vocabularies

We empirically observe that after tokenization by pretrained multilingual transformers, corpora from different languages share a large overlap in subwords, reported in Table 5.2. For example, consider Table 5.2a, where the corpus in each language is tokenized using the mBERT tokenizer. To illustrate the information captured in the table, we provide two examples:

- Overlapping subwords between Arabic and English represent 98.1% (row `en`, column `ar`) of subwords in the Arabic corpus and 62.2% (row `ar`, column `en`) in the English corpus.
- Overlapping subwords between `te` and `th` represent 87.3% (row `th`, column `te`) in Telugu and 61.0% (row `te`, column `th`) in Thai.

The numbers of shared subwords were a total surprise to us. In Table 5.2b, where the tokenization is performed using XLM-R_{Large}, the overlap percentages are generally smaller but remain quite high. We can rule out data quality issues because these results are consistent with the whitespace tokenization analysis above; furthermore, these figures pass both sanity checks and spot checks.

The reality here is twofold: First, many shared subwords across languages are *not* from either language. That is, the overlap of subwords found in both the English and Arabic corpora (for example) comprises a large number of subwords *from other languages*! This makes sense given the results in Table 5.1, e.g., the English corpus code-switches into every other language (recall that our analysis is performed in terms of *unique* words.) Second, we observe that even a small number of words can produce a large number of subwords in the vocabulary space of that model. This is because the vocabulary space for *all* languages is rather constrained: the numbers of subwords in mBERT and XLM-R_{Large} are respectively 120K and 250K, but shared by around 100 languages. For example, consider Table 5.1 row `bn`, column `ar`: the subwords produced by the mBERT tokenizer from the 2.2% of Arabic words in the Bengali corpus covers 57.3% of the Arabic subwords processed from the Arabic corpus, as seen in Table 5.2a.

The upshot is that multilingual transformers are almost never presented with clean, "pure", monolingual corpora. Hence, we expect the models to learn connections across multiple languages (using the cross-lingual anchors) and model the subwords jointly, thus enabling downstream cross-lingual transfer capabilities. Furthermore, subword overlap may explain why English BERT works well even when fine-tuned on other languages in

Section 4.6. The English corpus already contains a non-trivial amount of subwords from other languages! We hypothesize that these shared subwords (which we show exist even in monolingual English BERT) provide anchors for downstream cross-lingual transfer. Indeed, this explanation is consistent with the literature [57, 26, 138, 33, 103, 63], where at least some researchers agree that higher subword overlap correlates with improved cross-lingual ability.

5.2 Impact of Incidental Multilingual Tokens in Embedding Finetuning

The investigations above show that there is a large number of token overlaps across languages, some caused by naturally shared words among languages in the same script, while the others are from unintentional multilingual text such as code-switching or translation examples [15]. Further investigation on how such incidental multilingual text affects cross-lingual transfer performance at the embedding finetuning stage is conducted in later work [79]. We find that the effectiveness of cross-lingual transfer is mostly preserved without any incidental multilingual text between the training data on source language and evaluation data on the target language: removing IMT tokens shows *insignificant differences* in the transfer results in 38 of 41 pairs of languages, where the decrease in effectiveness in most pairs of languages is less than 2% and overall less than 10%. This suggests that models are able to perform cross-lingual transfer via semantic understanding of the input sentences or passages.

5.2.1 Experimental Design and Setup

IMT Token Removal. The same token removal procedure applies for both training data in the source language (L_s) and evaluation data in the target languages (L_t). Given training data in L_s, we remove words detected to not be in the same script as L_s. Specifically, we use an off-the-shelf language detection library LANGDETECT.[4] We pre-process the training data for the language L_s by first tokenizing the data using the mBERT tokenizer, then removing all tokens that `langdetect` finds to be in a script other than the script of L_s. For example, if L_s is Russian, our process will remove all *non-Cyrillic* tokens from training passages. The remaining tokens, which are in the same script as L_s, form new training data used for fine-tuning mDPR. Similarly, given evaluation corpus and queries in L_t, we remove

[4] https://pypi.org/project/langdetect

	w/ IMT?	fa	hi	bn	ru	fr	es	fi	id	sw	te	ar	th	zh	ja
fa	✓	0.474	0.318	0.349	0.390	0.283	0.380	0.383	0.337	0.303	0.371	0.433	0.410	0.312	0.393
	×	–	0.318	0.322†	0.372	0.287	0.376	0.383	0.334	0.307	0.369	–	0.373†	0.310	0.375
	%	–	100.1%	92.5%	95.3%	101.7%	98.8%	99.9%	99.1%	101.3%	99.4%	–	91.0%	99.3%	95.5%
hi	✓	0.343	0.385	0.343	0.335	0.218	0.297	0.308	0.299	0.273	0.327	0.363	0.310	0.244	0.339
	×	0.331	–	0.340	0.315	0.221	0.293	0.309	0.307	0.253	0.308	0.342	0.296	0.252	0.313
	%	96.5%	–	99.3%	94.1%	101.6%	98.8%	100.6%	102.7%	92.8%	94.4%	94.3%	95.4%	103.3%	92.1%
bn	✓	0.441	0.359	0.597	0.502	0.263	0.369	0.453	0.432	0.362	0.497	0.498	0.523	0.337	0.482
	×	0.396	0.380	–	0.458†	0.274	0.365	0.466	0.430	0.364	0.521	0.495	0.493	0.328	0.488
	%	89.8%	105.8%	–	91.2%	104.0%	99.0%	102.8%	99.6%	100.5%	104.8%	99.3%	94.2%	97.4%	101.3%

Table 5.3: nDCG@10 on the development set of Persian (fa), Hindi (hi), Bengali (bn) from MIRACL. **Column**: source languages L_s; **Rows**: target languages L_t. Under each target language, the first two rows are trained on official MIRACL training data (*w/ IMT*) and data with non-self script tokens removed (*w/o IMT*). The third row (%) is the relative effectiveness of *w/o IMT* compared to *w/ IMT*, highlighted in , where higher saturation indicates higher performance. Significantly different results are marked with † ($p < 0.01$ with paired t-tests).

tokens that are detected to be in a different script from L_t. Punctuation and numbers are always kept in this process.

Training and Inference. All our experiments are based on Tevatron [39], a flexible framework for the training and inference of common retrieval methods. During training, we start from the mBERT checkpoint provided by HuggingFace [136], then finetune it on two versions of the MIRACL training data in each language L: the official MIRACL training data[5] (baselines) and the one with non-L-script tokens removed as described in Section 5.2.1. Similarly, we adopt two versions of evaluation collections given language L: the official MIRACL collection[6] and the collection with non-L-script tokens removed. We evaluate the models using the official metric nDCG@10 of MIRACL. We use the same hyper-parameters as previous works [150]: All experiments are fine-tuned for 40 epochs with a batch size of 64 and a learning rate of 1e-5. We limit the query lengths to 32 tokens and the passage lengths to 256 tokens on both training and inference steps.

Language Selection. Models are evaluated on three languages: fa, hi, and bn, and trained extensively in 14 languages from MIRACL. These three evaluation languages form a language group that contains both similar and diverse linguistic features: On the one hand, they belong to the same language family, have a substantial amount of loan words from

[5] https://huggingface.co/miracl/miracl
[6] https://huggingface.co/miracl/miracl-corpus

each other, and have the same word order (Subject–object–verb; SOV). On the other hand, they are all written in different scripts, which makes direct token sharing impossible. They also have different gender systems and morphological typologies (Persian is agglutinative, while the other two are fusional).

5.2.2 Results

Table 5.3 compares zero-shot results before and after removing incidental multilingual text as mentioned in Section 5.2.1. We fine-tune models on fourteen source languages individually and evaluate on three target languages. Under each source–target language pair, we report three scores: (1) *w/ IMT*: using the official training and evaluation data, which contains IMT text; (2) *w/o IMT*: tokens in a non-self script are removed from training and evaluation data as mentioned in Section 5.2.1; (3) *Percentage*, the relative percentage of *w/o IMT* scores compared to *w/ IMT*. Entries under *Percentage* are highlighted in , where higher saturation indicates higher effectiveness and the white background indicating relative effectiveness at 90%.

Cross-lingual Transfer Capacity is Mostly Preserved without Incidental Multilingual Text. As most visibly shown by the percentage row, all source–target language pairs achieve over 90% relative effectiveness after shared tokens are removed, where most of them have minimal effectiveness loss ($< 2\%$) or even outperform the "w/ IMT" baseline. Among the 41 pairs, only 3 pairs show a significant drop in transfer effectiveness after token removal. This indicates that the cross-lingual transfer capacity of dense retrievers can be effectively achieved via the semantic understanding of the input sentences alone, with no overlapping tokens between the training and evaluation data.

5.3 Token-Level Semantic Concepts

The results above show that cross-lingual transfer can occur even when the source and target languages share no token overlap, suggesting that models may capture semantic information beyond surface forms. Building on this observation, the following work investigates whether such conceptual representations may already exist at the level of input tokens [146].

[6] "w/o IMT" score is not computed for pair `fa`–`ar` since they are in the same script.

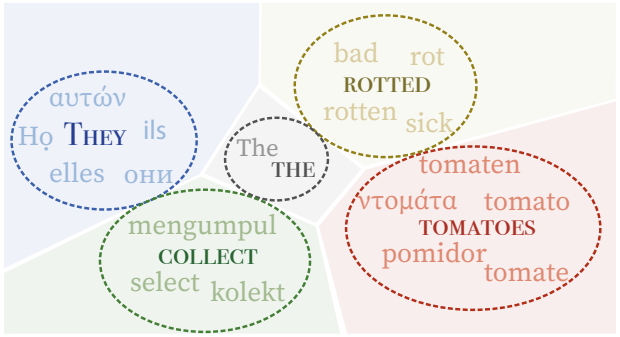


Figure 5.1: Words with similar meanings or inflectional changes fall under semantic concepts, as indicated by the colors. The sentence "They collected the rotted tomatoes." is from XNLI [25].

