Benchmarks, Data, and Evaluation for Robust Retrieval and Retrieval-Augmented Generation on Heterogeneous Domains and Languages

by

Nandan Thakur

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

© Nandan Thakur 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 Examiner: **Soumen Chakrabarti**
Professor
Department of Computer Science
Indian Institute of Technology (IIT) Bombay

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

Internal Member: **Charles L.A. 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: **Mark D. Smucker**
Professor
Department of Management Science and Engineering
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

The thesis includes contributions from nine published papers. Nandan Thakur is the first author of the papers described in Chapters 3, 5, 6, 7, 9, 10, and 11, and is responsible for the ideation, methodology, implementation, conducting experiments, analyzing results, and paper writing. Nandan Thakur is the co-first author (* denotes equal contribution) of the papers discussed in Chapters 4 and 8, and is responsible for the ideation, methodology, and paper writing. List of publications and co-authors:

- Chapter 3: Systematic Evaluation of Neural Retrieval Models on the Touché 2020 Argument Retrieval Subset of BEIR. Nandan Thakur, Luiz Bonifacio, Maik Fröbe, Alexander Bondarenko, Ehsan Kamalloo, Martin Potthast, Matthias Hagen, Jimmy Lin. *In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2024.

- Chapter 4: 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 (TACL),* 2023.

- Chapter 5: FreshStack: Building Realistic Benchmarks for Evaluating Retrieval on Technical Documents. Nandan Thakur, Jimmy Lin, Sam Havens, Michael Carbin, Omar Khattab, Andrew Drozdov. *In Proceedings of the Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025.

- Chapter 6: Leveraging LLMs for Synthesizing Training Data Across Many Languages in Multilingual Dense Retrieval. Nandan Thakur, Jianmo Ni, Gustavo Hernandez Abrego, John Wieting, Jimmy Lin, Daniel Cer. *In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT),* 2024.

- Chapter 7: Hard Negatives, Hard Lessons: Revisiting Training Data Quality for Robust Information Retrieval with LLMs. Nandan Thakur*, Crystina Zhang*, Xueguang Ma, Jimmy Lin. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) Findings*, 2025.

- Chapter 8: Ragnarök: A Reusable RAG Framework and Baselines for TREC 2024 Retrieval-Augmented Generation Track. Ronak Pradeep*, Nandan Thakur*, Sahel Sharifymoghaddam, Eric Zhang, Ryan Nguyen, Daniel Campos, Nick Craswell, Jimmy

Lin. *In Proceedings of the 47th European Conference on Information Retrieval (ECIR) Findings,* 2025.

- Chapter 9: Assessing Support for the TREC 2024 RAG Track: A Large-Scale Comparative Study of LLM and Human Evaluations. Nandan Thakur, Ronak Pradeep, Shivani Upadhyay, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, Jimmy Lin. *In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR),* 2025.

- Chapter 10: "Knowing When You Don't Know": A Multilingual Relevance Assessment Dataset for Robust Retrieval-Augmented Generation. Nandan Thakur, Luiz Bonifacio, Xinyu Zhang, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, Jimmy Lin. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) Findings*, 2024.

- Chapter 11: MIRAGE-BENCH: Automatic Multilingual Benchmark Arena for Retrieval-Augmented Generation Systems. Nandan Thakur, Suleman Kazi, Ge Luo, Jimmy Lin, Amin Ahmad. *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT),* 2025.

Abstract

Modern-day applications increasingly rely on retrieval models and large language models (LLMs) to solve natural language processing (NLP) tasks that are knowledge-intensive, such as information retrieval (IR), question-answering, or fact-checking. However, recent progress has been shaped by assumptions that are increasingly misaligned with real-world deployment, that scaling training data and model size reliably improves out-of-domain robustness, and that leaderboard performance on static, English-centric benchmarks is a meaningful proxy for generalization. LLMs face limitations, especially in such tasks, producing "hallucinations" when handling queries beyond their parametric knowledge. Retrieval-augmented generation (RAG) enhances LLMs by retrieving relevant documents from an external knowledge base via a retrieval system. Similarly, with the onset of pretrained transformers, retrieval systems have demonstrated strong downstream accuracy in the in-domain setting. However, their robustness in the zero-shot scenario remains limited, suggesting that these assumptions do not consistently hold in practice and that current benchmarks and training datasets fail to capture real-world distribution shifts.

Despite recent advances, the progress of retrieval and RAG systems remains hindered by three fundamental challenges. First, existing retrieval benchmarks are largely static, English-focused, and evaluate on homogeneous domains, limiting their ability to measure generalization under domain shift, multilinguality, and evolving corpora. Second, the training data used to fine-tune retrievers is often noisy, imbalanced, sparsely annotated, and scarcely available in non-English languages, so that increasing scale through mixtures of supervised and synthetic data does not reliably translate into improved robustness without principled curation. Third, evaluating RAG systems requires judging not only retrieval relevance but also factuality, grounding, and completeness of long-form answers, for which traditional retrieval metrics are insufficient and large-scale human evaluation is impractical. Together, these limitations constrain reliable progress and obscure real-world performance, motivating the need for realistic benchmarks, high-quality training data, and scalable, trustworthy evaluation methodologies.

Given the breadth of the challenges outlined, this thesis will focus on all three aspects critical to studying and improving robustness in retrieval and RAG systems. The first part focuses on retrieval benchmarks. It revisits argument retrieval within the BEIR benchmark, analyzing why neural retrievers underperform BM25, and improves benchmark robustness through corpus denoising and post-hoc relevance assessment. Subsequently, it introduces MIRACL, a large-scale multilingual ad hoc retrieval benchmark spanning 18 languages with 78k queries and 726k human judgments, improving systematic evaluation of retrieval systems through better coverage, across high- and low-resource languages. To move beyond

static and fixed corpora, the thesis further proposes FreshStack, an automatic framework for constructing realistic benchmarks on technical and niche recent domains to avoid data contamination, by combining automatic corpus collection, automatic nugget generation, and nugget-level support assessment using LLM as judges and hybrid retrieval systems.

The second part examines the role of training data quality for fine-tuning retrieval systems. To alleviate the data scarcity across non-English languages, the thesis presents SWIM-IR, by generating a large-scale synthetic multilingual training dataset covering 33 languages and 28 million query–document pairs, and shows that multilingual dense retrievers fine-tuned on synthetic data only can match or exceed the downstream accuracy of supervised models. It further studies the quality of supervised retrieval data, demonstrating that dataset pruning and false-negative relabeling using cascading LLM judges in RLHN (ReLabeling Hard Negatives) yield consistent improvements in out-of-domain robustness across heterogeneous domains for dense retrievers and rerankers.

The final part of the thesis addresses the evaluation of retrieval-augmented generation. It introduces Ragnarök, a standardized benchmarking framework, implemented through the TREC 2024 RAG track with the MS MARCO V2.1 collection, topics, a unified input–output format, and strong retrieval and reranking baselines. It further investigates support assessment in RAG evaluation at scale, showing substantial agreement between human judgments and LLM-based judges, thereby validating their use as practical evaluators in the TREC RAG track. Further, the thesis presents NoMIRACL, a human-annotated dataset for relevance assessment in multilingual RAG, revealing persistent trade-offs between abstention and correctness in multilingual LLMs. Finally, it introduces MIRAGE-BENCH, a synthetic arena-style multilingual RAG benchmark that trains a surrogate judge from heuristic signals and LLM preferences, enabling scalable and cost-effective evaluation with high agreement with LLM-based rankings.

Overall, this thesis advances the methodological foundations of robust retrieval and retrieval-augmented generation by demonstrating that reliable progress is not limited to model architectures and scaling, but also depends on realistic benchmarks, high-quality training data, and principled, scalable evaluation frameworks. By grounding evaluation in heterogeneous domains and languages, the thesis provides a more reliable basis for measuring progress and deploying robust retrieval and RAG systems in real-world settings.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Jimmy Lin, for supporting me throughout my Ph.D. journey. Working with him gave me considerable freedom to pursue my research independently. I am extremely thankful to him for allowing me to work remotely from Edmonton during a crucial period of my studies. He taught me how to prioritize my research and proofread papers (especially checking each reference carefully). Lastly, I am thankful for his multitasking, for always being active on Slack, and for providing assistance even at the last minute before paper deadlines. I am fortunate to have had such a great mentor and role model during my Ph.D.

Next, I would like to thank my committee members—Charlie, Wenhu, Mark, and Soumen—for taking the time to read my thesis and for providing constructive feedback. In addition, I am grateful for the support of the David R. Cheriton Scholarship from 2024 to 2026, which allowed me to focus on my research without worrying too much about my finances during my Ph.D.

I was fortunate to have extremely nice and wonderful mentors during my internships: Dan and Jianmo at Google; Amin and Suleman at Vectara; and Omar, Andrew, Sam, and Michael at Databricks. I learned a great deal—from how to execute fast-paced research projects to how to communicate my findings effectively in weekly meetings and collaborate with peers. I am also grateful for all the free food, accommodation, and for the opportunity to experience big-tech research at top industry labs in the US.

I would like to thank my lab mates—Cystina, Xueguang, Luiz, and Ehsan—for being my co-authors and, more importantly, good friends, and for all the fun hours spent outside academics, e.g., playing cards at SIGIR. I would also like to thank my TREC RAG coordinators, Shivani and Ronak, for all their help and support in leading the track for two consecutive years. Similarly, I want to thank Ralph, Minghan, Kexin, Jack, Matt, Sahel, Manveer, and Odunayo for the discussions and collaborations, and wish good luck to the newer students: Nour, Steven, Lingwei, Hosna, and Pouya. In addition, I had the pleasure of collaborating with some truly exceptional researchers: Mehdi and David during MIRACL; Sascha and Maik during Touché 2020; and Nick, Daniel, Ian, and Hoa during TREC RAG. I am also grateful to the amazing URA's I got to mentor and learn from them: Nathan, Suraj, Jonathan and Linsen.

Reflecting on my path before the Ph.D., I am grateful to Johannes and Iryna for taking a chance on me and for providing the support that allowed me to start my research career in Germany. I owe a lot of gratitude to where I am today solely to Nils, my mentor at the UKP Lab, who showed me that research can be fun and how to think like an effective researcher—how simple ideas are often the most effective and widely adopted in practice.

Moving on to my friends, I am grateful to Vipin and Saikiran, my roommates and closest friends in Canada, for all the memorable moments, including 4PM SLC coffee sessions, our Banff trip, and cooking dinner together at Ezra. I am also thankful to Ala, Pranali, Kutu, Shlok, Devashish, Sayan, Baltej, Chanpreet and Sonell for all the fun times and conversations over the years during my time in Waterloo and Edmonton. Lastly, I am grateful to all my undergraduate and school friends who have supported me throughout my Ph.D., with whom I've had long calls, especially during the pandemic—Niranjan, Shreya, Anish, Manu, Mehta, Pandya, Urvashi, Ginodia, Debu, Varun, Abhik, Rohit, Akshit, Pramit, Barun, and many others.

I am grateful to my brother and sister-in-law, Spandan and Sonal, for letting me live rent-free at their house during my first internship at Google. I have very fond memories of Kaju always begging me for cat treats, and Kulfi getting excited playing with straws and other cat toys. I also thank Mama and Mami (Bengali for uncle and aunt) for helping me settle in Canada and for not making me feel homesick by including me in all family events, and my cousins—Sonu, Parinaz, and Khushi—for all the memorable conversations and parties in Canada.

Completing this journey would have been a distant reality without the unwavering support of my parents, Ma (Bengali for mom) and Baba (Bengali for dad). They supported my decision to leave my well-settled job in Bangalore to pursue part-time research in Germany and to now build a career in it. They have always encouraged me to pursue my passions and stood by me through countless challenges. I owe it to you both for your support and faith in my own decisions, which have shaped me into the person I am today. I would also like to thank my parents-in-law, Ma and Baba, for gracefully accepting me into their family wholeheartedly without any reservations.

Lastly, and most importantly, I truly believe that choosing Waterloo was destined in my life, as it led me to meet my girlfriend, fiancée, and now my beloved wife, Esha Saha. From the first time I met you as a graduate student serving coffee at a Math CnD coffee event, to tying the knot together in Delhi, Esha, you have been my unwavering support and warmest companion throughout this journey. From cooking together to binge-watching new series to embarking so many trips in the North America, I would have been very lonely without your companionship. You have always believed in me, converted me from a somewhat arrogant into a humble researcher and person. Even during my darkest days, I found warmth in your smile and our conversations. I know you will always & forever support and love me unconditionally. I can proudly say that finding my soulmate and marrying you is the greatest achievement during my time at Waterloo.