Human understanding of text depends on general semantic concepts of words that are robust to their superficial forms (Figure 5.1). The most obvious examples are semantically equivalent words in different languages: code-switching "they" to "они", or "tomatoes" to "tomate".[7] The sentence's meaning is preserved for people who understand both languages. The robustness in understanding may also apply to some of the inflectional changes: swapping "tomatoes" for "tomato", while losing the plural form, still conveys the overall sentence meaning. Finally, even when words are replaced with ones that are only vaguely related in semantics, e.g., from "rotted" to "bad", resulting in a phrase that is less precise, the sentence meaning could still be interpreted as being in line with the original sentence. While the definition and representation of concept have not been universally agreed upon [52, 37, 43, 40, 36, 113], and their alignment varies across languages depending on domain and cultural context [125], it is generally accepted that concepts underlying words could be shared or even universal across languages [16, 43] and have a pivotal role in many aspects of language understanding [90, 36].

On the other hand, language models perceive input text and perform prediction based on individual tokens and their embeddings. We thus ask: *To what degree do current multilingual language models (mLMs) understand based on token-level semantic concepts?* Based on the existing mLMs, we form "semantic tokens" by grouping the subwords based on the similarity of their word embeddings, which henceforth share the same "semantic embedding".[8] The updated mLMs are evaluated on five downstream tasks, which cover thirty

[7] "они": "they" in Russian; "tomate": "tomato" in Spanish

[8] We understand that the formed semantic tokens are not perfect and may contain subwords of loosely related or unrelated meanings. On one hand, we provide inspection of the formed semantic tokens in Figure 5.8 and 5.11, which suggest that the grouped tokens could reflect coherent semantics. On the other

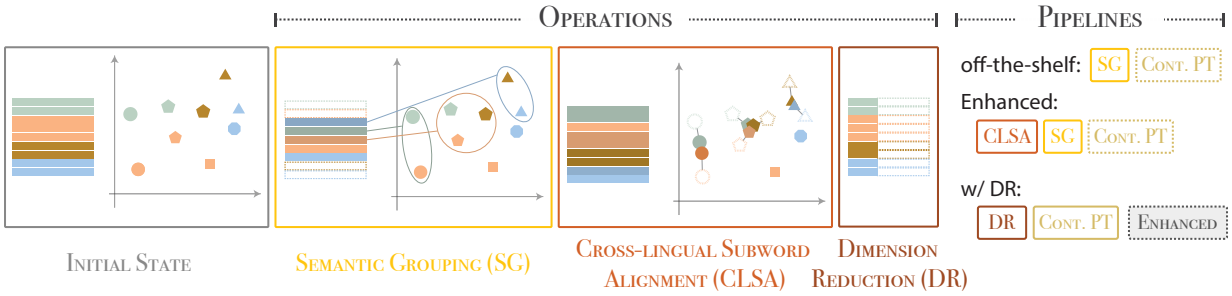


Figure 5.2: Illustrations of the operations that modify the word embeddings and the pipelines composed of these operations. "Cont. Pt": Continual Pretraining, whose illustration is skipped due to its prevalence. The colored rows represent the embeddings of the subwords and the coordinates depict their spatial distances. The shapes indicate the underlying semantics and the colors indicate the languages of the subwords. Under the "Pipelines", the solid boxes denote the required operations that are core to the pipeline or essential to the model functionality; the dashed boxes denote the optional operations that are only to provide additional improvement. Better viewed in color.

languages in total and include classification and embedding tasks in different granularities.

We find that with a small number of semantic tokens and their embeddings, the mLMs can preserve most of the downstream effectiveness: semantic tokens in 5% of the original vocabulary size achieve 90% of the effectiveness on classification tasks, and 20% of the semantic tokens achieve over 85% effectiveness on the embedding tasks. These findings suggest that while nuances exist in the meaning of each subword, the general semantics representations could get the models a long way in prediction-making.

Next, we eliminate the confounding factor of embedding size: While forming semantic tokens, the number of the word embedding parameters is also reduced. Does the change in effectiveness stem from changes in subwords or reduced embedding parameters? We thus apply the semantic grouping to the word embeddings with reduced parameter size via truncating the embedding dimension, finding that the same observation persists even when the parameters could not be further reduced via dimension reduction. This suggests that the above results are not confounded by the embedding size but rather are a result of the semantic grouping.

Additional experiments suggest that the findings generalize to mLMs with different tokenizers and model sizes. Inspection shows that the grouped subwords indeed exhibit

hand, we consider our results as a lower bound, where techniques that form more accurate semantic tokens may provide a clearer analytical signal.

a wide range of semantic similarities: numbers, punctuation, synonyms, and translations across multiple languages under different scripts. Lastly, we find that the zero-shot results on certain classification tasks with semantic tokens are on par with or even better than the original models, suggesting that the shared subword-level semantics may serve as the transfer anchors for cross-lingual generalization. These interventions also reveal potential practical opportunities for vocabulary compression and post-hoc representation enhancement.

Our contributions are as follows: (1) We find that mLMs can preserve a majority of the downstream effectiveness with a small number of shared subword-level semantics (Section 5.4.1, 5.4.2). (2) We show that the findings are general across mLMs with different tokenizers, model sizes, and other aspects (Section 5.4.3). (3) Inspection reveals that the grouped subwords exhibit a wide range of semantic similarities (Section 5.4.4). (4) The zero-shot results suggest that the shared subword-level semantics may serve as transfer anchors for cross-lingual generalization (Section 5.4.5).

5.3.1 Operations and Pipelines

The experimental setting in this paper can be categorized into several pipelines composed of multiple independent operations, centering on semantic grouping. All operations, except for the continual pretraining, only affect the vocabulary V and word embeddings $E \in R^{(|V|,D)}$, where $|V|$ is the vocabulary size and D is the initial word embeddings dimension. Figure 5.2 illustrates all operations except the continual pretraining due to its prevalence, as well as the pipelines composed of the operations.

Semantic Grouping (SG) Given the vocabulary V and its word embeddings E, multiple semantically similar subwords are grouped into a single "semantic token" and henceforth share the same "semantic embedding". The semantic embedding is then initialized by the averaged embeddings of the grouped words. That is, after the grouping, the updated LM has a new vocabulary V' composed of semantic tokens and new word embeddings $E'_{vocab} \in R^{(|V'|,D)}$, where $|V'| < |V|$. We define the grouping ratio as $r_G = |V'|/|V|$.

K-Means. Subwords are grouped via K-Means based on the cosine distance of their word embeddings. We choose K-Means due to its flexibility in the produced number of groups $|V'|$, and use the cosine distance as experiments show that it has better performance, especially at high word embeddings dimensions (Section 5.4.6). The inspections of the groups show that they could reflect coherent semantics (Figure 5.8 and 5.11). In this work, we set $|V'|$ to correspond to a grouping ratio $r_G \in \{5\%, 10\%, 20\%, 40\%\}$.

First-k. As the baseline, we keep only the first-k emerged subwords while training the tokenizer, where $k = |V'|$. In this way, the size of the vocabulary and word embedding vectors remains the same as in the corresponding semantically grouped models, yet each embedding corresponds to only a single subword as the original LM.

Cross-lingual Subword Alignment (CLSA) Other than clustering on the off-the-shelf word embedding, we investigate manually aligning the embeddings of subwords across different languages. Specifically, we gather cross-lingual word pairs from bilingual dictionaries (i.e., MUSE [24]; PanLex[9]) and concept lists of multilingual words (i.e., Concepticon [76]; ColexNet [80]). We only preserve the words that are tokenized into a single subword. Then, the word embeddings are trained using InfoNCE loss [97] with in-batch negatives. Note that only the parameters of word embeddings are used and updated in this operation, while the rest of the model remains untouched.

Dimension Reduction (DR) Section 5.4.2 involves experiments that reduce the dimension of word embeddings, where we simply *remove* the final $D - d$ dimensions of the word embeddings to form the new word embeddings $E_d^{'} \in R^{(|V|,d)}$ $(d < D)$, and pad each embedding vector with zeros on the fly. Note that the positional and token-type embeddings are not affected by this operation. While there are alternative options to reduce the word embedding parameters, we adopt DR for its simplicity and to minimize changes in model architecture. Due to its great modifications to the word embeddings, this operation is always followed by continual pretraining (Section 5.3.1).

Continual Pretraining The above operations may lead to a potential mismatch between the word embeddings and the rest of the model parameters. To address the mismatch, we continually pretrain the entire LM using Masked Language Modeling (MLM) objectives to align the updated embeddings and language model parameters [31]. All continual pretraining uses Wikipedia data of 28 languages,[10] where the languages are selected based on the coverage of downstream tasks.

5.3.2 Evaluation Tasks and Datasets

We evaluate all configurations on five multilingual downstream tasks, where four are classification and one is embedding. The classification tasks include word-level understanding,

[9] https://panlex.org/

[10] ar, bg, bn, de, el, en, es, fa, fi, fr, ha, hi, id, ig, ja, ko, lg, ru, rw, sw, te, th, tr, ur, vi, wo, yo, zh

Dataset Name	Task Name	Task Type	Granularity	# L.	Languages
MasakhaNER	NER	classification	word-level	10	am, ha, ig, rw, lg, sw, wo, yo, luo, pcm
XNLI	NLI	classification	sentence-level	15	ar, bg, de, el, en, es, fr, hi, ru, sw, th, tr, ur, vi, zh
TyDi QA	QA	classification	passage-level	11	ar, bn, en, fi, id, ja, ko, ru, sw, te, th
MIRACL	P Reranking P Retrieval	classification embedding	passage-level	18	ar, bn, de, en, es, fa, fi, fr, hi, id, ja, ko, ru, sw, te, th, yo, zh

Table 5.4: Downstream Tasks and Datasets. Granularity suggests the information level required from input data to perform the task. "P Retrieval": Passage Retrieval; "P Reranking": Passage Reranking.

sentence-level understanding, and passage reranking, whereas the embedding task includes passage retrieval. All evaluation datasets cover at least 10 diverse languages. See Table 5.4 for details. We hope these provide a comprehensive evaluation regarding the task nature, model structures, and languages.