Dedication

This thesis is dedicated to my wife, parents, family and friends.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract vi

Acknowledgements viii

Dedication x

List of Figures xviii

List of Tables xxiii

1 Introduction 1

1.1 Challenges in Retrieval and RAG Systems 2

1.2 Benchmarks, Data and Evaluation for Robust Retrieval and Retrieval-Augmented Generation 3

1.3 Contributions 4

2 Background and Related Work **6**
2.1 Preliminaries: Information Retrieval Fundamentals . . . 6
2.1.1 Task Definitions: Ad hoc and Multilingual Retrieval . . . 6
2.1.2 Retrieval Evaluation Metrics . . . 7
2.2 Retrieval & Reranker Models . . . 8
2.2.1 Okapi BM25: Lexical Retrieval . . . 9
2.2.2 Dense & Multi-Vector Retrieval . . . 9
2.2.3 Sparse Retrieval . . . 11
2.2.4 Cross-Encoder or Reranker . . . 11
2.3 Retrieval Benchmarks . . . 12
2.3.1 BEIR Benchmark and Zero-Shot Retrieval Evaluation . . . 13
2.3.2 Domain-specific Retrieval Benchmarks . . . 13
2.3.3 Multilingual Retrieval Benchmarks . . . 15
2.3.4 Retrieval Benchmarks Beyond the Classical Retrieval Task . . . 16
2.4 Retrieval Training Data . . . 17
2.4.1 Pretraining on Weakly Supervised Datasets . . . 17
2.4.2 Synthetic Training Data Generation . . . 17
2.4.3 Supervised Training Data & Artifacts . . . 18
2.5 Retrieval-Augmented Generation Background . . . 19
2.5.1 Task Definition: Retrieval-Augmented Generation . . . 19
2.6 RAG Evaluation Benchmarks and Collections . . . 20
2.6.1 RAG Collections: Corpora & Topics . . . 20
2.6.2 Related Benchmarks & Toolkits . . . 21
2.6.3 Multilingual RAG Benchmarks . . . 21
2.7 RAG Evaluation Methodologies . . . 22
2.7.1 Why Retrieval Metrics Are Insufficient . . . 22
2.7.2 LLM-as-a-Judge and Relevance Evaluation . . . 23
2.7.3 Support or Groundedness Evaluation . . . 23
2.7.4 Nugget or Fact-Based Evaluation . . . 24

3 Revisiting Argument Retrieval Benchmarks **25**
3.1 Touché 2020 Experimental Setup . . . 26
3.2 Evaluation Experiments . . . 28
3.2.1 Black-Box Model Evaluation on Touché 2020 . . . 28
3.2.2 Filtering Short Documents in the Touché 2020 Corpus . . . 30
3.2.3 Adding Post-hoc Relevance Judgments . . . 32
3.2.4 Axiomatic Error Analysis on Touché 2020 . . . 34
3.3 Summary . . . 36

4 Benchmarking Multilingual Retrieval Across a Continuum of Languages **37**
4.1 Dataset Overview . . . 38
4.2 Dataset Construction . . . 40
4.2.1 Corpora Preparation . . . 41
4.2.2 Annotation Workflow . . . 41
4.3 Quality Control . . . 43
4.3.1 Automatic Verification . . . 43
4.3.2 Manual Assessment . . . 44
4.4 MIRACL Analysis . . . 46
4.4.1 Question Words . . . 46
4.4.2 Queries with No Relevant Passages . . . 46
4.5 Experimental Results . . . 48
4.5.1 Baselines . . . 48
4.5.2 Cross-Lingual Transfer Effects . . . 49
4.6 Summary . . . 50

5 Building Realistic Benchmarks on Technical Documents **51**
5.1 The FreshStack Framework . . . 52
5.1.1 Stack Overflow Domain and Question Selection . . . 53

5.1.2 Automatic Corpus Collection with GitHub Documents 53
5.1.3 Nugget Generation with Stack Overflow Question & Answer 54
5.1.4 Retrieval: Oracle & Inference Setting 55
5.1.5 Nugget-Level Support Assessment with Retrieved Documents . . . 56
5.2 Dataset Statistics & Evaluation 57
5.2.1 Dataset Statistics 57
5.2.2 Cost Comparsion: FreshStack vs. Human Judgments 58
5.2.3 Retrieval Evaluation Metrics 58
5.3 Document Pooling & Judgment 58
5.3.1 Document Judgment Pooling Results 59
5.4 Qualitative Analysis 60
5.4.1 Nugget Quality Evaluation 60
5.4.2 Relevance Judgment Quality Evaluation 61
5.5 Main Experiments 62
5.5.1 Document Retrieval Results 63
5.5.2 FreshStack Discussion 64
5.6 Summary 64

6 Scaling Multilingual Retrieval with Synthetic Training Data 65
6.1 Overview 65
6.2 SWIM-IR Dataset Overview 66
6.2.1 Summarize-then-Ask Prompting (SAP) Design Formulation 66
6.2.2 SWIM-IR Dataset Construction 68
6.2.3 Human Validation & Content Filtration 68
6.3 Experiments 71
6.3.1 Datasets and Metrics 71
6.3.2 Experimental Methods 71
6.3.3 Training Methodology 72

6.3.4 Experimental Results . . . 72
6.3.5 Effectiveness of Summarization & K-shot Exemplars . . . 74
6.3.6 How much Synthetic Data to Generate? . . . 75
6.3.7 Indo-European Language Transferability . . . 75
6.4 Cost Comparison . . . 75
6.5 Summary . . . 76

7 Revisiting Supervised Training Data Quality for Robust Retrieval 77
7.1 Overview . . . 77
7.2 LLM Judge Cascading Framework . . . 79
7.2.1 Training Dataset Modification . . . 80
7.2.2 Dataset Postprocessing & Statistics . . . 80
7.3 Experimental Setting . . . 81
7.4 Experimental Results . . . 84
7.4.1 Preliminary Results: Dataset Pruning . . . 84
7.4.2 Main Results: Relabeling False Negatives . . . 84
7.5 Human Validation . . . 87
7.6 Qualitative Analysis of False Negatives . . . 88
7.7 Summary . . . 89

8 A Framework for Benchmarking Retrieval-Augmented Generation 90
8.1 Overview . . . 90
8.2 Ragnarök Framework Description . . . 91
8.2.1 RAG Input/Output Definitions . . . 92
8.3 Document Collection . . . 93
8.4 Topic Collection . . . 94
8.5 TREC RAG Baselines . . . 95
8.5.1 Retrieval . . . 95

8.5.2 Augmented Generation . . . 98
8.5.3 RAG-Bench Evaluation . . . 99
8.6 Ragnarök System Arena . . . 99
8.7 Summary . . . 101

9 Support Assessment in Retrieval-Augmented Generation 102
9.1 The TREC 2024 RAG Track . . . 103
9.2 Support Assessment . . . 104
9.3 Computational Cost & Evaluation Tradeoffs . . . 105
9.4 Support Evaluation Metrics . . . 106
9.5 Experimental Results . . . 108
9.5.1 Weighted Precision and Recall . . . 108
9.5.2 Confusion Matrices . . . 109
9.6 Annotator Disagreements . . . 110
9.7 Qualitative Analysis . . . 111
9.8 Summary . . . 112

10 Relevance Assessment in Multilingual Retrieval-Augmented Generation 113
10.1 Relevance Assessment Definition . . . 114
10.2 NoMIRACL Dataset . . . 115
10.2.1 Data Construction Procedure . . . 116
10.2.2 Languages Covered and Dataset Usage . . . 116
10.3 Experimental Setup . . . 117
10.3.1 Evaluation Setup and Metrics . . . 117
10.3.2 Evaluation Models . . . 117
10.3.3 Experimental Settings . . . 118
10.4 Experimental Results . . . 119
10.4.1 NoMIRACL Non-relevant Subset . . . 119

10.4.2 NoMIRACL Relevant Subset . . . 120
10.4.3 NoMIRACL Overall Comparison . . . 120
10.5 Empirical Analysis . . . 121
10.6 Further Studies . . . 122
10.7 Summary . . . 124

11 Automatic Benchmarking of Multilingual Retrieval-Augmented Generation 125
11.1 An Overview of MIRAGE-BENCH . . . 126
11.1.1 Distinction and Extension from MIRACL . . . 126
11.1.2 MIRAGE-BENCH Evaluation Dataset . . . 127
11.1.3 MIRAGE-BENCH Training Dataset . . . 128
11.2 Multilingual RAG Evaluation . . . 128
11.2.1 Heuristic-based Evaluation . . . 128
11.2.2 Arena-based Evaluation . . . 130
11.2.3 Learning to Approximate Rankings . . . 130
11.3 Experimental Settings . . . 132
11.3.1 Multilingual Baselines . . . 132
11.4 Experimental Results . . . 133
11.4.1 Heuristic-based Results . . . 133
11.4.2 Arena-based Results . . . 135
11.4.3 Ablations & Discussion . . . 136
11.5 Summary . . . 138

12 Conclusion and Future Work 139

References 142

List of Figures

1.1 An illustration of a modern retrieval & RAG system, including a first-stage retrieval (K_1 usually 1,000), reranking (K_2 usually 10–100), and an optional process (dashed arrows) of using the LLM for generating answers in RAG. A lexical retriever relies on exact term overlap, whereas a sparse retriever learns contextualized term weights and expansions within a sparse vocabulary space (definitions and examples provided in Section 2.2.3). 2

2.1 An overview of the diverse tasks and datasets in the BEIR benchmark. . . 12

2.2 Comparison of neural retrieval and reranker models with BM25 on the BEIR benchmark; **green** denotes the number of datasets (out of 18) where the neural retrieval or reranker model (labeled on the x-axis) outperforms BM25; **red** is vice-versa. 14

3.1 Boxplots showing the average length in words (x-axis) of the top-10 Touché 2020 results retrieved by the models on the y-axis (sorted by decreasing nDCG@10; oracle: avg. length of all documents judged as relevant). The results retrieved by neural models are much shorter in comparison to BM25. 27

3.2 **Left:** MS MARCO (blue, top) vs. Touché 2020 (green, bottom) document length distributions. The x-axis shows document length in words; the y-axis is log-scaled frequency. **Right:** Heuristic filtering experiment to determine the best threshold n for filtering short documents in Touché 2020. 31

4.1 Diagram illustrating the MIRACL annotation workflow. 40

4.2 Examples of disagreements between annotators and reviewers. 45

4.3 Case study of how language kinship affects cross-lingual transfer. Results are grouped according to the target language and each bar indicates a source language (used for fine-tuning mDPR). Within each panel, the relationship between the source and target languages moves from closer to more distant from left to right (self, same language sub-family, same language family, different language family). Striped bars denote that the source language has a different script from the target language. The exact nDCG@10 score is shown at the top of each bar. 49

5.1 The FreshStack framework: (1) Stack Overflow questions and answers are sourced for recent and niche domains. (2) GitHub repository documents are collected and chunked to form the corpus (for each domain). (3) Nuggets or atomic facts within the question and answer are generated with GPT-4o. (4) Ensemble techniques and models retrieve documents, which construct our document judgment pools. (5) GPT-4o evaluates support for every document-nugget pair as a binary judgment. 52

5.2 A data instance example from LangChain generated with FreshStack. . . . 54

6.1 An illustration of SAP (*Summarize-then-Ask Prompting*) versus standard prompting for English query generation on English Wikipedia. SAP assists the LLM in improving the query generation quality (orange box) by identifying the relevant sections of the input passage (highlighted in red) via the extractive summarization (yellow box) as an intermediate reasoning step. . 67

6.2 An illustration of the cross-lingual SWIM-IR dataset construction procedure. 68

6.3 SAP versus Standard prompting and K-shot ablations. 74

6.4 Language transfer patterns of SWIM-X (left) and the impact of annotation cost on retrieval performance (right). 76

7.1 Example of a training instance (query, ground truth positives, and unlabeled hard negatives) with detected false negatives taken from HotpotQA. The false negative passage (*Splash Works*) is **mislabeled** as it is relevant in answering the user's query. The relevant parts of the text useful in answering the query are highlighted in **blue**. 78

7.2 Flowchart for **RLHN** (**R**e**L**abeling **H**ard **N**egatives): (1) Provide the query, ground-truth or positive passages, and hard negative passages from a training instance as input, (2) Prompt a cost-effective LLM judge (e.g., GPT-4o-mini) and evaluate whether any hard negative is misclassified, (3) If yes, repeat the prompt with an accurate LLM judge (e.g., GPT-4o) (4) Output the relabeled hard negative passages (which are found relevant) and either remove them or relabel them as ground-truth passages in our experiments. 79

7.3 Distribution of training pairs (with at least one false negative) across detected false hard negatives. 58% have one false negative, 19% have two, and so on. 81

7.4 Dataset pruning by leaving one dataset out during fine-tuning $E5_{base}$ on the BGE-training collection. 84

7.5 nDCG@10 scores on BEIR (Avg. 16 and Avg. 7) and AIR-Bench (Avg. 5) by fine-tuning $E5_{base}$ on a subset of the 100K, 250K, 400K, and 680K training pairs using the "RLHN" technique for both stages. 85

8.1 Schematic diagram of the Ragnarök. Given a user topic (left), the process consists of two steps: (1) (R) retrieval (+ rerank), where the topic yields the top-k relevant segments from our document collection (e.g., potty training articles); and (2) (AG) augmented-generation, where the retrieved segments with a suitable prompt template are fed to the large language model (LLM) to generate the post-processed answer response (JSON) containing individual sentence-level citations. 91

8.2 An illustration of the retrieval and reranking pipeline in Ragnarök. 95

8.3 ChatQA prompt template [203] used for RAG generation with in-text citations with LLaMA3.1-70B and GPT-4o in our Ragnarök framework. 98

8.4 WebUI showcasing the Ragnarök System Arena and the user query, "what inspired pink floyd's the wall?", with answers from two pipelines side-by-side comparing GPT-4o (left) and Command R+ (right). 100

9.1 Pairwise prompt template used by the GPT-4o judge for support evaluation. 105

9.2 Correlations between scores from human and GPT-4o judges for the manual from-scratch condition (top) and the manual with post-editing condition (bottom), measuring weighted precision and recall. Scores observed exactly on the diagonal indicate a perfect overlap (Kendall's $\tau = 1$) between the human and GPT-4o judge score. Red markers show run-level scores, yellow triangles show per-topic averages, and blue dots or green boxes show all individual topic/run combinations. Each plot is annotated with rank correlations showing Kendall's τ. 107

9.3 Confusion matrices comparing predictions from human and GPT-4o judges for the manual from-scratch (left) and the manual with post-editing (right) conditions. 109

9.4 Inter-annotator agreement score (Cohen's κ) for our unbiased study on disagreements between GPT-4o and human (NIST) annotators. 110

9.5 Support label prediction by different judges for each support category (FS, PS, NS) in the disagreement analysis on 537 sentence–passage pairs. 110

10.1 LLM robustness evaluation as a binary tree in NoMIRACL. When dealing with queries in the non-relevant subset, the LLM is expected to disregard all noisy passages and refrain from answering (TN). Conversely, for queries in the relevant subset, the LLM should recognize the relevant passage and provide a valid answer (TP). 114

10.3 An overview of the data construction procedure involved in NoMIRACL. . 115

10.2 Confusion matrix for relevance assessment evaluation; (Subset) denotes the ground-truth; (Pred.) denotes the LLM output prediction. 115

10.4 Hallucination rate (in %) = FP/(FP + TN) on the non-relevant subset (F) in NoMIRACL. Lower the hallucination rate is better. Models are sorted in terms of parameter size or recency. 119

10.5 Error rate (in %) = FN/(FN+TP) on the relevant subset (T) in NoMIRACL. Lower the error rate is better. Models are sorted in terms of parameter size or recency. 119

10.6 Average model accuracy on relevant (y-axis) and non-relevant subset (x-axis) in NoMIRACL. Models closer to the top-right corner are better. 121

10.7 The prompt and its ablations used for LLM robustness evaluation for 18 languages in NoMIRACL. The instruction is provided in English, similar to Ahuja et al. [7]. Vanilla zero-shot prompt template is written in black. Three ablations and their modifications are highlighted by color: (i) **role**, (ii) **repeat**, and (iii) **explanation** that replaces the preamble of the vanilla template. 123

11.1 The MIRAGE-BENCH evaluation flowchart consists of three steps: (i) heuristic-based features evaluating the baseline model response across several dimensions; (ii) exhaustive pairwise comparisons with GPT-4o as a judge on a small subset of queries to train our surrogate judge. (iii) After training, we utilize our surrogate judge to output the model ranking on the whole subset of queries, to construct the RAG arena-based leaderboard. 126

11.2 MIRAGE-BENCH arena-based leaderboards: (left heatmap) Bradley-Terry model coefficients with GPT-4o as a pairwise judge for a subset of 100 sampled queries; (right heatmap) Synthetic rankings using heuristic-based features and a random forest model as a surrogate judge on all queries. Each highlighted cell denotes the rank of the LLM (lower the better). LLMs are sorted by lowest to highest average rank. 131

11.3 Lollipop plots denoting the average heuristic-based feature scores achieved by LLM baselines for each language in MIRAGE-BENCH. x-axis denotes the 18 languages; whereas y-axis plots every heuristic feature score. Models in the same LLM family are represented in the same color in a lollipop (as multiple circles). 134

11.4 Boxplot with the feature importance value (averaged across 18 languages in MIRAGE-BENCH) observed by the learning to rank (random forest) model. 135

11.5 Ablation experiments: (a) Sampling experiments and (b) approximate ranking performance after fine-tuning. 137

List of Tables

3.1 Dense (left) and multi-vector / sparse (right) retrieval models on the BEIR Touché 2020 subset; nDCG@10 scores taken from the referenced publications. 26

3.2 Example of the top-ranked document for the query *"Should social security be privatized?"* showing that neural models may retrieve documents with a relevant conclusion / title (within < >) but a non-relevant premise / body. 28

3.3 Error rates as the percentage of a model's top-k results that are non-relevant (judgment of 0 or unjudged) and shorter than 20 words. Lower error rates are better. 29

3.4 The Touché 2020 dataset characteristics before and after filtering and post-hoc judgments. Reported are the total number of documents, the average document length, the number of queries, the number of relevance-judged documents, and the number of documents per relevance grade: non-relevant (0), relevant (1), and highly relevant (2). 32

3.5 Retrieval effectiveness as nDCG@10 and missing judgments as hole@10 on the original, filtered (cf. Section 3.2.2), and post-hoc judged (cf. Section 3.2.3) Touché 2020. 33

3.6 Agreement (in %) with the length normalization axiom LNC2 when retrieving with (w/) or without (w/o) the title on Touché 2020. BM25 agrees perfectly with LNC2. 34

3.7 Agreement (in %) with the term frequency, document length, and semantic similarity axioms for all models on the original (O) and the denoised (+D) data. 35

4.1 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 training data for neural models. 38

4.2 Descriptive statistics for all languages in MIRACL, organized by split. # **Q**: number of queries; # **J**: number of human judged pairs (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. 39

4.3 Query distribution of each split in English MIRACL, compared to SQuAD. 46

4.4 Baseline results on the MIRACL dev set, where "K. Avg." and "S. Avg" indicate 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) 47

5.1 FreshStack dataset statistics; Dataset count measures the number of queries, documents, GitHub repositories, and average nuggets per query; Avg. length measures the average text lengths; % containing code measures the fraction of queries and answers with code snippets; Relevance judgments measure relevant documents per nugget and per query. 56

5.2 Pooling results by retrieval baselines (including fusion) in inference or oracle settings during FreshStack construction. α-N@10 denotes α-nDCG@10, C@20 denotes Coverage@20 and R@50 denotes Recall@50. Stack Overflow Answer & Nuggets both rely on the gold answer for retrieval (oracle setting), whereas other methods do not rely on the gold answer for retrieval (inference setting). The best result is highlighted in **bold**. 60

5.3 Expert human evaluation by the author of this thesis for nugget quality and nugget-document relevance judgments on 60 randomly sampled queries in LangChain. 61

5.4 Document retrieval results on FreshStack with retrieval and reranker baselines (including fusion). Best scores or upper baselines in the oracle setting are taken from Table 5.2. The reranker is the Voyage AI rerank-2 model [9] reranking the top 100 documents. If the reranker improves upon the retrieval model, it is highlighted in **green** else **red**. We highlight the best result in **bold**. 62

6.1 We construct SWIM-IR, a "synthetic" multilingual dataset with 28 million PaLM 2 generated training pairs across 33 languages in our work; (Q Gen.) denotes the query generation technique; (Cross. and Mono.) denotes the retrieval task and (query→document) language pair; (# L and # Train) denotes the language count and available training pairs. 66

6.2 Human validation statistics on SWIM-IR. Annotators rate fluency, adequacy, and language correctness on a 0/1/2 scale. 69

6.3 Experimental results showing Success@5kt for cross-lingual retrieval on XOR-Retrieve dev [23]; (PLM) denotes the pre-trained language model; (PT) denotes the pre-training dataset; (*) Dr.DECR is fine-tuned in a complex training setup across more datasets (§6.3.3); WikiM denotes WikiMatrix [299]; XOR denotes XOR-Retrieve; SWIM-X (ours) is fine-tuned on 500K and 7M synthetic data. 70

6.4 Experimental results for monolingual retrieval on MIRACL dev [398]. All scores denote **nDCG@10**; (Hyb.) denotes Hybrid retriever with ranked fusion of three retrievers: mDPR, mColBERT and BM25; BM25, mDPR and Hybrid scores taken from [398]; Cohere-API is used as a reranker on top of 100 BM25 results, taken from [153]. SWIM-X (ours) is fine-tuned on 180K synthetic training pairs. 71

6.5 Experimental results for cross-lingual retrieval on XTREME-UP test [288]. ($^{\heartsuit}$) denotes the mContriever-X model fine-tuned without MS MARCO [29]; Two variants of SWIM-X considered, both fine-tuned on 120K synthetic data: (1) SWIM-X (120K)^{MT} fine-tuned using Google Translate, i.e., translated prompt exemplars for 15 languages, whereas (2) SWIM-X (120K) is fine-tuned using prompt exemplars sampled from XTREME-UP training split for all languages. 73

7.1 Training dataset statistics (left) and RLHN relabeling cost analysis (right). 80

7.2 Performance of encoder (top) and decoder (bottom) backbones <= 500M parameters, fine-tuned on the full BGE training collection for 3 epochs with above mentioned hyperparameters, and evaluated on four datasets on BEIR (T-COVID is TREC-COVID, NFC. is NFCorpus). The best score is **highlighted** and the second best is underlined. 82

7.3 Retrieval results measuring nDCG@10 on 16 datasets in the BEIR benchmark by fine-tuning retrieval models on 7 datasets from the BGE training collection with relabeled false negatives. The seven unseen (or out-of-domain) datasets during fine-tuning are highlighted with † and their average scores are provided in Avg. 7; Rem. denotes Remove. 83

7.4 Retrieval results measuring nDCG@10 on five specialized domains in AIR-BENCH dev (version `24.05`) after fine-tuning $E5_{base}$ and Qwen2.5-7B. . . . 85

7.5 Reranker results measuring nDCG@10 on 16 datasets in BEIR by fine-tuning reranker models (based on Qwen2.5-3B) on RHLN (Stage 1 and Stage 2) variants. 86

7.6 Cohen's κ agreement of GPT-4o-mini and GPT-4o with judgments on 670 query–negative pairs with three human assessors (the authors from the original publication). 87

7.7 Qualitative analysis showcasing the different varieties of false negatives detected by RLHN. The first two questions are taken from HOTPOTQA, the third from MS MARCO, and the last from FEVER. The text supporting the query is highlighted in **green**, partially supporting in **orange**, and not supporting with **red**. 88

8.1 Comparison of document and segment counts between versions V2 and V2.1 (our version after removing near-duplicates) of the MS MARCO collection. 93

8.2 TREC-RAGgy 2024 and TREC-Researchy 2024 topic distribution; top-5 categories in topic classification for TREC-RAGgy 2024 (each topic classified into a single category), intrinsic attributes for TREC-Researchy 2024, and the first word in all topics. 94

8.3 Results on the Document Ranking Task of the TREC-RAGgy 2024. 97

9.1 A sample RAG answer segmented into multiple sentences with citations to passages retrieved from the MS MARCO V2.1 segment collection for a Taylor Swift query. We highlight the cited passages, provided as a list (that can be empty, i.e., zero citations). 103

9.2 Descriptive statistics for support judgments for the (1) manual from-scratch and (2) manual with post-editing conditions for 45 participant submissions on 36 topics. 108

9.3 Examples of annotation mistakes by either GPT-4o or the human judge for the topic "*how taylor swift's age affects her relationships*" taken from the disagreement analysis. The fragment of the passage that supports the answer sentence is highlighted. 111

10.1 Empirical results on the complete NoMIRACL English (`en`) non-relevant (above) and relevant (below) subsets. The analysis is bracketed into three categories, where **green** category denotes an accurate response, **orange** denotes limitations in understanding the instruction and **red** denotes model hallucination or error respectively. 122

10.2 Hallucination and error rates on the NoMIRACL test split (non-relevant and relevant subsets) with three types of prompting techniques on Mistral-7B (v0.3). 124

11.1 Dataset statistics for 18 languages in MIRAGE-BENCH; All tokens are calculated using the GPT-4o tokenizer [246]; (Rel.) denotes relevancy; (# Avg GPT-4o) counts the tokens in the GPT-4o generated context and answer; Training data is not available for two languages (denoted by —): German (de) and Yoruba (yo). 127

11.2 Kendall τ rank correlation between Bradley-Terry model (using GPT-4o as judge) and the synthetic arena-based leaderboard (using our surrogate judge). 133

11.3 Train and Holdout R^2 scores using different learning to rank model choices. Each experiment has been repeated 50 times with four held-out models. . . 136

11.4 Kendall Tau (τ) scores using different features for training the surrogate judge. 138

Chapter 1

Introduction

Large language models (LLMs) have achieved remarkable success in natural language processing (NLP), though they still face significant limitations, especially in domain-specific, or knowledge-intensive tasks [155] and even languages [338], notably producing "hallucinations" [400] when handling queries beyond their training data or requiring current information. Retrieval-Augmented Generation (RAG) enhances LLMs by retrieving relevant documents from an external knowledge base via a retrieval system. By referencing external knowledge, RAG effectively reduces the problem of generating factually incorrect content.

Traditionally, lexical-based information retrieval systems have been popular such as BM25 [281], which rely on bag-of-words and term-matching approaches. However, the advancement above has also set a new foundation for information retrieval (IR). As language models have become more adept at understanding context and generating text, retrieval systems need to adapt and evolve. In this context, neural retrieval systems that encode dense vector representations of query and documents in a high-dimensional space have achieved state-of-the-art downstream accuracy in various retrieval tasks, such as open-domain question answering [55, 177].

The RAG methodology modifies the traditional retrieval task—which returns ranked lists of documents—by using LLMs to source information from multiple sources as documents to generate a direct answer to a user's query [180]. This approach relies on a retrieval system with quick access to vast and up-to-date knowledge sources, to handle queries on specific domains or low-resource languages, effectively. Figure 1.1 illustrates the complete pipeline. In this thesis, we describe challenges in effectively evaluating retrieval and RAG systems, which contribute toward improving their robustness on heterogeneous domains and languages.

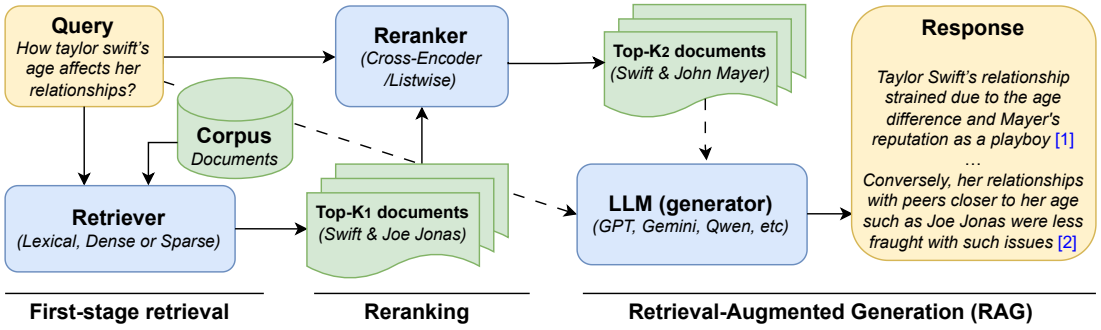


Figure 1.1: An illustration of a modern retrieval & RAG system, including a first-stage retrieval (K_1 usually 1,000), reranking (K_2 usually 10–100), and an optional process (dashed arrows) of using the LLM for generating answers in RAG. A lexical retriever relies on exact term overlap, whereas a sparse retriever learns contextualized term weights and expansions within a sparse vocabulary space (definitions and examples provided in Section 2.2.3).

1.1 Challenges in Retrieval and RAG Systems

Modern information access increasingly relies on retrieval systems and large language models (LLMs), as progress is no longer driven solely by scaling up model parameters or embedding dimensions [233, 236]. Evaluating such systems on heterogeneous benchmarks and improving training data quality are both essential for improving robustness in retrieval systems. Similarly, while the RAG methodology has become an industrial standard, the techniques and benchmarks needed to reliably evaluate them continue to lag behind.

The first challenge concerns **retrieval benchmarks**. At a fundamental level, this is an issue of *availability*: existing benchmarks are constructed over static, homogeneous corpora, with evaluations primarily focusing on English or a general domain. As a result, they fail to capture model generalization on domain-specific corpora [329], and multilingual settings [398]. Moreover, even when benchmarks exist, artifacts such as noisy documents, pooling bias, or even outdated collections, can unfairly penalize and lead to misleading conclusions about system performance.

The second challenge concerns **data quality**. In contrast to benchmarks, large-scale training data is available, but their reliability is questionable. State-of-the-art (SoTA) retrievers are typically fine-tuned on large-scale mixtures of human-labeled (i.e., supervised) and synthetic training datasets [399]. However, supervised training data for non-English languages remains unevenly distributed and scarce [71]. Conversely, widely used English supervised datasets often contain sparse or incomplete relevance judgments [29], leading to data inaccuracies and false negatives that can harm retrieval robustness. Moreover, even

large supervised training collections do not necessarily translate into improved robustness in retrieval models. Consequently, the challenge here is not the absence of data, but the difficulty of ensuring its quality and reliability for robust training of retrieval models.