MasakhaNER [1] is a named entity recognition (NER) benchmark including 10 underrepresented African languages. We use its version 1.0 in this work.

TyDiQA-GoldP [20] is the gold passage task of TyDiQA [20], a question answering (QA) dataset that includes 11 topologically different languages. It requires predicting correct answer based on gold answer passages. We refer to it as TyDiQA for simplicity.

XNLI [25] is a natural language inference (NLI) dataset that extends the development and test sets of MultiNLI [135] to 15 diverse languages.

MIRACL [151] is a monolingual information retrieval dataset that provides training data for 16 diverse languages and evaluation data for an additional 2 languages. Two tasks are performed on MIRACL: passage retrieval and passage reranking, which fall under embedding and classification tasks, with their model formulations described in Sections 2.1.2 and 2.1.3 respectively. In the rest of this paper, MIRACL refers to the passage retrieval task and MIRACL (rerank) refers to the passage reranking task. We use the classic DPR [58] and monoBERT [93, 94] models for each task.

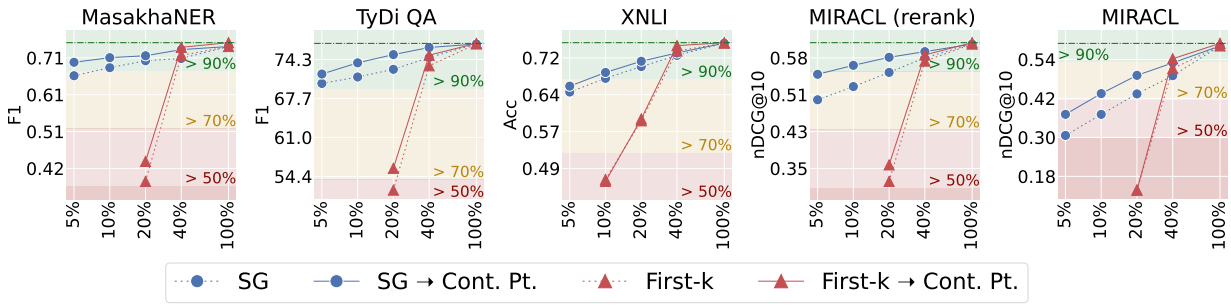


Figure 5.3: Results of mBERT with vocabulary and embeddings after semantic grouping (SG) or simply reduced size (First-k). **x-axis**: the grouping ratio r_G in log scale. The background colors indicate the relative performance to the oracle results, i.e., continual pretrained mBERT with full vocabulary, indicated by the green dashed lines on top. green: >90%, yellow: 70%–90%, red: 50%–70%. The scores of First-k at $r_G = \{5\%, 10\%\}$ are skipped in the figures as they greatly skew the y-axis.

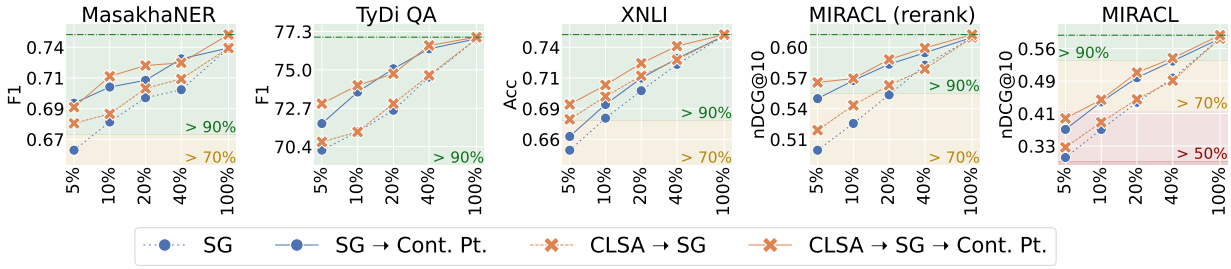


Figure 5.4: Results of mBERT after semantic grouping (SG) with and without applying cross-lingual subword alignment (CLSA).

5.4 Token-Level Semantic Concepts Results

5.4.1 Semantic Grouping

Figure 5.3 illustrates the results of applying SG on mBERT [31] across a spectrum of grouping ratios r_G (x-axis). The background colors indicate the range of relative effectiveness compared to the oracle results, which are the scores of the original mBERT with continual pretraining applied. Each point in the figures represents the average score across all languages for the given configurations.[11]

[11] Due to space constraint, Section 5.4 only provides visualization of all results, where numerical scores can be found in Table 5.5. We also investigate the impact on individual languages. While the overall trend

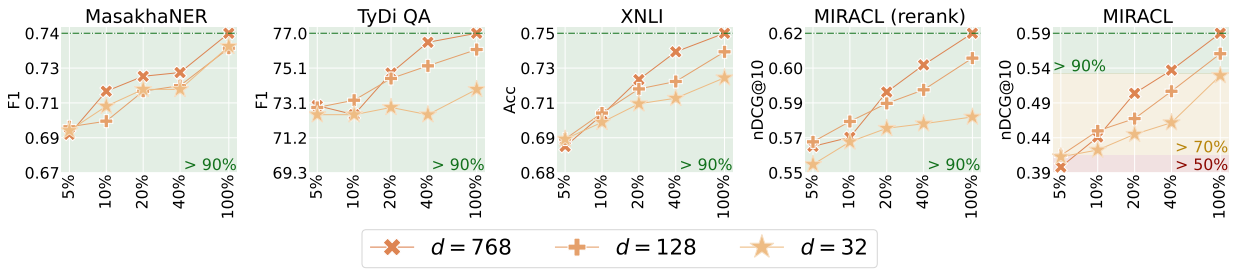


Figure 5.5: Results of semantic grouping with word embedding dimension $d \in \{768, 128, 32\}$, with CLSA and continual pretraining. Background colors design is identical to Figure 5.3. The relative performance of all classification tasks are higher than 90%, thus with full green background.

Semantically similar subwords could share the same embeddings to a large degree. In classification tasks (the left four sub-figures), applying SG alone (the blue dashed lines) can already preserve over 90% effectiveness on the downstream tasks with 10% of the original vocabulary size. After applying continual pretraining (the blue solid lines), the same level of effectiveness ($> 90\%$) can be preserved with only 5% of the original vocabulary size. The embedding task (the rightmost sub-figure) is comparatively more sensitive to the semantic grouping, yet still maintains over 85% effectiveness with 20% of the original vocabulary size after the continual pretraining. The different behaviors suggest that classification tasks may require only coarse semantic representations to make predictions, while the embedding tasks require more fine-grained lexical representations to produce reasonable sentence- or passage-level representations.

Comparison with First-k. Figure 5.3 also shows the results of First-k (the red lines) described in Section 5.3.1. This is to compare SG with mLMs that have the same number of word embedding entries by adopting a smaller vocabulary size. As the figure shows, while First-k could still share similar effectiveness with SG at $r_G = 40\%$, removing more subwords from the vocabulary deteriorates the effectiveness drastically on all downstream tasks, regardless of whether continual pretraining is applied. This pair of results suggests that while simply reducing the number of subwords has a detrimental effect on the mLMs capacity, which echoes previous findings [23, 68, 3, 123], not all subwords require a unique representation. In other words, the effect of the number of the word embedding entries should be disentangled from the size of the vocabulary.

Enhance the semantic similarity via post-hoc operations. In the above results,

per language is similar, we do not observe a consistent impact over languages across different benchmarks.

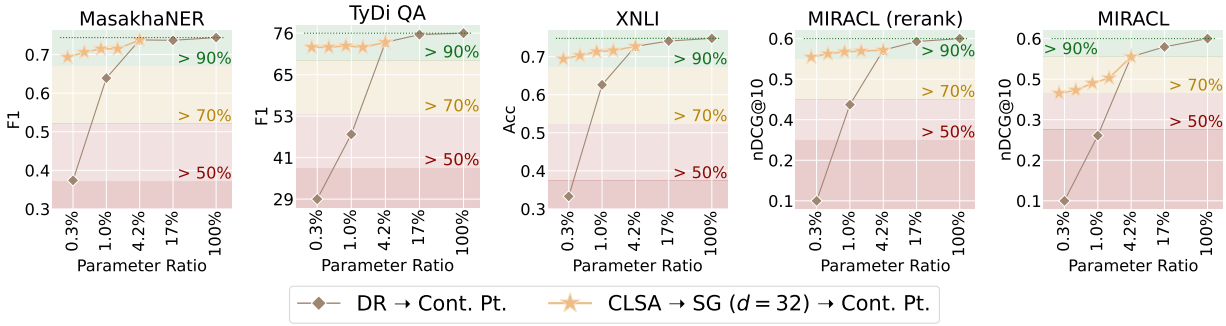


Figure 5.6: Comparison the effectiveness of Dimension Reduction (DR) and semantic Grouping (SG) at the same level of word embedding parameters size. **x-axis**: the ratio of the word embedding size after DR or SG in log scale. The brown lines: DR with dimension $d \in \{2, 8, 32, 128, 768\}$ and full vocabulary, corresponding to embedding parameters ratio of {0.3%, 1.0%, 4.2%, 17%, 100%}. The yellow lines: SG with dimension $d = 32$ and vocabulary grouping ratio $r_G \in \{5\%, 10\%, 20\%, 40\%\}$, corresponding to embedding parameters ratio of {0.21%, 0.42%, 0.84%, 1.7%, 4.2%}. Background colors design is identical to Figure 5.3.

the semantic grouping is applied on the off-the-shelf mLMs, exploiting the spatial structure of the untouched word embedding. *Could the similarities of tokens under the same semantic concepts be further enhanced by post-hoc operations on the mLMs?* We show that this direction is possible and promising, using the cross-lingual subword alignment (CLSA) operation as an example.