The third challenge arises in **RAG evaluation**. Unlike retrieval systems, RAG systems produce long-form answers that should be evaluated not only for retrieval accuracy, but also for factual correctness, grounding, attribution, and completeness in the generated answer. Evaluation metrics designed for ranking or question answering are insufficient for this purpose, and large-scale human evaluation is expensive and cumbersome. Automated metrics and LLM-based judges offer promising alternatives, but their reliability and limitations are not yet well understood, particularly in multilingual and domain-specific settings.

Together, these challenges indicate that progress in robust retrieval and RAG systems is constrained by the lack of realistic benchmarks, reliable training data, and scalable evaluation frameworks. Addressing these methodological gaps is essential toward robust, reproducible, and trustworthy retrieval and RAG systems across domains and languages.

1.2 Benchmarks, Data and Evaluation for Robust Retrieval and Retrieval-Augmented Generation

We address the above challenges by constructing benchmarks that enable realistic evaluation, as well as developing methods for data curation and data generation aimed at improving out-of-domain robustness. We also develop a framework for automated LLM-as-a-judge evaluation to help standardize RAG evaluation across heterogeneous domains and languages.

Chapter 2 lays the groundwork by providing a comprehensive background on information retrieval (IR) and Retrieval-Augmented Generation (RAG), from the perspective of retrieval benchmarks, retrieval training data and RAG evaluation approaches.

The subsequent chapters tackle the benchmarking challenge. Chapter 3 begins with an in-depth investigation of why neural retrievers underperform BM25, by revisiting Touché 2020, the argument retrieval subset of BEIR [329]. We further denoise the document corpus and conduct post-hoc judgments to alleviate noise and sampling bias, enabling a robust evaluation of retrieval systems for argument retrieval. Chapter 4 presents MIRACL, a large-scale multilingual retrieval dataset spanning 18 languages and 726k human relevance judgments, to accelerate retrieval progress in non-English languages. MIRACL enables a systematic evaluation of retrieval systems on both high and low-resource languages. Chapter 5 introduces FreshStack, an automatic framework to construct realistic evaluation

benchmarks on technical and niche domains using LLMs as judges, such as LangChain, while avoiding data contamination due to (perpetual) recency.

The next chapters explore the data challenge, by focusing on data generation and curation of fine-tuning datasets. Chapter 6 introduces a synthetic data generation pipeline to generate training data at scale across multiple languages. Dense retrievers fine-tuned on the synthetic dataset can outperform supervised counterparts, generated at a much lower cost. Similarly, on the supervised training dataset side, Chapter 7 explores dataset-level pruning and false negative relabeling to curate datasets for training retrieval models efficiently and improves their downstream accuracy on heterogeneous domains.

Finally, we move from retrieval to retrieval-augmented generation, where we focus on standardization and evaluation. Chapter 8 introduces Ragnarök, an end-to-end RAG framework providing critical data resources and reproducible baselines, creating essential infrastructure for a rigorous RAG assessment and serving as the backbone for the TREC 2024 RAG Track. Building on Ragnarök, Chapter 9 utilizes LLM as judges for automatic support assessment in RAG answers and validates this approach at scale in TREC 2024 RAG Track. On the multilingual side, Chapter 10 introduces NoMIRACL, a dataset for evaluating passage-level relevance in multilingual RAG across diverse languages by formulating relevance assessment as a binary classification task and evaluating LLMs' ability to abstain when non-relevant passages are retrieved. Lastly, Chapter 11 presents MIRAGE-BENCH, a synthetic multilingual arena-based benchmark that couples heuristic evaluations to emulate pairwise LLM-as-a-judge judgments via a surrogate judge.

1.3 Contributions

In this thesis, we contribute to the field of IR and NLP by creating robust benchmarks, curating human-labeled training datasets and generating synthetic training datasets, and accelerating RAG evaluation and standardization. Our contributions are as follows:

- Revisiting argument retrieval benchmarks (Chapter 3): We investigate why neural retrievers underperform BM25 on argument retrieval, denoise the short and non-argumentative documents, and conduct a post-hoc relevance judgment study in the Touché 2020, to improve evaluation on argument retrieval subset of BEIR.

- Benchmarking multilingual retrieval across a continuum of languages (Chapter 4): We present MIRACL, a multilingual ad hoc retrieval dataset spanning 18 languages, with 78k queries and 726k human relevance judgments. The benchmark enables systematic evaluation of retrieval models on both high and low-resource languages.

- Building realistic benchmarks on technical documents (Chapter 5): We introduce FreshStack, an automatic framework to construct realistic benchmarks on niche and technical domains, using (1) automatic corpus collection from technical documentation, (2) nugget generation from community-asked questions and answers, and (3) nugget-level support, retrieving documents using a fusion of hybrid architectures.

- Scaling multilingual retrieval with synthetic training data (Chapter 6): We present SWIM-IR, a large-scale synthetic training dataset spanning 33 languages with 28M training pairs, constructed using LLMs with a summarize-then-ask prompting scheme. Fine-tuning on SWIM-IR yields dense retrievers that are competitive with human-supervised models across multiple multilingual benchmarks.

- Revisiting supervised training data quality for robust retrieval (Chapter 7): We evaluate the quality of supervised retrieval training data and demonstrate that dataset pruning and false negative relabeling using LLMs improves robustness in terms of out-of-domain (OOD) accuracy for both dense retrievers and rerankers.

- A framework for benchmarking retrieval-augmented generation (Chapter 8): We propose Ragnarök, a framework for benchmarking RAG and implement it through the TREC 2024 RAG track. In addition, we release a new collection, topics, standardized I/O, and strong baselines, providing a common evaluation for future RAG systems.

- Support assessment in retrieval-augmented generation (Chapter 9): We conduct a large-scale study of support evaluation, comparing human judgments with an LLM-based judge. Our results show substantial agreement between humans and the LLM, indicating LLM judges are a practical and reliable alternative for support evaluation.

- Relevance assessment in multilingual retrieval-augmented generation (Chapter 10): We introduce NoMIRACL, a human-annotated multilingual dataset for evaluating relevance in RAG across 18 languages. Using hallucination rate and error rate metrics, we show that current LLMs struggle to balance abstention and grounding, revealing substantial robustness gaps in multilingual RAG.

- Automatic benchmarking of multilingual retrieval-augmented generation (Chapter 11): We propose MIRAGE-BENCH, a synthetic arena-based benchmark for multilingual RAG that couples heuristic signals with LLM judgments via a surrogate judge. The approach enables scalable, cost-effective evaluation across 18 languages while preserving high agreement with LLM-based rankings.

Chapter 2

Background and Related Work

In this chapter, we provide the necessary background to understand fundamental concepts and related works in both Information Retrieval (IR) and Retrieval-Augmented Generation (RAG). We first cover IR as a standalone discipline, covering fundamentals (Section 2.1), model architectures (Section 2.2), benchmarks (Section 2.3) and training data (Section 2.4). Next, we turn to RAG, highlighting the background (Section 2.5), benchmarks (Section 2.6) and evaluation methodologies (Section 2.7). These sections motivate our contributions to developing benchmarks, data and evaluation for robust IR and RAG systems.

2.1 Preliminaries: Information Retrieval Fundamentals

2.1.1 Task Definitions: Ad hoc and Multilingual Retrieval

Ad hoc Retrieval & Test Collection. The standard *ad hoc* retrieval task in information retrieval, where given a corpus $\mathcal{C}$ containing a set of documents, the retrieval model's task is to return for a given query q an ordered list of top-k documents from $\mathcal{C}$ that maximizes some standard retrieval quality metric such as nDCG. Following the Cranfield paradigm [407, 347], to measure the *ad hoc* retrieval task in the standard way, we require a test collection (or a retrieval benchmark) consisting of three things: (1) A document collection or corpus C, (2) A set of user queries, and (3) A set of relevance judgments or the ground truth, containing a judged assessment of relevance (multi-level or binary) of query-document pairs. Finally, the list of ranked top-k documents is assessed against the relevance judgments, to calculate the downstream retrieval accuracy.

Multilingual Retrieval. Multilingual retrieval is the ability to process a query in any language, and return text relevant to the user query in any language. "Multilingual" is an umbrella term that broadly covers: *monolingual* and *cross-lingual* retrieval. Given a natural language query q in some language $\mathcal{L}_n$ documents drawn from the same language $\mathcal{C}_n$. *Monolingual* retrieval focuses across diverse languages, where the queries and the corpora are in the same language $\mathcal{L}_n$ (e.g., Thai queries searching Thai documents), as opposed to *cross-lingual* retrieval, where the queries and the corpora are in *different* languages (e.g., searching Swahili documents with Arabic queries).

2.1.2 Retrieval Evaluation Metrics

Depending on the nature and requirements of real-world applications, retrieval tasks can be either recall-focused or precision-focused. For example, the duplicate patent retrieval task [137] is recall-focused, since the goal is to retrieve *all* potential duplicates given a new patent that is being filed. In contrast, traditional web search retrieval tasks [29] are primarily precision-focused, as they evaluate whether the top-ranked webpage shown to the user is relevant.

We first cover decision-support metrics that are rank-unaware—i.e., metrics for which document order does not affect the score—such as Recall (Recall@k). We then discuss rank-aware retrieval metrics, including MAP@K, MRR@k and nDCG@k, as well as diversity-aware metrics such as α-nDCG@k.

Recall@k. Recall at k is calculated as the fraction of the relevant documents that are successfully retrieved within the top k extracted documents. Given Q is the set of queries, A_i^* is the set of relevant documents for the i^{th} query, and A_i is a scored list of documents provided by the model, from which top k are extracted. The metric is defined as:

$$\text{Recall@}k = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{|\max_k(A_i) \cap A_i^*|}{|A_i^*|} \tag{2.1}$$

Success@5kt. Introduced by Asai et al. [23] as Recall@5kt, Sucess@5kt measures the fraction of questions for which the correct, minimal answer is contained within the top 5,000 tokens of the retrieved documents concatenated together as retrieved context. We rename it to Success@5kt, as we find it more intuitive for evaluation.

MAP@k. Mean Average Precision (MAP) at k computes the mean of average precision. Given, N is the total number of relevant documents in top-k for a query q, Precision@k

computes the number of relevant documents up to k and divided by k. Lastly, rel_k equals 1 if the document at position k is relevant and 0 otherwise. The definition is below:

$$\text{AP@}k = \frac{1}{N}\sum_{k=1}^{k} \text{Precision@}k \times rel_k, \qquad \text{MAP@}k = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \text{AP@}k, \tag{2.2}$$

MRR@k. Mean Reciprocal Rate (MRR@k) is a rank-aware metric suitable for binary relevance judgments that measures the reciprocal rank of first relevant item. Given, $rank_i$ is the position of the first relevant item for query i in the top-k results. The definition of the metric can be found below:

$$\text{MRR@}k = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{2.3}$$

nDCG@k. Normalised Cumulative Discount Gain (nDCG@k) [142] is suitable for both tasks involving binary and graded relevance judgments. NDCG@K normalizes DCG@K using the Ideal DCG@K (IDCG@K) rankings. Both definitions can be found below:

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)}, \qquad \text{nDCG@}k = \frac{\text{DCG@}k}{\text{iDCG@}k}, \tag{2.4}$$

where $\text{rel}_i = 1$ if document d_i is relevant to the query and 0 otherwise, and iDCG@k is the DCG@k of the ideal ranked list (i.e., with binary judgments, all relevant documents appear before non-relevant documents).

α-nDCG@k. Introduced in Clarke et al. [72], this variant of nDCG measures search diversification. The α parameter is a geometric penalization for redundant documents, i.e., each redundant document achieves a penalization of $(1-\alpha)$ if present in a topic with the same user intent. We ask the reader to refer to Clarke et al. [72] for more information.

2.2 Retrieval & Reranker Models

In this section, we review the key model architectures used in modern retrieval systems, including lexical, dense (and multi-vector), and sparse retrieval models. We then cover reranker architectures. Some of these models serve as important baselines or backbones in the subsequent chapters.

2.2.1 Okapi BM25: Lexical Retrieval

Okapi BM25 (Best Match 25) [281] is a lexical or bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document within the collection. BM25 is useful in ranking documents of different lengths in the collection and is a strong baseline for zero-shot retrieval evaluation. More formally, given a query q, containing keywords $\{q_1, \ldots, q_n\}$, the BM25 score of a document d is computed as:

$$\text{score}(q,\ d) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, d) \cdot (k_1 + 1)}{f(q_i, d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)} \tag{2.5}$$

where $f(q_i, d)$ is the number of times keyword q_i occurs in document d, $|d|$ is the length of document d in words, and avgdl is the average document length in the collection of documents. k_1 which indicates the weight of the term overlap is set to $k_1 = 1.2$ and b which indicates the document length normalization term is set to $b = 0.75$. Finally, $\text{IDF}(q_i)$ measures the inverse document frequency weight of term q_i, is computed as follows:

$$\text{IDF}(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right) \tag{2.6}$$

where N is the total number of documents in the collection and $n(q_i)$ is the number of documents in the collection containing q_i query term.

2.2.2 Dense & Multi-Vector Retrieval

A dense retriever is a foundational neural retrieval model representing a document or query in a compact format using a single dense embedding (i.e., a fixed array of floats). Given a query q and a document d, a dense retriever can either produce independent representations, $\mathbf{E}_Q(d)$ and $\mathbf{E}_D(d)$, or compute them using the same retriever ($\mathbf{E}_Q = \mathbf{E}_D$). Similarly, for the final representation, the [CLS] token is chosen, or mean pooling across all tokens, or the [EOS] or final token for decoder-only architectures, in the final layer. The inner product measures the semantic similarity between a query and a document:

$$s_{q,d} \triangleq \text{sim}(q, d) = \mathbf{E}_Q(q)^T \mathbf{E}_D(d) \tag{2.7}$$

which is optimized according to the InfoNCE loss during fine-tuning:

$$L(q_i, d_i^+, d_{i,1}^-, \ldots, d_{i,n}^-) = -\log\left(\frac{\exp(s_{q_i,d_i^+})}{\exp(s_{q_i,d_i^+}) + \sum_{j=1}^{n} \exp(s_{q_i,d_{i,j}^-})}\right) \tag{2.8}$$

where q_i is the query representation, d_i^+ is the representation of a positive (relevant) document, and d_j^- is the representation of a negative (non-relevant) document. The similarity score $s_{q,d}$ ranks documents for query q using a nearest neighbor vector search. In our thesis, we explore two broad families of dense retrieval models:

Encoder-only. Sentence-BERT [275] and DPR [156] are among the earliest dense retrieval models, constructed using pre-trained language models such as BERT [86]. TAS-B or its variants [128, 377, 263, 209], a bi-encoder architecture initialized using the DistilBERT backbone [294]. These models fine-tune on the MS MARCO dataset [29], coupled with effective hard negative sampling and batching techniques. Similarly, Contriever [139] and its variants [244], include an additional unsupervised pre-training stage, where they are trained to retrieve random crops of text segments, similar to inverse cloze task (ICT) [177], using the loss function referenced in Equation 2.8. RetroMAE [375] pretrains the retrieval model in a unsupervised training stage based on the masked auto-encoder, and is further trained on augmented data in DRAGON+ [195]. For multilingual experiments, a multilingual dense retriever is initialized using a multilingual backbone such as either mT5 [379] or mBERT [86]. Later, encoder models such as E5_{base} [358] that focus on general-purpose embeddings, by pre-training and fine-tuning on large and highly curated domain-specific pre-training datasets such as C4 [266] or CCMatrix [300]. OpenAI text embeddings [233] examine scaling up the dense retrieval model size to large parameter sizes, such as 175B parameters. Arctic-Embed [222, 388] and NV-Retriever [228] train models using an effective hard negative mining strategy, and incorporate curriculum learning, by gradually increasing negative hardness during training. In contrast, models such as GTE [186] modifies the standard InfoNCE loss function with variants [227]. Similarly, CDE [229] computes embeddings by encoding the neighbor document information into the encoded representation. Finally, there are newer encoder architectures that replace the traditional BERT, by incorporating rotary positional embeddings (RoPE) [318], such as ModernBERT [366] to attend to tokens beyond a sequence length of 512 tokens, used as a backbone in the latest GTE models.

Decoder-only. Using pre-trained decoder-only models on the scale of billion-parameters for dense retrieval has become recently popular. One of the earliest works in SGPT [230] and GTR-XXL [236] used models such as GPT as the backbone of dense retrievers. RepLLaMA [211] and E5 (Mistral) [359], correspondingly utilized high-performing LLMs such as LLaMA [334] and Mistral [145], by using the `[EOS]` token appended at the end of the query and document, to obtain the query and document embeddings. Similarly, other dense retrievers such as the BGE [59] uses Gemma-2 [323], and NV-Embed [173] uses Mistral 7B as the backbone. LLM2Vec [31] enables decoder-only LLMs into dense retrievers, by enabling bi-directional instead of causal attention. Lastly, Qwen3 embeddings [399] and Stella [395] include effective knowledge distillation from larger models as teachers.

Multi-vector. Multi-vector models such as ColBERT [159, 296] and its variants, such as PLAID [295] or XTR [175], compute token-level embedding representations for the query q and document d. $s_{q,d}$, estimated via *late interaction*. For each query token q_i, late interaction computes the sum of maximum similarity computations with all document token d_j, as shown below:

$$s_{q,d} = \sum_{i \in [|E_q|]} \max_{j \in [|E_d|]} E_{q_i}^T \cdot E_{d_j} \tag{2.9}$$

CITADEL [183] uses a dynamic lexical routing that is learned, instead of all-to-all routing in ColBERT, and static exact match routing in COIL [109]. Lastly, related works such as MUVERA [87], propose a retrieval mechanism which reduces multi-vector similarity search to efficient single-vector similarity search.

2.2.3 Sparse Retrieval

Sparse models perform retrieval by term weighting based on BERT-based networks, which go along with document and query expansion [401, 107, 192, 217, 241, 106]. The first category (including DeepImpact and uniCOIL) learns scalar weights for terms present within the document [192, 217]. On the other hand, models such as SPLADEv2 pools token weights on BERT's vocabulary space [107, 106, 169] and thus representing queries and documents with possibly different term tokens. The model first gets BERT contextualized token embeddings $s_j (j = 1, ..., l)$ for all document tokens $x_j (j = 1, ..., l)$. Then it maps these document token embeddings into token-prediction logits $w_{i,j} (i = 1, ..., |V|)$ via a MLM layer:

$$w_{i,j} = \text{MLM}(s_j)[i] \tag{2.10}$$

where $[i]$ represents taking the element at the i-th position from a vector. The final text representation is obtained via ReLU activation, log operation, and max-pooling:

$$\phi_{\text{SPLADEv2}}(x_i) = \max_{j=1,...,l} \log(1 + \text{ReLU}(w_{i,j})) \tag{2.11}$$

2.2.4 Cross-Encoder or Reranker

The basic idea behind rerankers is to decompose the *ad hoc* retrieval down into a series of stages. Following an initial retrieval stage (also called candidate generation or first-stage retrieval), each subsequent stage reranks the list of candidates passed from the previous stage until the final top-k results are produced for consumption.

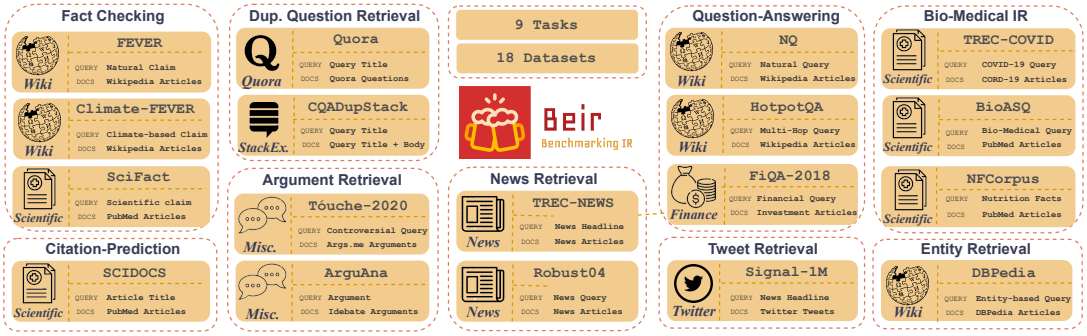


Figure 2.1: An overview of the diverse tasks and datasets in the BEIR benchmark.

We focus on recent neural rerankers based on pretrained language models such as monoT5 [240, 239, 170]. A reranker can also be viewed as a scoring model. However, instead of using a bi-encoder architecture, rerankers typically adopt a cross-encoder architecture, in which question–document pairs are jointly encoded and fed into a single model to enable more fine-grained token-level interactions:

$$s_{rerank}(q, d) = \zeta\big(\text{concat}(q, d)\big) \tag{2.12}$$

where ζ denotes the reranker that takes a question–document pair as input and outputs the similarity score s_{rerank}. One way to implement the `concat` function is to use special tokens as separators, for example, `[CLS]` q `[SEP]` d `[SEP]`.

Listwise Reranker. The listwise paradigm, instead of generating a similarity score, directly reorders an entire list of input documents in terms of relevance. The input processes query–documents list rather than individual query–document pairs. The models primarily used for listwise reranking are LRL [212], RankGPT [320] and RankZephyr [257], where an LLM is prompted to reorder a list of documents in order of relevance to the query.

2.3 Retrieval Benchmarks

Existing retrieval benchmarks are typically constructed for individual information-seeking tasks, such as question-answering [351, 157, 55, 215, 165, 77], claim verification [332, 89, 355], entity retrieval [123, 254], or argument retrieval [354, 35]. The BEIR benchmark [329] was one of the first comprehensive benchmarks to evaluate retrievers across heterogeneous domains and tasks. In this section, we first introduce BEIR, revisit domain-specific and multilingual benchmarks, and then cover benchmarks that extend beyond the retrieval task.

2.3.1 BEIR Benchmark and Zero-Shot Retrieval Evaluation

The BEIR benchmark introduced in Thakur et al. [329] is among the first large-scale efforts to construct a benchmark focused on heterogeneous domains and tasks, including factoid question answering, entity retrieval, and argument retrieval. Retrieval systems are applied in a zero-shot setup, with no available training data to train the system. An overview of the BEIR benchmark is provided in Figure 2.1. The BEIR benchmark was subsequently integrated within MTEB as the retrieval subset [232]. For BEIR, the dataset selection methodology is motivated by the following three factors:

1. **Diverse tasks**: Information retrieval is a versatile task and the lengths of queries and indexed documents can differ between tasks. Sometimes, queries are short, like a keyword, while in other cases, they can be long like a news article. Similarly, indexed documents can sometimes be long, and for other tasks, short like a tweet.
2. **Diverse domains**: Retrieval systems should be evaluated in various types of domains. From broad ones like News or Wikipedia to highly specialized ones such as scientific publications. We include domains that provide a representation of real-world problems and are diverse ranging from generic to specialized.
3. **Task difficulties**: Our benchmark is challenging and the *difficulty* of a task included has to be sufficient. If a task is easily solved by any algorithm, it will not be useful to compare various models used for evaluation. We evaluated several tasks based on existing literature and selected popular tasks that we believe are recently developed, challenging, and are not yet fully solved with existing approaches.

Experimental Results. Findings on BEIR as shown in Figure 2.2 crucially reveal that while dense retrievers outperform lexical methods such as BM25 on many datasets (e.g., MS MARCO [29], Natural Questions [165]), they fail to generalize reliably across diverse domains in BEIR. This observation challenges the prevailing assumption that improvements solely in-domain, i.e., MS MARCO [29] translate to general retrieval robustness, motivating subsequent work on model generalization, training data diversity, and evaluation design. Eversince the introduction of BEIR, it has been instrumental in improving generalization capabilities of several neural retrieval models [233, 139, 296, 106, 358].

2.3.2 Domain-specific Retrieval Benchmarks

Task Definitions. Argument retrieval is the task of ranking documents based on the topical relevance to argumentative queries (i.e., queries about debated topics like "Should

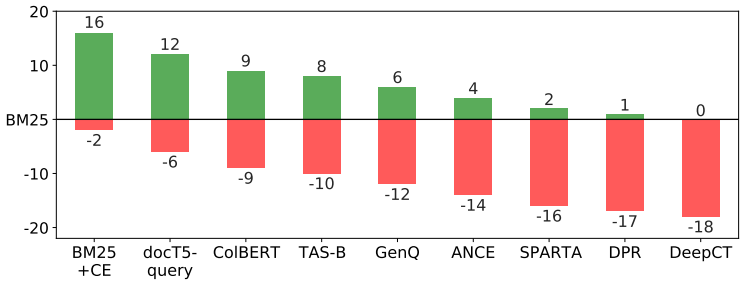


Figure 2.2: Comparison of neural retrieval and reranker models with BM25 on the BEIR benchmark; **green** denotes the number of datasets (out of 18) where the neural retrieval or reranker model (labeled on the x-axis) outperforms BM25; **red** is vice-versa.

bottled water be banned?"), i.e., the documents should contain appropriate arguments pertinent to the query. An argument is often modeled as a conclusion (i.e., a claim that can be accepted or rejected) and a set of supporting or attacking premises (i.e., reasons to accept or reject the conclusion like statistical evidence, an anecdotal example, etc.) [353, 311]. Code retrieval, is the task of ranking code snippets, or answers, to technical questions [136].

Argument Retrieval. Argument retrieval [256, 301] majorly made use of lexical retrieval models such as BM25 [282], DirichletLM [394], DPH [14], and TF-IDF [149]. These models were also commonly used to retrieve argumentative documents in argument search engines. For instance, popular argument search engines such as args.me [353], ArgumenText [311], and TARGER [63], all utilize BM25 for retrieving argumentative documents. The Touché 2020 dataset evaluating argument retrieval for controversial questions [35] is included as the argument retrieval subset in the BEIR benchmark for zero-shot evaluation of neural retrieval models in Thakur et al. [329]. Another related dataset is ArguAna [354], that evaluates the task of ranking top counter-arguments for a given argument.

Code-based Retrieval. Neural code generation [206] requires LLMs to generate code from scratch for generic programming questions. Existing datasets such as CQADupstack [130], LoTTE [296], and Stack Overflow-QA [184] retrieves the relevant answer snippet in response to a real user question on Stack Overflow. On the other hand, Neural Code Search [182], which contains simple questions to retrieve code snippets from GitHub on popular programming domains, such as Android. Finally, CodeSearchNet [136], COIR [184], LiveCodeBench [141], and CodeRAG-Bench [365] focus on the retrieval of high-level programming snippets on popular public repositories.

2.3.3 Multilingual Retrieval Benchmarks

Historically, there have been monolingual retrieval evaluations of search tasks in non-English languages, for example at TREC, FIRE, CLEF, and NTCIR. These community evaluations typically release test collections built from newswire articles, which provide only dozens of queries with modest amounts of relevance judgments and are insufficient for training neural retrieval models. These organizations also provide evaluation resources for cross-lingual retrieval, but they cover relatively few language pairs. For example, the TREC 2022 NeuCLIR Track [172] evaluates only three languages (Chinese, Persian, and Russian) in a cross-lingual setting.

Multilingual QA Datasets. XOR-TyDi [23] is a cross-lingual QA dataset built on TyDi by annotating answers in English Wikipedia for questions that TyDi considers unanswerable in the original (non-English) language. This setup does not allow researchers to examine *monolingual* retrieval in non-English languages. Another point of comparison is MKQA [204] and LAReQA [286], which convert XQuAD [21] and MLQA [179] into retrieval datasets. Questions are paired with exact answers in different languages, and evaluation is conducted in an open-retrieval setting by matching those answers in retrieved text; thus, MKQA and LAReQA are not "true" retrieval datasets. Furthermore, because the authors translate questions to achieve cross-lingual alignment, the translations may not be "natural", as noted by Clark et al. [71].

Synthetic Datasets. Since collecting human relevance labels is laborious and costly, other works adopt workarounds to build multilingual datasets. For example, Bonifacio et al. [41] automatically translate the MS MARCO dataset [29] from English into 13 other languages. Similarly, the BEIR benchmark [329] is translated into Polish [370], Hindi [4], Dutch [205], and other languages [8, 310]. However, translation is known to introduce artifacts such as "translationese" [71, 178, 345, 28, 93, 264], which may lead to training and evaluation data of questionable value. Alternatively, Sun et al. [319] build synthetic bilingual retrieval datasets in CLIRMatrix based on the parallel structure of Wikipedia, covering 139 languages. Constructing datasets automatically using heuristics avoids expensive human annotations and can be easily scaled to many languages. However, such datasets are inherently limited by the original resources from which they are constructed.