Figure 5.4 compares the results of applying SG on mBERT with and without CLSA, where the orange lines are consistently higher than or similar to the blue ones regardless of whether the model has been continually pretrained. On certain datasets, e.g., XNLI and MIRACL, CLSA brings visible improvement at all grouping ratios, with more significant improvements at lower grouping ratios on the other datasets. This suggests that post-hoc operations on embedding geometry can strengthen the semantic similarity among subwords.[12]

[12]Note that CLSA is used as a proof-of-concept instead of a general solution: As mentioned in Section 5.3.1, CLSA is applied to words that are tokenized into single subword, which limits the coverage on the entire vocabulary intrinsically. Thus instead of promoting the CLSA method itself, we use its success in enhancing the semantic similarity as a proof-of-concept that word embeddings can support further post-hoc representation enhancement.

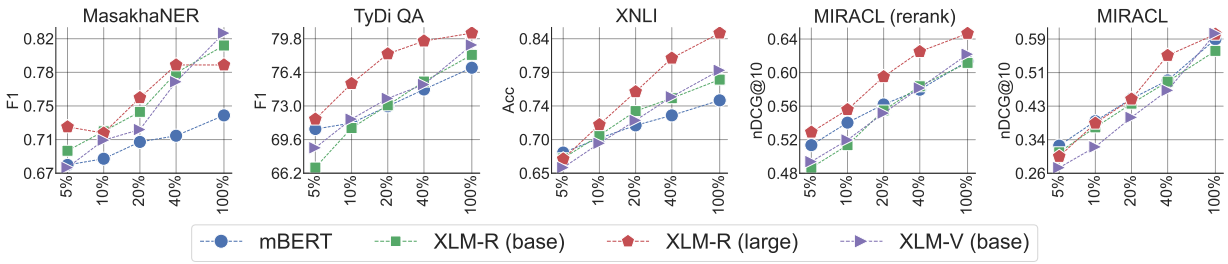


Figure 5.7: Results of SG on multiple backbones, all with CLSA applied and no continual pretraining. All models show similar trend regarding semantically grouped subwords.

5.4.2 Confounding Effect from Embedding Parameters

While applying the SG, the number of the word embedding parameters is also reduced by the same degree. Do the changes on effectiveness truly stem from the semantic grouping, or any form of word embedding parameters reduction have the same effect? To address the concern, we apply SG on word embeddings with reduced embedding dimension via the dimension reduction (DR) operation, to see whether the effectiveness diminishes as the embedding parameters are reduced. We first show that SG maintains the same level of effectiveness on the reduced dimensions. Moreover, while the embedding dimension has reached its limit, SG could further push down the overall word embedding parameters to a level that could not be achieved by reducing the word embedding dimension alone.

SG maintains the same level of effectiveness with reduced embedding dimensions. Figure 5.5 shows the results of SG with three word embedding dimensions $d \in \{768, 128, 32\}$, with both CLSA and continual pretraining applied. While DR reduces the effectiveness of the mLM with full vocabulary, the slope of the curve flattens as the dimension reduces, indicating less relative effectiveness drop on the downstream tasks at lower embedding dimension d. Specifically, the SG results on embedding dimension $d = 32$ is on par and even outperform the results on the initial dimension $d = 768$ as vocabulary size reaches 5%. These results show that the effect of SG is largely independent from the embedding parameters.

SG achieves lower number of embedding parameters beyond dimension reduction. Apart from the above observation, Figure 5.6 unites the results of SG at embedding dimension $d = 32$ and DR from the perspective of the total number of word embedding parameters (x-axis). The brown lines show the DR results of reducing the embedding dimension d from 768 to $\{2, 8, 32, 128\}$ with the vocabulary untouched, yielding embedding parameter ratios of {0.3%, 1.0%, 4.2%, 17%}. With DR, the effectiveness on all

downstream tasks is largely preserved until dimension $d = 32$ and drops sharply once the dimension falls beneath it, indicating that saving the embedding parameters via dimension reduction has reached its limit.

On the other hand, the yellow lines show the models starting from word embedding with dimension $d = 32$ (thus 4.2% embedding parameters) and then applying SG with grouping ratio r_G from 40% to 5%,[13] yielding embedding parameter ratios from 4.2% to 0.21%. The effectiveness differences between the two lines are clear: while further reducing the embedding parameters by similar scale, SG could largely preserve the downstream effectiveness whereas DR fails miserably. This suggests that SG is complementary to DR in its effect on word embedding parameters, and that it may provide a new perspective on understanding how much vocabulary and embedding capacity is necessary in multilingual word embeddings.

5.4.3 Backbones

The experiments above are all based on mBERT [31]. How would the findings generalize to mLMs with different tokenization algorithms, vocabulary size, model size, and pre-training corpora? We select three additional mLMs to address the above concern: XLM-R (base), XLM-R (large) [23], and XLM-V (base) [68], which all deploy ULM [60] to construct the vocabulary and are pretrained on CC100, but differ from each other in terms of model sizes, tokenization pre-processing, vocabulary allocation, and vocabulary size.

Results of SG are shown in Figure 5.7, where background colors are omitted as different backbones do not share the same oracle results. Instead, we compare the results of the other backbones with mBERT (the blue lines) to see whether they follow similar trends as more subwords are semantically grouped. Overall, the slopes of the four curves are similar on all five benchmarks, indicating that the findings are likely to generalize over different multilingual LMs with various tokenization processes and model sizes.

5.4.4 Semantic Categories among Grouped Subwords

This section provides insights into the semantic similarities among grouped subwords. Figure 5.8 illustrates examples of eight semantic groups, which are based on the model with embedding dimension $d = 768$ and the application of CLSA.[14]

[13] Identical to the yellow lines in Figure 5.5.

[14] Note that only parts of the subwords are displayed per group due to space limitations. For the full list of subwords, please refer to Figure 5.11.

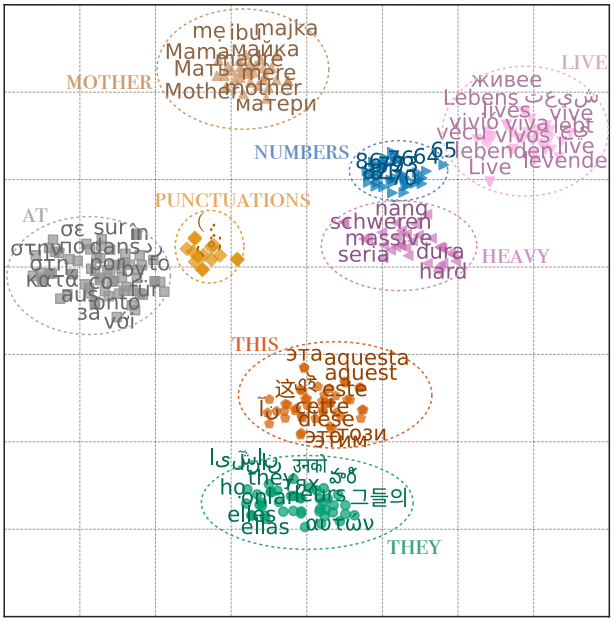


Figure 5.8: Eight semantic tokens formed by grouped subwords on mBERT word embedding with dimension $d = 768$ and CLSA applied. The labels outside the circles are either the keyword of the cluster, or "`NUMBERS`"/ "`PUNCTUATION`" if the cluster is a collection of numbers or punctuation.

Among the eight displayed example groups, six of them are selected from the Swadesh list [122], a widely used compilation of basic concepts across languages. The keywords are selected to cover various parts of speech, including pronouns, nouns, adjectives, prepositions, and verbs. The remaining two groups are labeled as "`NUMBERS`" and "`PUNCTUATION`", which exhibit strong clustering upon manual inspection. Overall, we identify several patterns of semantic similarities among the grouped subwords:

1. Semantically identical or similar words across languages: *mother vs. mẹ , they vs.* 그들은 .[15]
2. Semantically related words from the same language: *at vs. on, heavy vs. hard.*
3. Semantically related words across languages: *heavy vs. intenso, who vs. που*.[16]
4. Numbers in similar range: *the cluster numbers ranging from 60 to 100.*
5. Punctuation: *., ;!*

[15] *mẹ*: "mother" in Vietnamese; *Cette*: "this" in French; 그들은 : "they" in Korean.

[16] *intenso*: "intense" in Spanish; *που*: "where" in Greek.

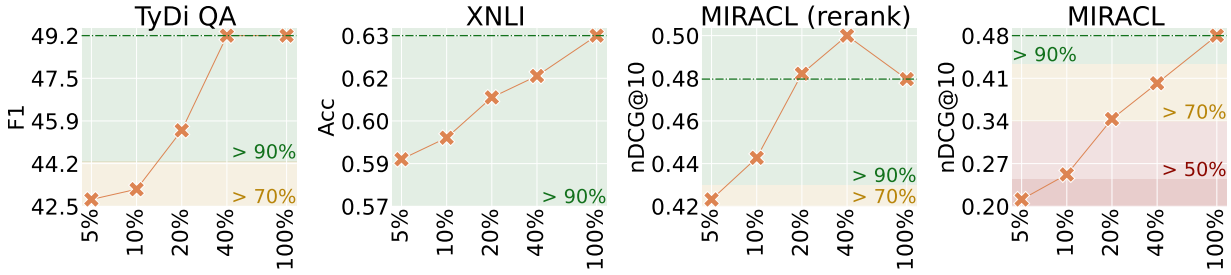


Figure 5.9: Results when transferring from English to other languages on four benchmarks, with CLSA and continual pretraining applied. MasakhaNER is skipped as English training data is not available. Background colors design is identical to Figure 5.3. The dashed green line denotes the zero-shot performance without semantic grouping.

We also notice that some words are not grouped based on desired semantics: The current methods are limited in the polysemous situation and controlling the desired semantic per group: For example, in the group of "I", while the desired semantic is the first person singular pronouns in different languages, the group mainly includes the single letter such as "A", "B", "C" on mLMs with $d = 768$, as shown by Figure 5.11. suggesting that future work is necessary to better handle such cases.

5.4.5 Cross-lingual Transfer

We investigate how the semantic grouping affects the cross-lingual transferability from English to other languages. We evaluate on four of the above tasks, where MasakhaNER is skipped in this section as English training data is not available.