Multilingual Retrieval Datasets. Since MIRACL (introduced in Chapter 4) inherits from Mr. TyDi, it is useful to discuss the key differences between the two. Mr. TyDi [397] is a human-labeled retrieval dataset built on top of TyDi QA [71], covering 11 typologically diverse languages. While Mr. TyDi enables training and evaluation of monolingual retrieval, it has three shortcomings that we address in MIRACL: (1) *Limited positive documents*:

candidate documents for annotation are selected only from the top-ranked Wikipedia article based on a Google search. (2) *Inconsistent documents*: the Mr. TyDi corpora comprise a mix of TyDi QA documents and custom segments. (3) *Lack of typologically similar languages*: the 11 languages included in TyDi QA are deliberately chosen to span diverse typologies, belonging to 11 sub-families from 9 families.

Related Benchmarks. Extensions of the English MTEB benchmark [232] are widely adopted across languages. The Chinese MTEB (C-MTEB) [376] collects publicly available Chinese retrieval datasets. Similarly, Turkish MTEB (Tr-MTEB) [30], Russian MTEB (ruMTEB) [308], and the Scandinavian Embedding Benchmark (SEB) [95], among others, include either translated or non-translated retrieval datasets for specific language families. The MMTEB benchmark [94] aggregates many retrieval datasets across diverse languages, to construct a massive multilingual benchmark. Similarly, AfriQA [243] and CIRAL [5] are cross-lingual retrieval datasets with a focus on African languages, and XTREME-UP [288] contains a retrieval subset focused on low-resource Indic languages.

2.3.4 Retrieval Benchmarks Beyond the Classical Retrieval Task

Instruction-following & Reasoning Benchmarks. BERRI [24] and InstructIR [245], are some of the earliest benchmarks to evaluate the instruction-following ability in retrieval systems. Similarly, IFIR [309] extends the retrieval task on expert domains. BIRCO [362] evaluate retrieval systems based on multifaceted objectives, and RAR-b [374] adapts existing commonsense, math, and code datasets into a retrieval setting to test whether models can directly retrieve answers to reasoning problems. BRIGHT introduced in Su et al. [317] collects realistic user queries and align them with relevant natural documents from large corpora, to form a reasoning-intensive benchmark.

LLM-generated Synthetic Benchmarks. Cocktail [81] evaluates retrieval models on a mix-sourced data containing both web and LLM-generated documents within the document corpus. AIR-Bench [58] utilizes high-performing LLMs to generate test questions automatically across diverse domains, such as Law and Finance.

Long-context Retrieval Benchmarks. Needle in a haystack problem [154, 200] is a technique for testing the in-context retrieval ability of long context LLMs by randomly inserting the key information into an arbitrary position. LoCo [289] constructs a 12 task benchmark to measure long-context retrieval where chunking is not possible or not effective. Similarly, LongEmbed [405] is a long-context retrieval benchmark, including datasets such as NarrativeQA [161] containing queries on a corpus comprising long stories.

2.4 Retrieval Training Data

2.4.1 Pretraining on Weakly Supervised Datasets

Pretraining is an effective strategy to initialize a pretrained backbone model to learn the retrieval task effectively. The Inverse Cloze Task (ICT) [177] randomly selects a token span from a text segment to serve as the query, while the remaining tokens form the key. This approach proves effective for pretraining retrieval models [53, 305]. CoCondenser [108] pretrains by randomly sampling text spans between two documents and, by adding a head representation and skip connections to early layers, enforces the learning of global semantic representations of the input text. SimCSE [112] takes an input sentence and predicts itself under a contrastive objective. Contriever [139] uses MoCo [125] with random cropping as the pretraining objective and shows it to be more effective than ICT. Spider [271] uses spans that appear multiple times in a document to create pseudo-examples.

Contrastive Learning. E5 [358] curates large pretraining corpora such as Common Crawl and Reddit, constructs training pairs (e.g., title as query and text as document), and pretrains retrieval models using a contrastive loss with only in-batch negatives and large batch sizes, following similar approaches such as OpenAI embeddings [233] and BGE [376]. However, $\text{E5}_{\text{mistral}}$ [359] suggests that an expensive pretraining stage is not required when using a strong LLM as the backbone. Nonetheless, more recent retrieval models with LLM backbones, such as GTE [186], KaLM [133, 402], Jina [315, 118], and Qwen-3 embeddings [399], continue to rely on contrastive pretraining over diverse corpora. Nomic [242] pretrains using contrastive learning on 235 million pairs.[1]

2.4.2 Synthetic Training Data Generation

Synthetic data generation is a widely studied topic in IR research, with many methods proposed to enhance retrieval systems using artificially created data. Numerous SoTA retrieval models are trained on massive synthetic datasets, including $\text{E5}_{\text{mistral}}$ [359], Qwen-3 [399], Gecko [176], and Gemini embeddings [174]. As a result, synthetic training datasets play a crucial role in creating training data at scale for diverse domains and languages.

Synthetic Query Generation. Traditionally, docT5query [241] has been a prominent method for generating synthetic training queries in English and is widely used for unsupervised domain adaptation [187, 209, 329]. GPL [357] employs a cross-encoder to produce

[1] https://huggingface.co/datasets/sentence-transformers/embedding-training-data

pseudo-labels for query–document pairs, and Thakur et al. [328] extend this approach to adapt efficient retrieval models across domains.

LLM-based Data Generation. Bonifacio et al. [40] propose InPars, which uses few-shot prompting of GPT-3 [45] to generate synthetic queries. Similarly, several follow-up works [292, 143, 43, 291, 91] adopt a comparable setup. Dai et al. [83] propose Promptagator, which studies task-dependent few-shot LLM prompting and uses the resulting synthetic data for both retrieval and ranking models. DuQGen [52] generates representative queries by clustering target documents. In contrast, HyDE [110], Query2Doc [360], and GenRead [389] generate synthetic documents or answers instead of queries. BGE-M3 [59] uses GPT-3.5 to generate synthetic data, whereas $E5_{mistral}$ [359] uses GPT-4 to generate diverse synthetic data by first prompting an LLM to produce a diverse pool of candidate tasks and then generating data as $(\text{query}, \text{positive}, \text{hard negative})$ triplets conditioned on specific tasks. Gecko [176] proposes an LLM-based approach to identify better positive and hard negative documents. Lee et al. [174] use Gemini to generate synthetic queries for web documents, followed by a Gemini-based auto-rater to filter low-quality examples. Related work, such as Chen et al. [57], generates synthetic data using efficient LLMs as teachers, and Chitale et al. [66] evaluate different augmentation models at scale.

Synthetic Multilingual Datasets. Prior work investigates techniques for building multilingual datasets to improve fine-tuning and evaluation of dense retrieval models. Datasets such as NeuCLIR [172] and MKQA [204] are constructed using human annotations. Similarly, mMARCO [41] is generated via machine translation of MS MARCO [29]. CORA [26] leverages a generator to mine cross-lingual retrieval training data. Likewise, QuiCK [278] and Zhuang et al. [406] use a query generator to produce synonymous queries in other languages, while DR. DECR [185] uses parallel queries and sentences to perform cross-lingual knowledge distillation. Finally, JH-POLO [171] and Generate-Distill [171] prompt LLMs to generate English queries from language-specific documents in NeuCLIR.

2.4.3 Supervised Training Data & Artifacts

Supervised Training Datasets. Popular large-scale supervised training datasets originate with MS MARCO [29], which contains over 500K training pairs. Subsequently, the BGE collection [181] provides a comprehensive set of training datasets for retrieval (e.g., NQ, MS MARCO), clustering (e.g., TwentyNewsgroups), and classification (e.g., AmazonReviews) tasks. Nomic [242] fine-tunes its models on 1.6M supervised training pairs, including subsets from MEDI [316]. Qwen-3 embeddings [399] fine-tune on approximately 12M training pairs (including synthetic pairs) but do not release dataset details.

Sparsely Annotated Datasets. Popular IR training datasets, such as MS MARCO [29], are shallow-pooled and sparsely judged by human assessors [214, 20]. Assessors observe a small number of documents from a baseline retrieval system, select those that are relevant to the query, and label them as ground truth. In contrast, non-relevant judged documents (i.e., documents that are seen but ranked below the ground truth) are not provided. As a result, fine-tuning typically assumes that the remaining documents in the corpus are negatives, while a small set of mined documents similar to the query are labeled as hard negatives.

Pseudo-labeling. Instead of relying solely on human labels in supervised training datasets, pseudo-labeling addresses the problem of sparse annotations by employing alternative techniques to estimate query–document relevance. Examples include knowledge distillation from cross-encoders [263, 127, 357], ranking documents via LLM prompting [320], or using composite measures of embedding similarity with ground-truth documents [393].

False Negatives. False negatives are *unlabeled positive documents* that arise due to pooling bias in the dataset construction procedure, where human annotators judge and label only a small number of samples selected by baseline retrievers. Qu et al. [263] first highlight the issue of false negatives in retrieval datasets, noting that some hard negative documents should instead be classified as positives. Rather than explicitly *curating* the training datasets, RocketQA [263] fine-tunes models by distilling knowledge from cross-encoder scores for query–document pairs. Similarly, Cai et al. [49] train both a relevance model and a selection model to mitigate the impact of false negatives. Moreira et al. [228] and Meghwani et al. [220] examine filtering methods and selection criteria for negative sampling that avoid extremely hard negatives. Wang et al. [361] modify the training loss function using a contrastive confidence regularizer. In Gecko [176, 174], Gemini is used to relabel positive documents for synthetic queries and to identify better hard negatives.

2.5 Retrieval-Augmented Generation Background

After covering the necessary IR fundamentals in the previous sections, we now shift our focus to provide relevant background on Retrieval-Augmented Generation (RAG).

2.5.1 Task Definition: Retrieval-Augmented Generation

The knowledge stored in a large language model (LLM) has a cutoff date, beyond which an LLM has not been trained on new data, causing existing knowledge to saturate in

LLMs [124], and becoming prone to hallucinations by generating factually incorrect output [219, 273]. By grounding on external and up-to-date knowledge or data sources, retrieval-augmented generation (RAG) enables LLMs to generate better and more trustworthy output [119, 180, 139, 42, 25, 270, 303]. Instead of relying solely on memorization on training data, RAG retrieves documents at inference time and feeds them into the LLM as context.

Task Formulation. Mainstream RAG techniques follow a "*retrieve-then-read*" paradigm involving three stages: (1) **retrieval**: A retrieval system R first fetches relevant information at inference time from a corpus of D_n documents, for a given user query q, represented by $R(q) = [D_1, \ldots, D_k]$. (2) **augmentation**: The retrieved information along with the user query is *augmented* and passed as a prompt to the generation stage, $LLM(q \cup [D_1, \ldots, D_k])$. (3) **generation**: The generator LLM generates a long-form answer a conditioned on the given user query and retrieved information, as follows:

$$a = \arg\max_{y} P\big(y \mid \text{LLM}(q \cup [D_1, \ldots, D_k])\big) \tag{2.13}$$

RAG has achieved remarkable results in various tasks such as open-domain question answering (ODQA) [180, 139, 335], argument extraction [90] and code generation [404]. Many real-world web applications, such as Bing Search [224], LangChain [54], and many others, have incorporated RAG. Recently, more RAG paradigms have emerged extending the standard RAG framework [99, 148], however, for the scope of our thesis, we will focus solely on the "*retrieve-then-read*" paradigm or the naive RAG architecture.

2.6 RAG Evaluation Benchmarks and Collections

2.6.1 RAG Collections: Corpora & Topics

Corpus Selection. Current RAG datasets are typically constructed using English Wikipedia as the document collection. However, its scale is limited and does not provide sufficiently rich and comprehensive information to support RAG systems. A prominent alternative, ClueWeb22 [249], offers an extensive collection of 22 billion curated web pages and is previously utilized in TREC tracks such as the TREC Conversational Assistance Track (CAsT) [250] and the TREC Interactive Knowledge Assistance Track (iKAT) [12]. However, ClueWeb22 remains gated and poses a significant hurdle for researchers with limited computational resources or those unable to obtain access credentials. Another alternative is the MS MARCO V2 document collection, which is used primarily in the TREC Deep Learning (DL) track. The open availability of MS MARCO makes it an attractive option for researchers seeking to develop and evaluate RAG systems at scale.

Topic Selection. Recently, there is a surge in datasets that provide topics with long-form answers for evaluating RAG systems. ASQA [314], ELI5 [98], and QAMPARI [15] are used for evaluation in the Automatic LLMs' Citation Evaluation (ALCE) framework [113]. Similarly, related long-form QA datasets include AquaMuse [163], ExpertQA [216], and TruthfulQA [196]. Another recently introduced dataset is ClapNQ [283], which is created from a subset of Natural Questions (NQ) [165], and HAGRID [152], which is built on a subset of MS MARCO Dev [29]. Almost all existing datasets are built on English Wikipedia and are therefore prone to the overfitting observed in existing retrieval benchmarks [329, 232].

2.6.2 Related Benchmarks & Toolkits

Related RAG Benchmarks. Existing RAG systems are primarily proprietary and difficult to reproduce. Open-source frameworks such as LangChain [54] and LlamaIndex [198], while available, are not research-friendly and lack proper evaluation and benchmarking. FlashRAG [148] is a similarly motivated toolkit that aims to improve the RAG experience for researchers. Datasets that focus on unanswerable queries, such as SQuAD 2.0 [268], are created adversarially to resemble datasets with answerable queries. Similarly, conversational QA datasets such as CoQA [274] and QuAC [68] also contain unanswerable queries. RAGAs [96] introduces several metrics to evaluate faithfulness, answer relevance, and context relevance in RAG, leveraging LLMs (such as ChatGPT) for automatic evaluation based on custom prompts. ARES [290] demonstrates that fine-tuning an open-source LLM with a small amount of training data suffices to produce an effective automatic judge. Relatedly, eRAG [293] uses the LLM itself to evaluate retrieval effectiveness, and TruLens [336] provides automated evaluations similar to RAGAs for context relevance, groundedness, and answer relevance. Finally, Yu et al. [390] introduces a RAG benchmark to evaluate hallucination, which combines faithfulness and correctness evaluation.

RAG Frameworks and Toolkits. RAGChecker [287] is a fine-grained evaluation framework that incorporates a suite of diagnostic metrics for both retrieval and generation modules. Open RAG Eval [344] incorporates nugget, support, and relevance evaluation into an open-source framework. R-EVAL [337] provides RAG evaluations across domains.

2.6.3 Multilingual RAG Benchmarks

Prior work on RAG evaluation is conducted almost exclusively in English. For example, benchmarks such as ALCE [113], FreshLLM [352], ClapNQ [283], HAGRID [152], and CRAG [384] all include long-form answers for English-only queries and are based on

collections that draw documents from English Wikipedia, MS MARCO [29], or NQ [165]. Similarly, the TREC 2024 and 2025 RAG tracks, which are TREC competitions for RAG evaluation, focus on evaluating English queries.