On two classification tasks: TyDiQA and MIRACL (rerank), the zero-shot results at grouping ratio $r_G \in \{40\%, 20\%\}$ are on par or even better than the oracle results, where subwords and embeddings are not semantically grouped. That is, fine-grained semantic tokens show similar cross-lingual transferability to the original subword vocabulary in some circumstances, suggesting that they may serve as anchors for cross-lingual transfer in current language models.

On the other hand, cross-lingual transfer on embedding tasks is more challenging especially with small group ratio, i.e., coarse-grained semantic tokens. While the zero-shot results on MIRACL at $r_G \in \{20\%, 40\%\}$ remains in the reasonable range — achieving more than 70% of the relative zero-shot effectiveness — the relative zero-shot results falls under 50% when r_G further drops. This mirrors the in-domain results from previous sections that

embedding tasks require more fine-grained semantic information to form effective sentence or paragraph representations.

5.4.6 Ablation on Distance Metric

We compare the results using cosine versus Euclidean distance in Figure 5.10, where the results show that grouping on Euclidean distance greatly underperformance cosine distance especially at higher dimension ($d = 768$). We interpret this as that the vector norm is an undesired feature when pursuing the semantic similarity between the subwords, which amplifies the distance between semantic similar subword at high dimension.

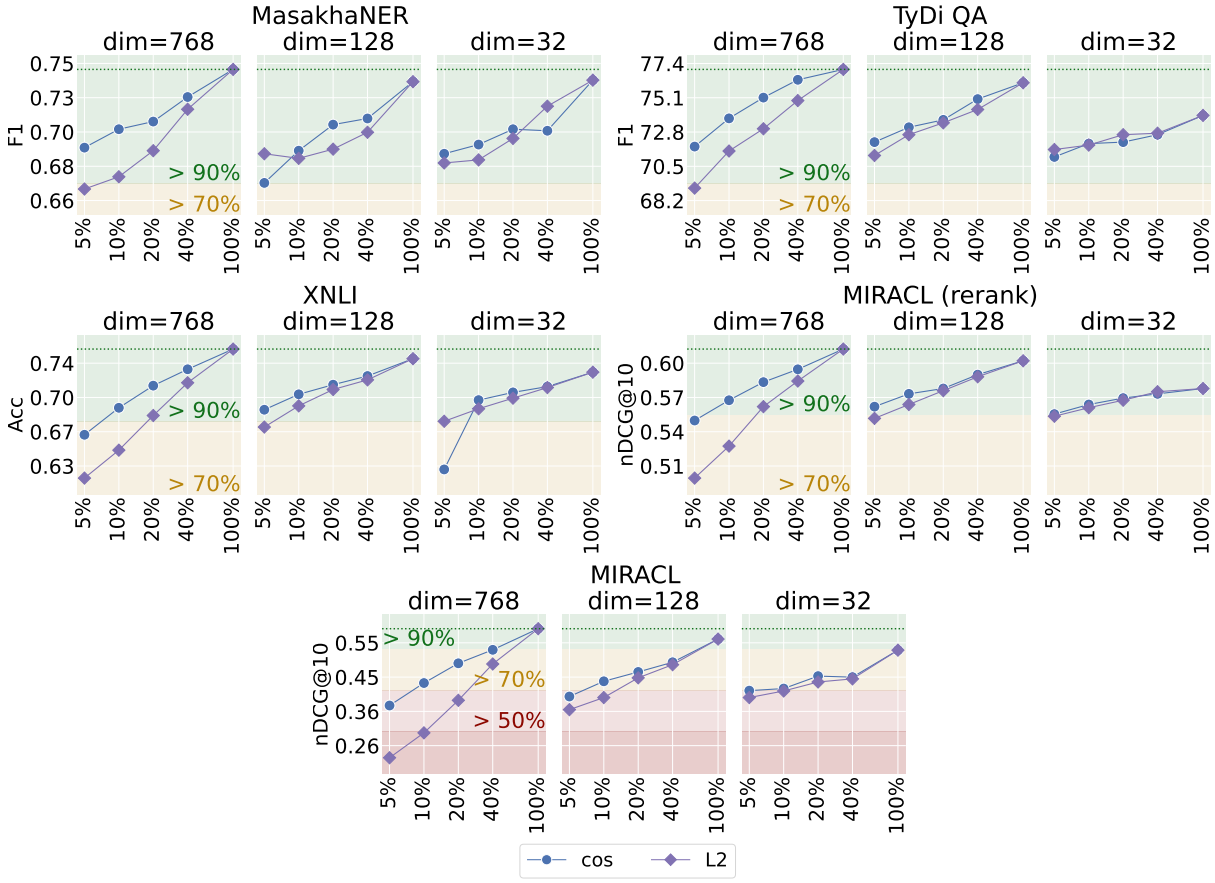


Figure 5.10: Comparison of L2 versus cosine distance when using K-Means to group the subwords.

5.5 Discussion on the Behaviors Difference across Tasks

Finally, we discuss the behavioral differences between classification and embedding tasks, which are orthogonal to the main findings of this work on token-level semantic concepts. Our experiments cover four classification tasks and one embedding task (Table 5.4). Across all results, we observe that, to achieve the same level of effectiveness (e.g., 90%), classification tasks require only coarse semantic representations for prediction (e.g., around 5% of the original vocabulary size), whereas embedding tasks require more fine-grained lexical representations to form effective sentence or paragraph representations (e.g., more than 20% of the original vocabulary size). This reveals that tasks have different requirements on the information level to achieve similar effectiveness. In other words, some can tolerate higher semantic distortion, while others are more vulnerable. Although the four classification tasks differ in domain, input length, and language, we observe much smaller variation among them than between the classification tasks and the embedding task.

One possible hypothesese is that classification tasks map inputs to a relatively small, discrete output space, and may therefore models remain effective as long as coarse task-relevant distinctions are preserved. In contrast, embedding tasks require independently encoded queries and passages to remain geometrically comparable over a large candidate collection. Semantic grouping may distort many fine-grained distinctions that are unnecessary for classification but important for ranking *closely related passages*. Note that we leave this as a task-structure-based hypothesis rather than a tested conclusion, leaving the controlled experiments and verification to future work.

A similar discrepancy across tasks has also been observed in machine translation and understanding tasks [117]. When all bilingual documents are removed from the pretraining corpus (roughly 2% of the data), BLEU scores on the evaluated machine translation benchmarks drop by more than half, whereas performance on cross-lingual QA and other general reasoning tasks remains stable. These findings suggest that our analyses and interpretations on model behaviors should be evaluated on diverse types of tasks to ensure that the conclusions are not biased by the nature of a particular task.

5.6 Conclusion

Inspired by how human understand text based on semantic concepts rather than superficial forms, this work studies the degree to which current multilingual language models understand based on subword-level semantic concepts. We find that the general shared

semantics could get the models a long way in understanding languages and making predictions, especially for classification tasks. Additional experiments show that the observations generalize across mLMs with different tokenization algorithm, vocabulary size, model size, and pre-training corpora. Inspections on the grouped subwords suggest that they exhibit multiple patterns of semantic similarities, including synonyms and word translations in many languages.

Not only are subword-level semantics prominent in in-domain language understanding, in some cases, they also serve as anchors of cross-lingual transfer. The results reveal practical opportunities for vocabulary compression and post-hoc representation enhancement, while also helping explain how multilingual models may share meaning across languages. We hope that this work sheds light on understanding the multilingual vocabulary and word embeddings from the semantic perspective, and spurs further research on shared semantic information at the subword level across languages.