Multilingual RAG. On the multilingual side, RAG remains underexplored in prior work. The RGB [59] and CRUD-RAG [208] benchmarks are limited in language scope, covering only one additional language: Chinese (zh). The NeuCLIR 2024 track [218] evaluates long-form report generation but is limited to three languages. Chirkova et al. [65], subsequently included in BERGEN [272], evaluate multilingual open-domain QA across 13 languages. MEMERAG [79] extends the MIRACL dataset [398] by generating answers in five languages using a diverse set of LLMs and collecting judgments on the faithfulness and relevance of the answers from native expert annotators.

2.7 RAG Evaluation Methodologies

Evaluating RAG systems entails assessing both retrieval and generation outputs. These evaluations are multifaceted and require careful consideration. Although not exhaustive, this section focuses on three aspects that are crucial for RAG evaluation: relevance (retrieval), support, and nugget (generation) evaluation.

2.7.1 Why Retrieval Metrics Are Insufficient

The shift from document retrieval to answer generation requires new evaluation strategies and the development of new metrics, representing one of the most significant challenges in RAG today. Traditional retrieval evaluation focuses on constructing test collections following the Cranfield paradigm and relies heavily on pooling and human relevance judgments [348, 47, 122, 48]. This setup is ill-suited for assessing the quality of generative outputs in RAG systems.

The traditional retrieval task assumes that the system output consists of a ranked list of documents, and evaluation focuses on whether relevant documents appear high in the ranking. This approach works when the goal is to help users find relevant documents. However, the goal of the RAG task is different: the system is expected to generate coherent, factually accurate, and comprehensive answers.

The challenges of evaluating generative outputs in RAG systems are multifaceted and complex. A generated paragraph may be fluent, well-written, and engaging, yet factually

incorrect. Conversely, a response may contain accurate information but miss key details that are essential for fully answering the user's question. Traditional retrieval metrics such as nDCG@k and Recall@k, which work well for document retrieval, do not readily translate to the evaluation of generated text.

2.7.2 LLM-as-a-Judge and Relevance Evaluation

LLM-as-Judge Evaluation. Using LLM judges [403] in place of human judges gains popularity for evaluating different aspects of RAG, such as factuality [162], retrieval quality [293], and robustness [307], including specific aspects such as contextual faithfulness [235]. Researchers also leverage LLMs to perform pairwise comparisons between RAG responses [265, 121, 258] and suggest incorporating grading rubrics with strong LLMs [363].

Relevance Evaluation. This line of work investigates the use of LLMs for automatic relevance judgments to evaluate retrieval relevance in RAG. Cuconasu et al. [80] conduct a comprehensive and systematic examination of retrieval strategies in RAG. Faggioli et al. [97] are among the earliest to discuss prompting LLMs to provide relevance judgments, exploring a spectrum of human–machine collaboration strategies and articulating arguments for and against different approaches. Through clear instructions, well-defined roles, and the use of multiple skilled judges, efforts to compare human judgments with LLM assessments continue to be refined [97, 221, 213, 339, 1]. Upadhyay et al. [341] explore open-source reproductions of LLM judges and prompting strategies in Thomas et al. [331], and compare them against human judges, for retrieval evaluation in popular datasets, such as TREC-DL [77].

Limitations of LLM-based Data and Evaluation. While LLMs enable scalable data generation and automatic evaluation, they have certain limitations. First, reliance on synthetic queries and judgments can introduce technical debt: such outputs inherit the biases, blind spots, and capability limitations of the underlying model, which may propagate into downstream training and evaluation. Second, the rapid evolution of LLMs implies that evaluation results anchored to a specific model version may have limited temporal validity. To mitigate these risks, we incorporate human validation where feasible, emphasize reproducible pipelines, and interpret results with awareness of model-dependent effects.

2.7.3 Support or Groundedness Evaluation

Support evaluation measures whether the information in a RAG answer is factually supported by its cited documents. We evaluate support at the sentence level in the answer. We assume an answer r segmented into n sentences, $r = \{a_1, \cdots, a_n\}$, where each answer sentence a_i

can contain at most m document citations, $a_i = \{d_i, \cdots, d_m\}$, each of which is drawn from a corpus. Support is computed as a function $f(a_i, d_j) = s_{i,j}$, where f is a human or LLM judge that produces a scalar value $s_{i,j}$ indicating the extent to which the cited document d_j supports sentence a_i.

Apart from RAG, support is primarily explored in the literature on summarization [167, 144] and natural language explanations [27, 304]. Prior work on support evaluation in RAG uses different automatic judges, including natural language inference (NLI) models [113], LLMs with prompting [96], and fine-tuned custom LLMs [290]. Wu et al. [371] study the tension between an LLM's internal prior and supporting incorrect contextual information. Similarly, Ming et al. [225] present an evaluation benchmark consisting of academic question answering (QA) datasets with human validation, and Liu et al. [199] evaluate the quality of proprietary search engine outputs using crowdsourced human judges.

2.7.4 Nugget or Fact-Based Evaluation

Nugget evaluation measures factual correctness in terms of the nuggets captured within a RAG answer. Introduced in the TREC QA Track in 2003 [350], the nugget evaluation methodology emphasizes the identification of *nuggets*, or essential facts relevant to a good answer. Vorhees et al. [349] demonstrate the importance of constructing nuggets to evaluate definition questions, thereby aligning complex question answering with retrieval-augmented generation. In both cases, queries require coherent answers drawn from information present in multiple relevant documents. Subsequent studies refine and extend the basic nugget evaluation methodology [191, 85, 194]. Lin et al. [189, 190] explore automated approaches for nugget evaluation.

The nugget methodology provides several key advantages for evaluation. First, it offers a more granular measure of information recall than traditional document-level metrics, allowing evaluators to assess not only whether a system finds relevant information, but also whether it includes specific facts that users find valuable. Second, it provides a framework for assessing completeness by measuring what proportion of important nuggets a system's response covers. Third, it accommodates the fact that multiple valid expressions of the same information may exist by focusing on underlying facts rather than surface wording.

Recently, the advent of LLMs enables researchers and practitioners to perform more reliable nugget evaluation automatically using LLMs as judges [218, 11, 326]. In Pradeep et al. [260], we explore nugget-based evaluation with the AutoNuggetizer framework, which applies LLMs to both automatically create nuggets and automatically assign nuggets to system answers.

Chapter 3

Revisiting Argument Retrieval Benchmarks

Substantial progress has been made in developing different types of neural retrieval models, including dense (e.g., [386, 177, 156, 128, 377]), sparse (e.g., [82, 401, 192, 106]), and multi-vector models (e.g., [159, 296, 108, 175]). However, surprisingly, *all* neural models continue to be less effective than BM25 on the Touché 2020 [35] subset of BEIR, an argument retrieval task; cf. Table 3.1 with results for BM25 and state-of-the-art neural retrieval models like $E5_{large}$ [358], CITADEL+ [183], SPLADEv2 [106], etc.

Motivated by this observation, in this chapter, we revisit the Touché 2020 dataset, the argument retrieval subset in BEIR, to understand the potential respective limits of current neural retrieval models. In the first stage, we conduct black-box evaluations (i.e., without requiring model retraining) to examine and possibly somewhat correct errors incurred by the neural models (Section 3.2.1). In the second stage, we carefully remove short documents (less than 20 words) from the Touché 2020 corpus (Section 3.2.2) and add missing judgments following the Touché guidelines (Section 3.2.3). Lastly, we attempt to answer the BM25 effectiveness via IR axioms (Section 3.2.4). We ask the following research questions in this chapter:

RQ1 *Why do neural models continue to underperform BM25 on the Touché 2020 dataset?*
RQ2 *Does neural retrieval model effectiveness improve by filtering the Touché 2020 document corpus?*
RQ3 *Are neural retrieval models unfairly penalized on Touché 2020 due to selection bias?*
RQ4 *Can retrieval axioms explain the BM25 effectiveness over neural retrieval models?*

Model	Reference	Type	nDCG@10
BM25 (BEIR)	Thakur et al. [329]	lexical	**0.367**
$E5_{large}$	Wang et al. [358]	dense	0.272
BGE-large	Xiao et al. [376]	dense	0.266
Promptagator	Dai et al. [83]	dense	0.266
DRAGON+	Lin et al. [195]	dense	0.263
GTR-XXL	Ni et al. [236]	dense	0.256
GPL	Wang et al. [357]	dense	0.255
RocketQAv2	Ren et al. [279]	dense	0.247
ANCE	Xiong et al. [377]	dense	0.240
RetroMAE	Xiao et al. [375]	dense	0.237
Contriever	Izacard et al. [138]	dense	0.204
TART-dual	Asai et al. [24]	dense	0.201
TAS-B	Hoffstätter et al. [128]	dense	0.162

Model	Reference	Type	nDCG@10
BM25 (BEIR)	Thakur et al. [329]	lexical	**0.367**
CITADEL+	Li et al. [183]	mult.-vec.	0.342
XTR (XXL)	Lee et al. [175]	mult.-vec.	0.309
CITADEL	Li et al. [183]	mult.-vec.	0.294
COIL-full	Gao et al. [109]	mult.-vec.	0.281
ColBERTv2	Santharam et al. [296]	mult.-vec.	0.263
ColBERT	Khattab et al. [159]	mult.-vec.	0.202
uniCOIL	Lin et al. [192]	sparse	0.298
SPLADEv2	Formal et al. [106]	sparse	0.272
SPLADE++	Lassance et al. [168]	sparse	0.244
DeepCT	Dai et al. [82]	sparse	0.175
SPARTA	Zhao et al. [401]	sparse	0.156

Table 3.1: Dense (left) and multi-vector / sparse (right) retrieval models on the BEIR Touché 2020 subset; nDCG@10 scores taken from the referenced publications.

3.1 Touché 2020 Experimental Setup

Touché 2020. The Touché 2020 task on controversial argument retrieval [35] uses a focused crawl of arguments for 49 test queries addressing socially important (and often controversial) issues like "Should bottled water be banned?". The document collection is the args.me corpus [10] containing 382,545 arguments. Each argument has a title in the form of a conclusion (i.e., a claim that an arguer could make) and a context containing several premises (reasons, opinions, or evidence that support or attack the claim). Arguments can be of varied length, they can span both sentences or multiple paragraphs. The task organizers also publish relevance judgments (non-relevant, relevant, and highly relevant) for 2,214 documents (cf. Table 3.4 for dataset characteristics). The documents are pooled using the top-5 pooling strategy from 12 ranked results submitted by participants. While the argument retrieval track in Touché has more recent versions [39, 37, 36], in our work, we use the Touché 2020 dataset, due to its availability in the BEIR benchmark [329].

Retrieval Models. For our experiments, we select different open-source retrieval models to better understand errors across different neural architectures. The selected models are the lexical model BM25 [282], the dense retrievers DRAGON+ [195], Contriever [138], and TAS-B [128], the sparse retriever SPLADEv2 [106], and the multi-vector retriever CITADEL+ [183].

Evaluation. To evaluate retrieval effectiveness on Touché 2020, we use nDCG@10 metric as it has been widely adopted in the BEIR benchmark [329]. In addition, we use the hole@k rate (i.e., the fraction of results retrieved by a model at cutoff k that do not have relevance judgments) to estimate the proportion of unjudged documents.

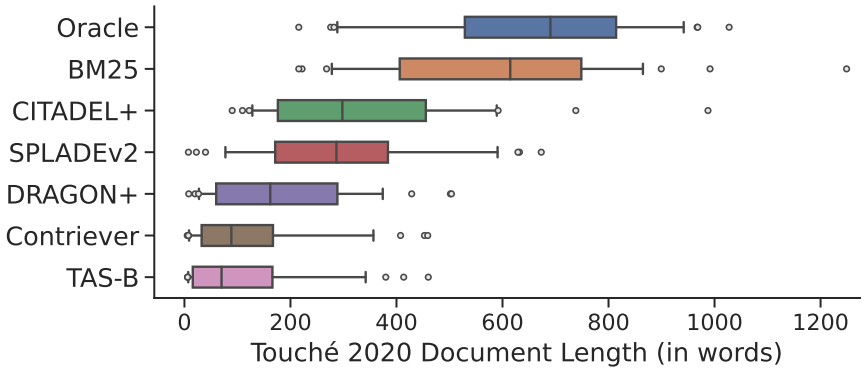


Figure 3.1: Boxplots showing the average length in words (x-axis) of the top-10 Touché 2020 results retrieved by the models on the y-axis (sorted by decreasing nDCG@10; oracle: avg. length of all documents judged as relevant). The results retrieved by neural models are much shorter in comparison to BM25.

Implementation Details. First, we conduct a reproducibility study with previously available models' checkpoints. We do not retrain any neural model and use up to a maximum of A6000 $\times$ 4 GPUs for inference. For BM25, we follow Thakur et al. [329] and use the multi-field (title and body indexed separately with equal weights) setting[1] available in Anserini [383] with default parameters ($k_1 = 0.9$ and $b = 0.4$). For our dense models, Contriever (mean pooling with dot product), TAS-B, and DRAGON+ (both `[CLS]` token pooling with dot product), we reproduce the results by converting model checkpoints using `sentence-transformers`[2] and evaluate them on Touché 2020 using BEIR evaluation.[3] For SPLADEv2 (max aggregation), we reproduce the model using the SPRINT toolkit [330].[4] Finally, for CITADEL+ (with distillation and hard negative mining), we use the original `dpr-scale` repository for reproduction.[5] Apart from DRAGON+, in our work, we successfully reproduce the nDCG@10 on Touché 2020 for all tested models.[6]

[1] https://github.com/castorini/anserini
[2] https://github.com/UKPLab/sentence-transformers
[3] https://github.com/beir-cellar/beir
[4] https://github.com/thakur-nandan/sprint
[5] https://github.com/facebookresearch/dpr-scale/tree/citadel
[6] For DRAGON+, we suspect the difference being caused by using A100 vs. A6000 GPUs.

BM25: <Social security should be privatized> Social Security has serious issues [...]
CITADEL+: <Social security should be privatized> Social security is a complete joke. Although it was originally designed [...] the young are forced to subsidize the old, a facet of socialism [...]