Keywords	dim=768	dim=128	dim=32
I	A, B, C, CH, E, H, I, M, N, O, Q, S, V, X, a, d, f, h, j, m, n, p, pm, t, ös, A, O, A, O, м, o, अ, অ, আ, ਮ, ಅ, ಅ	##इ, I, ich, mu, sì, yo, yá, бы, بی, من, میں, तब, तो, ही, ত, তখন, তা, ఈ, 予, 俺, 汝, 私, 저	I, Ich, Ja, ich, lord, me, mich, minut, mir, mis, mo, my, més, mí, self, sí, tôi, 们, 俺, 分, 别, 我, 来, 汝, 私
they	They,chúng,ellas,elles,ellos,eux,heidän,hulle, họ,ihnen,ils,leur,leurs,loro,mereka,niiden, niitä,onlar,onlara,onları,onların,their,them, they,αυτών,τους,им,их,они,тях,ایشان,آنها,آنان, उनकी,उनके,उनको,वे,এদের,তাদের,তারা,వారి,그들은, 그들의	Ihre,Leur,Sus,Their,chúng,deres,elles,ellos, euren,eux,heidän,ih,ihnen,ihre,ihrem,ihren, ihrer,ihres,ils,its,leur,leurs,loro,niiden,niitä, oni,onlar,onlara,onları,onların,su,sus,suya, suyos,their,them,they,τους,ги,ими,их,они, свои,тях,هم,ایشان,ان,آنها,آنان,उनकी,उनको,उन्हें,ये, এদের,তাদের,వారి,వారు,그녀의,그들은,그들의	Elles,Lui,They,chúng,dier,elles,estén,eux,,هم## hänelle,hänellä,ihnen,ils,leur,niitä,noi,onlar, onlara,onları,them,they,οποίες,τους,ги,ими,их, उनकी,उन्हें,তাদের,ఇవి,,هم,ایشان,آنها,آنان,них,они,тях వారు,これに,그들은
this	aquest, aquesta, cette, diese, diesem, dieser, dieses, esta, este, esto, eta, ini, nàyy, questa, questo, this, tämä, tämän, tätä, ésta, éste, тази, това, този, эта, этим, это, этот, ,این ,آن هذه ,هذا ,ذلك, এই, এটি, ఇది, ఈ, この, その, 之, 其, 当, 此, 这, 這	##この, ##その, Esta, This, diese, este, this, tämän, tätä, тази, тези, това, този, эта, эти, этой, этот, эту, ఈ, この, 斯, 此, 该, 这	Cette, Diese, Dieser, Esta, Estas, Meine, This, ces, diese, dieser, dieses, estas, este, esto, estos, ini, this, tämä, tämän, tätä, Эта, Эти, тази, тези, това, този, эти, этот, эту, هذه و, ఈ, この, これらの, 这
who	koji, mà, que, qui, that, who, yang, što, που, което, които, който, която, koj, что, што, що, كه ,جو ,אשר, जो, কে, যা, だと, 하는	Who, habían, jotka, kdo, ki, koga, quién, who, Кто, Что, кого, кое, кои, кто, що, كه, যিনি, 誰, 谁	cual, cuales, joita, joka, jota, jotka, lesquels, mitä, mà, que, qui, quien, quienes, welche, welcher, welches, which, who, yakni, yang, οποία, οποίο, οποίος, который, جو ,والتي ,التي ,الذي والذي ,كه, जिसे, जे, जो, だと
heavy	Heavy, ağır, berat, ciddi, difficile, difficult, difícil, dura, dure, duro, greu, hard, heavy, khó, leve, massive, nhẹ, nặng, pesante, schwer, schwere, schweren, seria, serious, seriously, severe, teško, zwaar, теж, 艱, 重, 难	Heavy, ağır, berat, deras, gravemente, heavy, leve, massive, nặng, pek, pesante, schwer, schwere, schweren, seriously, starke, teško, большим, теж, مشکل ,گرم ,شدید, 沉, 重, 큰	Greatest, Heavy, agir, ağır, deras, forte, giảm, heavy, intensa, intense, intenso, massive, nặng, pesante, plein, profonde, schwere, schweren, starke, большим, شدید, á, ő, 重, 커, 큰
fire	Feuer, Smoke, dust, feu, fire, fires, fogo, fuego, fuoco, humo, incendie, incendio, lửa, smoke, tobacco, النار ,آتش, 拂, 炎, 烟, 煙, 熏, 燻, 莟, 雾, 霧	Brand, Feuer, Fire, Fuego, ag, api, fai, feu, fire, fuego, incendio, llama, lửa, огонь, النار ,آتش, 火, 炎, 焰	##selen, ##サス, Fire, Fuego, Riot, bão, earthquake, fire, fuego, hurricane, incendio, lửa, neste, storm, terremoto, trận, tsunami, النار, 炎, 焰
mother	Mama, Mother, Mutter, ibu, ibunya, madre, majka, mama, mater, moeder, mother, māe, mère, mẹ, nurse, μητέρα, Мать, майка, матери, мати, мать, мајка, مادر ,ام ,أم, মা, 媽, 嬤, 嬤, 母, 어머니	Mama, Mother, Mutter, grandmother, ibunya, madre, majka, mama, mother, māe, mère, nang, tante, μητέρα, Мать, майка, матери, мать, мајка, مادر, 亲, 妈, 婆, 媽, 嬤, 嬤, 母, 어머니	Father, Maker, Mama, Mother, Robin, mother, Мать, баща, майка, матери, мать, مادر, 娃, 母, 父, 어머니
live	Lebens, Live, alive, hidup, leben, lebenden, lebt, lebte, lebten, levde, levende, live, lived, lives, living, sống, viva, vivant, vive, viven, vivent, vivió, vivo, vivos, vivre, vécu, živе, živi, жив, живее, живи, زنده ,حي ,تعيش, ライブ	Lebens, Live, alive, asui, elää, habita, hidup, leben, lebenden, lebt, lebte, lebten, live, lived, living, rege, sağ, sống, vital, viva, vivant, vive, viven, vivent, vivere, vivien, vivir, vivió, vivo, vivos, vivre, wonen, Žije, žije, žive, živi, жив, живе, живее, живеят, живи, жил, жили, существование, يعيش ,حي ,تعيش ,חיים, ライブ, 寿, 活	elää, errichten, existieren, existir, ganado, hidup, leben, lebten, live, living, pequeños, reside, resides, residing, sống, vivant, vive, viven, vivent, vivir, vivo, vivos, vivre, yaşayan, život, живеят, живи, съществование, تعيش ,يعيش, 住
at	at,au,auf,aus,by,dans,en,for,für,in,into,na, nach,on,onto,over,por,på,sobre,sur,to,trên,tại, upon,vào,vói,with,à,în,από,για,εν,επί,κατά,με, σε,στα,στη,στην,στις,στο,στον,στους,в,во,във,за, из,к,към,на,над,по,с,co,у,در ,برای ,با,ใน,ŏ,で,に, を,在,에	##ที่,aikaa,an,at,kepada,nσi,terhadap,untuk, έτος,κατά,στη,στις,στο,στον,στους,близ ,بر,ओर, సాల,ವಿಂದಿ,ವರ್ಷ,సంవత్సరం,จาก,ณ,ที่,เมื่อ,แห่ง,ใน,デ, 在,에,에서	at,atas,dalam,dans,dessus,in,inside,ninu,trên, tại,untuk,upon,varten,κατά,προς,σε,στα,στη, στην,στις,στο,στον,στους,близ,в,във,на,свыше, согласно ,بر,จาก,ณ,เมื่อ,ใน,にかけて,まで,在,에, 에서,위에,으로
80	63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96, 97, 98, 99	65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96, 97, 98, 99	66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96, 97, 98, 99
:	; ,: ,, ,) ,。 ,、 ,։, ,‘, ,; ,: ,/ ,.. ,- , , ,)	; ,: ,, ,) ,、 ,։, ,‘, ,; ,: , , ,)	##○:, ##○ः, :, ;, Her, Hän, ։, ！, :, ;

Figure 5.11: More examples of the grouped subwords on mBERT with CLSA

d	r_G	w/ CLSA?	w/ PT?	MasakhaNER		TyDiQA		XNLI		MIRACL (rerank)		MIRACL	
				F1	perc	F1	perc	Acc	perc	nDCG@10	perc	nDCG@10	perc
768	100%	–	×	0.735	98.7%	76.9	99.9%	0.750	99.9%	0.613	99.5%	0.586	98.7%
768	100%	–	✓	0.745	100.0%	77.0	100.0%	0.751	100.0%	0.616	100.0%	0.594	100.0%
						Figure 5.3							
768	40%	×	×	0.704	94.5%	74.6	96.9%	0.725	96.5%	0.584	94.8%	0.493	83.0%
768	20%	×	×	0.698	93.7%	72.6	94.3%	0.702	93.5%	0.553	89.8%	0.436	73.4%
768	10%	×	×	0.680	91.3%	71.3	92.6%	0.678	90.3%	0.523	84.9%	0.372	62.6%
768	5%	×	×	0.659	88.5%	70.2	91.2%	0.650	86.6%	0.495	80.4%	0.306	51.5%
768	40%	×	✓	0.727	97.6%	76.3	99.1%	0.730	97.2%	0.597	96.9%	0.533	89.7%
768	20%	×	✓	0.711	95.4%	75.1	97.5%	0.713	94.9%	0.585	95.0%	0.494	83.2%
768	10%	×	✓	0.706	94.8%	73.7	95.7%	0.690	91.9%	0.568	92.2%	0.437	73.6%
768	5%	×	✓	0.694	93.2%	71.8	93.2%	0.662	88.1%	0.549	89.1%	0.372	62.6%
						Figure 5.4							
768	40%	✓	×	0.712	95.6%	74.7	97.0%	0.729	97.1%	0.580	94.2%	0.487	82.0%
768	20%	✓	×	0.705	94.6%	73.0	94.8%	0.715	95.2%	0.563	91.4%	0.443	74.6%
768	10%	✓	×	0.686	92.1%	71.3	92.6%	0.697	92.8%	0.542	88.0%	0.389	65.5%
768	5%	✓	×	0.679	91.1%	70.7	91.8%	0.677	90.1%	0.516	83.8%	0.330	55.6%
768	40%	✓	✓	0.724	97.2%	76.5	99.4%	0.741	98.7%	0.602	97.7%	0.540	90.9%
768	20%	✓	✓	0.722	96.9%	74.8	97.1%	0.726	96.7%	0.590	95.8%	0.506	85.2%
768	10%	✓	✓	0.714	95.8%	74.1	96.2%	0.707	94.1%	0.570	92.5%	0.442	74.4%
768	5%	✓	✓	0.691	92.8%	73.0	94.8%	0.690	91.9%	0.566	91.9%	0.398	67.0%
						Figure 5.5							
128	40%	✓	✓	0.717	96.2%	75.2	97.7%	0.725	96.5%	0.591	95.9%	0.509	85.7%
128	20%	✓	✓	0.714	95.8%	74.5	96.8%	0.721	96.0%	0.585	95.0%	0.469	79.0%
128	10%	✓	✓	0.698	93.7%	73.3	95.2%	0.708	94.3%	0.577	93.7%	0.451	75.9%
128	5%	✓	✓	0.695	93.3%	72.9	94.7%	0.694	92.4%	0.568	92.2%	0.415	69.9%
32	40%	✓	✓	0.712	95.6%	74.7	97.0%	0.729	97.1%	0.580	94.2%	0.487	82.0%
32	20%	✓	✓	0.705	94.6%	73.0	94.8%	0.715	95.2%	0.563	91.4%	0.443	74.6%
32	10%	✓	✓	0.686	92.1%	71.3	92.6%	0.697	92.8%	0.542	88.0%	0.389	65.5%
32	5%	✓	✓	0.679	91.1%	70.7	91.8%	0.677	90.1%	0.516	83.8%	0.330	55.6%
						Figure 5.6							
128	100%	–	✓	0.737	98.9%	76.1	98.8%	0.741	98.7%	0.605	98.2%	0.564	94.9%
32	100%	–	✓	0.735	98.7%	76.9	99.9%	0.750	99.9%	0.613	99.5%	0.586	98.7%
8	100%	–	✓	0.639	85.8%	47.8	62.1%	0.626	83.4%	0.414	67.2%	0.276	46.5%
2	100%	–	✓	0.374	50.2%	29.4	38.2%	0.333	44.3%	0.123	20.0%	0.064	10.8%
						Figure 5.7: XLM-R base							
768	100%	✓	×	0.814	100.0%	78.2	100.0%	0.779	100.0%	0.611	100.0%	0.558	100.0%
768	40%	✓	×	0.783	96.2%	75.5	96.5%	0.753	96.7%	0.584	95.6%	0.484	86.7%
768	20%	✓	×	0.739	90.8%	73.1	93.5%	0.735	94.4%	0.556	91.0%	0.431	77.2%
768	10%	✓	×	0.717	88.1%	70.8	90.5%	0.701	90.0%	0.516	84.5%	0.374	67.0%
768	5%	✓	×	0.695	85.4%	66.8	85.4%	0.669	85.9%	0.490	80.2%	0.314	56.3%
						Figure 5.7: XLM-R large							
1024	100%	✓	×	0.792	100.0%	80.4	100.0%	0.844	100.0%	0.645	100.0%	0.598	100.0%
1024	40%	✓	×	0.792	100.0%	79.6	99.0%	0.809	95.9%	0.624	96.7%	0.547	91.5%
1024	20%	✓	×	0.755	95.3%	78.3	97.4%	0.762	90.3%	0.595	92.2%	0.442	73.9%
1024	10%	✓	×	0.715	90.3%	75.3	93.7%	0.716	84.8%	0.557	86.4%	0.384	64.2%
1024	5%	✓	×	0.722	91.2%	71.7	89.2%	0.668	79.1%	0.531	82.3%	0.304	50.8%
						Figure 5.7: XLM-V base							
768	100%	✓	×	0.828	100.0%	79.2	100.0%	0.792	100.0%	0.621	100.0%	0.600	100.0%
768	40%	✓	×	0.773	93.4%	75.2	94.9%	0.755	95.3%	0.582	93.7%	0.463	77.2%
768	20%	✓	×	0.719	86.8%	73.8	93.2%	0.722	91.2%	0.553	89.0%	0.398	66.3%
768	10%	✓	×	0.707	85.4%	71.7	90.5%	0.690	87.1%	0.522	84.1%	0.327	54.5%
768	5%	✓	×	0.676	81.6%	68.8	86.9%	0.656	82.8%	0.497	80.0%	0.277	46.2%

Table 5.5: Numerical results of experiments, where each is an averaged result of all languages per benchmark. The colors indicate the relative performance to the oracle results: green: >90%, yellow: 70%–90%, red: 50%–70% and the dark red color means <50%.

Chapter 6

Conclusion

This thesis shares projects on multilingual embedding models for retrieval from three connected perspectives. The work begins from the need for stronger multilingual retrieval resources, moves to practical training strategies under different resource conditions, and then examines the mechanisms behind cross-lingual transfer. The first part of the thesis introduced Mr. TYDI and MIRACL, two multilingual retrieval resources that provide large-scale query-passage supervision, careful relevance judgments, and broader language coverage. The second part examined training practices for multilingual dense retrievers, showing how effectiveness depends on the interaction between pretrained backbones, target-language data, translated data, distillation, and cross-lingual transfer. The third part analyzed shared tokens and token-level semantic concepts, suggesting that transfer is supported by concrete lexical and semantic structures inside multilingual vocabularies and embedding spaces.

Together, these contributions move multilingual retrieval research from infrastructure, to practice, to interpretation, which I hope can help support continuing work on multilingual information access systems to provide reliable access to information across more languages and domains. At the same time, they leave open several directions for future work on reducing the data and compute required to extend these systems to new languages:

Language Adaptation with Minimal Data and Computational Cost. The dominant paradigm today is still data-centric: models are scaled from English outward by collecting more manual or synthetic multilingual data, adding more languages during pretraining, and repeating task-specific post-training in each language. This has produced

progress, but it creates barriers for low-resource languages and for researchers who cannot afford to retrain large models for every language they want to support.

Language-Representation Efficiency. Other than model effectiveness, language representation efficiency is also a crucial factor to support equivalent information access. With BPE-based tokenization as the mainstream tokenization algorithm, languages that are less-represented in the tokenizer training corpora naturally disadvantaged in language representation learning, that text with equivalent semantic meaning requires more tokens to represent. This is more severe for languages non-Latin scripts as they require more bytes per character to represent, and becoming more severe when models are expected to support long context inputs in increasing convoluted tasks.

Multi-Perceptual Modeling. While some languages are better connected via shared scripts or spellings, others may be connected more strongly through phonetic or other perceptual similarities. Multi-perceptual modeling could help multilingual systems represent these relationships more explicitly, and potentially improving cross-lingual transfer efficiency by leveraging the similarity patterns across different perceptual dimensions.

References

[1] David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131, 2021.

[2] Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. jina-embeddings-v5-text: Task-targeted embedding distillation. *arXiv preprint arXiv:2602.15547*, 2026.

[3] Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, Charvi Jain, Alexander Weber, Lena Jurkschat, Hammam Abdelwahab, Chelsea John, Pedro Ortiz Suarez, Malte Ostendorff, Samuel Weinbach, Rafet Sifa, Stefan Kesselheim, and Nicolas Flores-Herr. Tokenizer choice for LLM training: Negligible or crucial? In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association*

for Computational Linguistics: NAACL 2024, pages 3907–3924, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[4] Peter M. Arkadiev. Morphology in typology: Historical retrospect, state of the art, and prospects. 2020.

[5] Jordi Armengol-Estapé, Casimiro Pio Carrino, Carlos Rodriguez-Penagos, Ona de Gibert Bonet, Carme Armentano-Oller, Aitor Gonzalez-Agirre, Maite Melero, and Marta Villegas. Are multilingual models the best choice for moderately under-resourced languages? A comprehensive assessment for Catalan. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4933–4946, Online, August 2021. Association for Computational Linguistics.

[6] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas, November 2016. Association for Computational Linguistics.

[7] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online, July 2020. Association for Computational Linguistics.

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

[9] Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. One question answering model for many languages with cross-lingual dense passage retrieval. In *Advances in Neural Information Processing Systems*, volume 34, pages 7547–7560, 2021.

[10] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268v3*, 2018.

[11] Lisa Ballesteros and W. Bruce Croft. Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 64–71, New York, NY, USA, 1998. Association for Computing Machinery.

[12] Ankur Bapna and Orhan Firat. Simple, scalable adaptation for neural machine translation. 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 1538–1548, Hong Kong, China, November 2019. Association for Computational Linguistics.

[13] Luiz Henrique Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. mmarco: A multilingual version of ms marco passage ranking dataset. *arXiv:2108.13897*, 2021.

[14] Leonid Boytsov, David Novak, Yury Malkov, and Eric Nyberg. Off the beaten path: Let's replace term-based retrieval with k-NN search. In *Proceedings of 25th International Conference on Information and Knowledge Management*, pages 1099–1108, Indianapolis, Indiana, 2016.

[15] Eleftheria Briakou, Colin Cherry, and George Foster. Searching for needles in a haystack: On the role of incidental bilingualism in PaLM's translation capability. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9432–9452, Toronto, Canada, July 2023. Association for Computational Linguistics.

[16] Alan Bundy and Lincoln Wallen. *Semantic Primitives*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1984.

[17] Steven Cao, Nikita Kitaev, and Dan Klein. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*, 2020.

[18] Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. Goldfish: Monolingual language models for 350 languages, 2026.

[19] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[20] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020.

[21] Cyril Cleverdon. *The Cranfield tests on index language devices*, page 47–59. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[22] Cohere. Cohere's embed models (details and application), 2025. Accessed: 2026-05-20.

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

[24] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *ArXiv*, abs/1710.04087, 2017.

[25] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[26] Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online, July 2020. Association for Computational Linguistics.

[27] Nick Craswell, Bhaskar Mitra, Daniel Campos, Emine Yilmaz, and Jimmy Lin. MS MARCO: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 1566–1576, 2021.

[28] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2020 deep learning track. *arXiv:2102.07662*, 2021.

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

[30] Hope Dawson, Antonio Hernandez, and Cory Shain. Morphological types of languages. In *Language Files: Materials for an Introduction to Language and Linguistics, 12th Edition*. Department of Linguistics, The Ohio State University, 2016.

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

[32] Junggeun Do, Jaeseong Lee, and Seung-won Hwang. ContrastiveMix: Overcoming code-mixing dilemma in cross-lingual transfer for information retrieval. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 197–204, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[33] Philipp Dufter and Hinrich Schütze. 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, November 2020. Association for Computational Linguistics.

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

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

[36] Mattia Fumagalli, Roberta Ferrario, et al. Representation of concepts in ai: Towards a teleological explanation. In *JOWO*, 2019.

[37] Liane Gabora, Eleanor Rosch, and Diederik Aerts. Toward an ecological theory of concepts. *Ecological Psychology*, 20(1):84–116, 2008.

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

[39] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Tevatron: An efficient and flexible toolkit for neural retrieval. In *Proceedings of the 46th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3120–3124, Taipei, Taiwan, 2023.

[40] Peter Gardenfors. *The geometry of meaning: Semantics based on conceptual spaces*. MIT press, 2014.

[41] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, pages 7780–7788, New York, New York, 2020.

[42] Daniela Gerz, Ivan Vulić, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 316–327, Brussels, Belgium, October-November 2018.

[43] Cliff Goddard and Anna Wierzbicka. *Words and Meanings: Lexical Semantics Across Domains, Languages, and Cultures*. Oxford University Press, 11 2013.

[44] Joseph Harold Greenberg. A quantitative approach to the morphological typology of language. *International Journal of American Linguistics*, 26:178–194, 1960.

[45] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[46] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

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

[48] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 113–122, 2021.

[49] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4411–4421, 2020.

[50] Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, et al. A survey on large language models with multilingualism: Recent advances and new frontiers. *Artificial Intelligence Review*, 2026.

[51] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

[52] Ray Jackendoff. Conceptual semantics. *Meaning and mental representations*, 1:81–97, 1988.

[53] Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online, November 2020. Association for Computational Linguistics.

[54] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv:1702.08734*, 2017.

[55] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

[56] Alexander Jones, William Yang Wang, and Kyle Mahowald. A massively multilingual analysis of cross-linguality in shared embedding space. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5833–5847, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[57] Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*, 2020.

[58] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

[59] Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*, pages 39–48, 2020.

[60] Taku Kudo and John Richardson. 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, November 2018. Association for Computational Linguistics.

[61] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[62] Carlos Lassance. Extending English IR methods to multi-lingual IR. *arXiv:2302.14723*, 2023.

[63] Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 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, November 2020. Association for Computational Linguistics.

[64] Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldaini, and Eugene Yang. Overview of the TREC 2022 NeuCLIR track. In *Proceedings of the 31st Text REtrieval Conference*, 2023.

[65] Gennadi Lembersky, Noam Ordan, and Shuly Wintner. Language models for machine translation: Original vs. translated texts. *Computational Linguistics*, 38(4):799–825, 2012.

[66] Jiaang Li, Yova Kementchedjhieva, Constanza Fierro, and Anders Søgaard. Do Vision and Language Models Share Concepts? A Vector Space Alignment Study. *Transactions of the Association for Computational Linguistics*, 12:1232–1249, 09 2024.

[67] Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 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, July 2023.

[68] Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore, December 2023. Association for Computational Linguistics.

[69] Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. How language-neutral is multilingual bert? *CoRR*, abs/1911.03310, 2019.

[70] Jimmy Lin, David Alfonso-Hermelo, Vitor Jeronymo, Ehsan Kamalloo, Carlos Lassance, Rodrigo Nogueira, Odunayo Ogundepo, Mehdi Rezagholizadeh, Nandan Thakur, Jheng-Hong Yang, and Xinyu Zhang. Simple yet effective neural ranking and reranking baselines for cross-lingual information retrieval. *arXiv:2304.01019*, 2023.

[71] Jimmy Lin, Daniel Campos, Nick Craswell, Bhaskar Mitra, and Emine Yilmaz. Fostering coopetition while plugging leaks: The design and implementation of the MS MARCO leaderboards. In *Proceedings of the 45th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022)*, pages 2939–2948, Madrid, Spain, 2022.

[72] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), pages 2356–2362, 2021.

[73] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: BERT and beyond. *arXiv:2010.06467*, 2020.

[74] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Morgan & Claypool Publishers, 2021.

[75] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 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, August 2021.

[76] Johann-Mattis List, Michael Cysouw, and Robert Forkel. Concepticon: A resource for the linking of concept lists. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2393–2400, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[77] Robert Litschko, Ekaterina Artemova, and Barbara Plank. Boosting zero-shot cross-lingual retrieval by training on artificially code-switched data. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3096–3108, Toronto, Canada, July 2023. Association for Computational Linguistics.

[78] Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. Evaluating multilingual text encoders for unsupervised cross-lingual retrieval. In *Proceedings of the 43rd European Conference on Information Retrieval*, pages 342–358, 2021.

[79] Andrew Liu, Edward Xu, Crystina Zhang, and Jimmy Lin. The impact of incidental multilingual text on cross-lingual transfer in monolingual retrieval. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part III*, page 165–173, Berlin, Heidelberg, 2025. Springer-Verlag.

[80] Yihong Liu, Haotian Ye, Leonie Weissweiler, Renhao Pei, and Hinrich Schuetze. Crosslingual transfer learning for low-resource languages based on multilingual colexification graphs. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8376–8401, Singapore, December 2023. Association for Computational Linguistics.

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

[82] Shayne Longpre, Yi Lu, and Joachim Daiber. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406, 2021.

[83] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. B-PROP: Bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1513–1522, 2021.

[84] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. Prop: Pre-training with representative words prediction for ad-hoc retrieval. pages 283–291, 2021.

[85] Xueguang Ma, Kai Sun, Ronak Pradeep, and Jimmy Lin. A replication study of dense passage retriever. *arXiv:2104.05740*, 2021.

[86] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2421–2425, New York, NY, USA, 2024. Association for Computing Machinery.

[87] Sean MacAvaney, Luca Soldaini, and Nazli Goharian. Teaching a new dog old tricks: Resurrecting multilingual retrieval using zero-shot learning. In *Proceedings of the 42nd European Conference on Information Retrieval, Part II*, pages 246–254, 2020.

[88] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July 2020. Association for Computational Linguistics.

[89] Microsoft. Harrier-oss-v1: Multilingual text embedding models, 2026. Model card.

[90] Gregory Murphy. *The big book of concepts*. MIT press, 2004.

[91] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 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, December 2022.

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

[93] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv:1901.04085*, 2019.

[94] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert. *arXiv:1910.14424*, 2019.

[95] Damaris Nübling. Inflectional morphology. In *The Cambridge Handbook of Germanic Linguistics*. Cambridge University Press, 2020.

[96] Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[97] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

[98] Qiwei Peng and Anders Søgaard. Concept space alignment in multilingual LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5511–5526, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[99] Perplexity AI. Diffusion-pretrained dense and contextual embeddings. *arXiv preprint arXiv:2602.11151*, 2026.

[100] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, 2020.

[101] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online, November 2020. Association for Computational Linguistics.

[102] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 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, November 2021. Association for Computational Linguistics.

[103] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics.

[104] Frans Plank. Split morphology: How agglutination and flexion mix. *Linguistic Typology*, 3:279–340, 1999.

[105] Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, 45(3):559–601, September 2019.

[106] Ella Rabinovich and Shuly Wintner. Unsupervised identification of translationese. *Transactions of the Association for Computational Linguistics*, 3:419–432, 2015.

[107] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[108] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[109] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016*

Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[110] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[111] Subendhu Rongali, Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. Continual domain-tuning for pretrained language models. *arXiv:2004.02288*, 2020.

[112] Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. Is multilingual BERT fluent in language generation? In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 29–36, Turku, Finland, September 2019. Linköping University Electronic Press.

[113] Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. Analyzing encoded concepts in transformer language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3082–3101, Seattle, United States, July 2022. Association for Computational Linguistics.

[114] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 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, July 2022. Association for Computational Linguistics.

[115] Henrique Schechter, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, et al. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*, 2025.

[116] Chen Shani, Jilles Vreeken, and Dafna Shahaf. Towards concept-aware large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13158–13170, Singapore, December 2023. Association for Computational Linguistics.

[117] Jiandong Shao, Raphael Tang, Crystina Zhang, Karin Sevegnani, Pontus Stenetorp, Jianfei Yang, and Yao Lu. The role of mixed-language documents for multilingual

large language model pretraining. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36807–36818, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[118] Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. Bilingual lexicon induction via unsupervised bitext construction and word alignment. 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 813–826, Online, August 2021. Association for Computational Linguistics.

[119] Peng Shi, He Bai, and Jimmy Lin. Cross-lingual training of neural models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2768–2773, 2020.

[120] Rion Snow, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. Learning to merge word senses. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1005–1014, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[121] Shuo Sun and Kevin Duh. CLIRMatrix: A massively large collection of bilingual and multilingual datasets for cross-lingual information retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4160–4170, Online, November 2020. Association for Computational Linguistics.

[122] Morris Swadesh. Lexico-statistical dating of prehistoric ethnic contacts. *Proceedings of the American Philosophical Society*, 96:452–463, 1952.

[123] Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies, 2024.

[124] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Neural Information Processing Systems: Datasets and Benchmarks Track*, 2021.

[125] Bill Thompson, Seán G Roberts, and Gary Lupyan. Cultural influences on word meanings revealed through large-scale semantic alignment. *Nature Human Behaviour*, 4(10):1029–1038, 2020.

[126] Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. Multilingual is not enough: Bert for finnish. *arXiv:1912.07076*, 2019.

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

[128] Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 315–323, New York, NY, USA, 1998. Association for Computing Machinery.

[129] Voyage AI. Voyage ai text embeddings, 2026. Documentation.

[130] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024.

[131] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

[132] Xinyi Wang, Sebastian Ruder, and Graham Neubig. 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, May 2022. Association for Computational Linguistics.

[133] Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime G. Carbonell. Cross-lingual alignment vs joint training: A comparative study and a simple unified framework. In *ICLR*. OpenReview.net, 2020.

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

[135] Adina Williams, Nikita Nangia, and Samuel Bowman. 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, June 2018. Association for Computational Linguistics.

[136] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[137] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv:1910.03771*, 2019.

[138] Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China, November 2019. Association for Computational Linguistics.

[139] Yuqing Xie, Wei Yang, Luchen Tan, Kun Xiong, Nicholas Jing Yuan, Baoxing Huai, Ming Li, and Jimmy Lin. Distant supervision for multi-stage fine-tuning in retrieval-based question answering. In *Proceedings of The Web Conference 2020*, pages 2934–2940, 2020.

[140] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, 2021.

[141] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with

pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022.

[142] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 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, June 2021.

[143] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality*, 10(4):Article 16, 2018.

[144] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online, June 2021. Association for Computational Linguistics.

[145] Jingtao Zhan, Qingyao Ai, Yiqun Liu, Jiaxin Mao, Xiaohui Xie, Min Zhang, and Shaoping Ma. Disentangled modeling of domain and relevance for adaptable dense retrieval. *arXiv:2208.05753*, 2022.

[146] Crystina Zhang, Jing Lu, Vinh Q. Tran, Tal Schuster, Donald Metzler, and Jimmy Lin. Tomato, tomahto, tomate: Do multilingual language models understand based on subword-level semantic concepts? In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1821–1837, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[147] Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. Jasper and stella: Distillation of sota embedding models, 2025.

[148] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *arXiv preprint arXiv:2407.19669*, 2024.

[149] Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. Mr. TyDi: A multi-lingual benchmark for dense retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137, Punta Cana, Dominican Republic, November 2021.

[150] Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, and Jimmy Lin. Toward best practices for training multilingual dense retrieval models. *ACM Trans. Inf. Syst.*, 42(2), sep 2023.

[151] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. MIRACL: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131, 2023.

[152] Xinyu Zhang, Andrew Yates, and Jimmy Lin. Comparing score aggregation approaches for document retrieval with pretrained transformers. In *Proceedings of the 43rd European Conference on Information Retrieval, Part II*, pages 150–163, 2021.

[153] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.