(a) Relevant documents

SPLADEv2: <Social Security R.I.B Should be Privatized> Thank you lannan13 for an invigorating debate.
DRAGON+: <Social Security R.I.B Should be Privatized> Pass
Contriever: <Social Security R.I.B Should be Privatized> Thank you lannan13 for an invigorating debate.
TAS-B: <Privatizing social security> Social security is in crisis

(b) Non-relevant documents

Table 3.2: Example of the top-ranked document for the query *"Should social security be privatized?"* showing that neural models may retrieve documents with a relevant conclusion / title (within < >) but a non-relevant premise / body.

3.2 Evaluation Experiments

In this section, we describe our evaluation experiments consisting of two independent parts. First, we conduct a black-box evaluation to understand the limitations of neural models on Touché 2020 (Section 3.2.1). Next, we denoise the data by filtering out short documents (Section 3.2.2) and conduct post-hoc relevance judgments (Section 3.2.3) to measure the unbiased nDCG@10 of neural models versus BM25 on Touché 2020. Finally, we attempt to understand our findings using information retrieval axioms (Section 3.2.4).

3.2.1 Black-Box Model Evaluation on Touché 2020

The neural retrieval model's training often involves one or several of the following steps, a particular training dataset selection [29, 165], choosing a training optimization objective [156, 127] and deciding whether to train with specialized hard negatives [263, 128]. These configurations are crucial for neural model effectiveness but lack explainability. Hence, our objective is to uncover the reasons for errors of retrieval models (BM25 vs. neural models) on Touché 2020, by treating models as black-boxes (without modifying parameters).

Quantitative Results. Figure 3.1 shows boxplots depicting the average document lengths of the top-10 retrieved documents by the models under investigation, where the whiskers plot the 95% confidence interval. The lengths are computed as word counts after applying `nltk` word tokenizer [33]. All neural models, on average, retrieve shorter documents containing less than 350 words (visible from medians and whiskers in Figure 3.1) in contrast to BM25,

Model	BM25	CITADEL+	SPLADEv2	DRAGON+	Contriever	TAS-B
Top-1	0.0%	6.1%	22.4%	40.8%	55.1%	59.2%
Top-5	0.4%	3.3%	15.9%	32.7%	40.4%	59.2%
Top-10	0.8%	4.5%	14.6%	26.5%	35.9%	51.6%

Table 3.3: Error rates as the percentage of a model's top-k results that are non-relevant (judgment of 0 or unjudged) and shorter than 20 words. Lower error rates are better.

which retrieves longer documents on average containing more than 600 words which best mimics the Oracle distribution. Dense models (TAS-B, Contriever, and DRAGON+) appear to retrieve the shortest arguments, followed by sparse (SPLADEv2) and muti-vector (CITADEL+). The decrease in nDCG@10 on Touché 2020 is found to be *perfectly correlated* with the increase in shorter top-10 retrieved documents (Spearman correlation $\rho = 1.0$). We suspect the retrieval of shorter arguments is due to the bias of the argument structure which we discuss in the next paragraph. Overall, this provides positive evidence for our hypothesis that the shorter documents present in Touché 2020 (cf. Figure 3.2a) negatively affect neural models in terms of effectiveness.

Empirical Evidence. Upon a careful analysis of the retrieved documents by the models under investigation, we observe an interesting pattern across the retrieved documents. We find that documents retrieved in Touché 2020 by neural models show a high overlap of the query terms with the argument conclusion (document title) which is often relevant, but includes a rather "noisy", i.e., short argument premise (document body) which is non-relevant, e.g., a single word "Pass" or "social security is in crisis" (an example for a test query is shown in Table 3.2). To quantify the empirical evidence, we compute an error rate (in %) by counting a mistake the model makes if the document retrieved (i) is non-relevant (relevance either 0 or unjudged), and (ii) is shorter than 1–2 sentences (a maximum of 20 words). From Table 3.3, we observe that dense retrievers suffer the most with TAS-B with the highest 51.6% error rate in top-10 retrieved documents. CITADEL+ contains a lower percentage of shorter non-relevant documents with a low 4.5% error rate. BM25 has the lowest error rate of 0.8%, which suggests that BM25 is empirically found to be robust against non-uniformity in document lengths present within the Touché 2020 corpus.

Reasoning. We hypothesize reasons for the observed error pattern. We start by assuming query q_i and the short non-relevant document $\hat{d}_i$ have a high word overlap due to similarity with the conclusion (title), i.e., the document is a good paraphrase of the query but does not contain information to answer the question.

As shown previously in Ram et al. [270], lexical overlap remains a highly dominant signal for relevance in dense retrievers, which we suspect causes the non-relevant short document, with a similar length to the query, closer within the dense embedding space representation. For sparse and multi-vector retrievers, the token overlap of query q_i and the shorter non-relevant document is high, which results in a higher similarity score. Neural retrieval models are fine-tuned on training datasets with documents of uniform length (e.g., MS MARCO), therefore have not seen varied document lengths. However, in contrast, the BM25 algorithm accounts for document length normalization within its parameter b [306]. As longer documents tend to have more term occurrences, leading to potential bias, document length normalization in BM25 acts as a useful parameter, improving robustness against sensitivity in document lengths. Neural models, conversely, suffer from noise present in the form of short and non-relevant documents in Touché 2020.

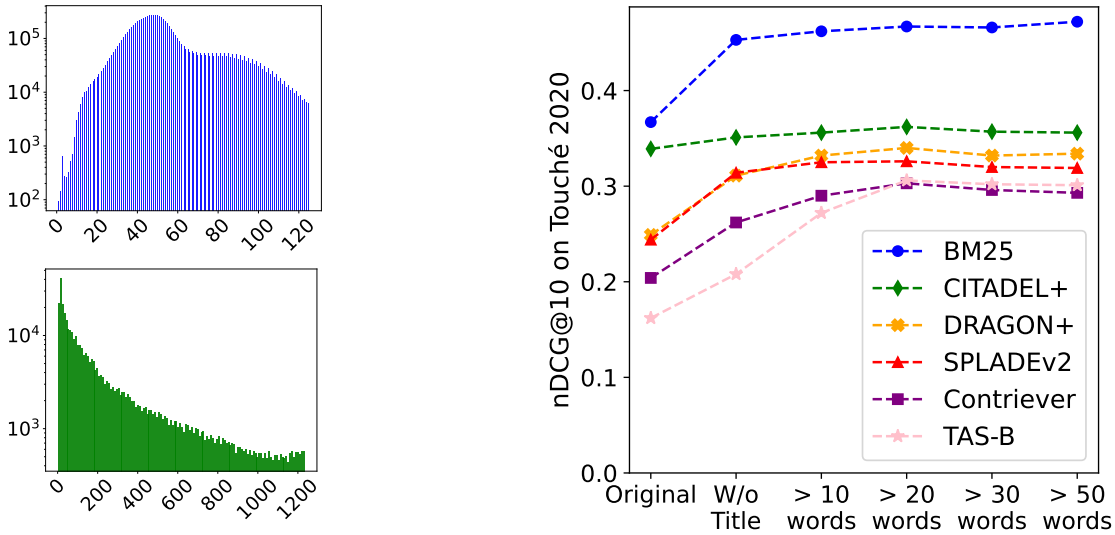
Answer to RQ1: *We observe that shorter documents present in Touché 2020 negatively affect all neural models in terms of effectiveness. Due to the document length normalization, BM25 is robust against non-uniform document lengths present in the Touché 2020 corpus.*

3.2.2 Filtering Short Documents in the Touché 2020 Corpus

As discussed in Section 3.1, the args.me corpus in Touché 2020 contains web-crawled arguments from various debating portals and thereby may contain noise as non-valid arguments. However, a valid document premise (or body) should provide evidence or reasoning that can be used to back up the conclusion (or title) as an argument [312, 333, 321]. But very short premises that are less than 1–2 sentences (e.g., "Pass" or "I agree") do not contain enough evidence to be classified as a valid argument.

To better understand the Touché 2020 corpus, we compare its document length distribution to the standard retrieval dataset MS MARCO [29]. The plots in Figure 3.2a show that the length distribution of Touché 2020 monotonically decreases with a high frequency of extremely short arguments (spike in the graph at 20–30 words) and a long tail of long arguments (even exceeding 1200 words), while the MS MARCO length distribution is much narrower with relatively few extremely short outliers.

The hypothesis for investigating **RQ2** lies in whether the effectiveness of neural models can be improved by cleaning, i.e., reducing noise in the Touché 2020 document corpus. To validate this, we filter out non-argumentative documents from the corpus. One way is to use argument classification to classify each document [277, 311] as either a valid or non-valid argument, however, it is computationally expensive to classify all arguments in Touché 2020 [277, 116]. Instead, we follow a simple heuristic and filter out potentially



(a) Document length distributions

(b) Heuristic filtering based on document length

Figure 3.2: **Left:** MS MARCO (blue, top) vs. Touché 2020 (green, bottom) document length distributions. The x-axis shows document length in words; the y-axis is log-scaled frequency. **Right:** Heuristic filtering experiment to determine the best threshold n for filtering short documents in Touché 2020.

non-valid arguments based on the document length. Our filtering technique removes the conclusion (across all documents in Touché 2020) and selects documents with premises greater than a threshold of at least n words in length.

Results after Filtering. Figure 3.2b shows that our filtering technique improves the nDCG@10 for all models. That removing the argument conclusion (i.e., title) alone improves the nDCG@10 for all models is probably caused by the inherent nature of argument retrieval, where premises are more important for a document to be classified as a valid argument than the conclusion. Without the conclusion, often also the lexical overlap with the query that confuses neural models (cf. Section 3.2.1) is decreased. A document length threshold of $n = 20$ words empirically yields the best nDCG@10 across all tested models, as the effectiveness saturates when removing premises with more than 20 words.

A limitation of filtering Touché 2020 with document length, is that we miss out on a few human-judged query-document pairs with document lengths shorter than 20 words. However, as Table 3.4 shows, overall 89% (382 out of 429) of the missed judgments were originally non-relevant (score 0), and only 3.7% (16 out of 429) are highly relevant (score 2). This suggests that shorter documents in the Touché 2020 corpus are likely to be non-relevant,

	Original	+ Filtered	++ Post-hoc
# **Documents**	382,545	303,732	303,732
Avg. length	293.5	358.7	358.7
# **Queries**	49	49	49
# **Judgments**	2,214	1,785	2,849
# **Relevance** = 2	636	620 (16 ↓)	1,136 (516 ↑)
# **Relevance** = 1	296	265 (31 ↓)	576 (311 ↑)
# **Relevance** = 0	1,282	900 (382 ↓)	1,137 (237 ↑)

Table 3.4: The Touché 2020 dataset characteristics before and after filtering and post-hoc judgments. Reported are the total number of documents, the average document length, the number of queries, the number of relevance-judged documents, and the number of documents per relevance grade: non-relevant (0), relevant (1), and highly relevant (2).

hence document length is a good and simple heuristic for checking valid arguments.

Answer to RQ2: *We observe that filtering by document length, by removing the argument title and very short or non-argumentative premises (less than 20 words) in the Touché 2020 corpus improves retrieval accuracy for all neural retrievers and BM25.*

3.2.3 Adding Post-hoc Relevance Judgments

Retrieval datasets can contain multiple biases induced by either the annotation guidelines, annotation setup, or human annotators. For instance, to avoid selection bias [197] in later studies using some retrieval dataset, popular information retrieval challenges, for instance at TREC [78, 172], aim to encourage the submission of diverse retrieval approaches to yield diverse judgment pools.

To quantify the selection bias in Touché 2020, we compute how many of the top-10 results of our tested models are unjudged in the original and denoised corpus versions. Table 3.5 shows that the respective hole@10 values all are greater than 50% (i.e., more than half of the top results of every model are unjudged in the Touché 2020 data).

Annotation Details. We conduct a post-hoc relevance judgment study to fill up the hole@10 across all tested models, i.e., annotating originally unjudged arguments, as filling up holes would account for denser judgments and a better estimate of nDCG@10. We hired 5+ annotators (graduate-level students at the University of Waterloo and the authors of the original publication) with prior debating experience and follow annotation guidelines

	Original		+ Filtered		++ Post-hoc	
Model	nDCG@10	hole@10	nDCG@10	hole@10	nDCG@10	δ inc.
BM25	**0.367**	61.6%	**0.467**	51.8%	**0.785**	△ 0.418
CITADEL+	0.339	60.2%	0.362	62.5%	0.703	△ 0.364
SPLADEv2	0.272	66.3%	0.326	63.3%	0.679	△ 0.407
DRAGON+	0.249	69.2%	0.340	63.9%	0.718	△ 0.469
Contriever	0.205	71.4%	0.303	65.9%	0.650	△ 0.445
TAS-B	0.162	77.8%	0.306	67.5%	0.682	△ 0.520

Table 3.5: Retrieval effectiveness as nDCG@10 and missing judgments as hole@10 on the original, filtered (cf. Section 3.2.2), and post-hoc judged (cf. Section 3.2.3) Touché 2020.

available in Bondarenko et al. [35]. We conduct the post-hoc judgments and fill up hole@10 for all tested models by evaluating each unjudged document with three relevance labels: 0 (non-relevant), 1 (relevant), and 2 (highly relevant). It cumulatively took 10–15 hours to judge 1,064 judgment pairs and we pay an hourly rate of 20 CAD per hour (except the authors of the original publication). Table 3.4 contains Touché 2020 statistics before and after the denoising and post-hoc judgment rounds. In our post-hoc judgment round, over 78% of the judgment pairs are judged relevant (with 48% highly relevant and 30% relevant), indicating that many "relevant" documents are retrieved by models but unjudged originally in Touché 2020. We measure the inter-annotator agreement score with Fleiss' κ [105]. Since argument retrieval is highly subjective and biased towards annotator preferences and beliefs [35, 134], we achieve a comparable score of $\kappa = 0.31$.

Results after Post-hoc Judgments. Re-evaluation scores of the retrieval models after post-hoc judgment rounds are shown in Table 3.5 (column '++ Post-hoc'). The maximum increase in nDCG@10 is observed in dense retrievers (TAS-B, Contriever, and DRAGON+) and the least in multi-vector retrieval with CITADEL+. This provides evidence that post-hoc relevance judgments to fill up holes are necessary for a fair evaluation of models. In our hypothesis, we suspected a bias towards lexical retrievers due to their dominance in the original candidates during original Touché 2020 judgment rounds. However, even after post-hoc relevance judgments with more and better "semantic", i.e., neural retrieval models, and denoising Touché 2020, BM25 continues to outperform all neural models by a margin of at least 6.7 points on nDCG@10, thereby making it still a robust baseline.

Answer to RQ3: *Re-evaluation after denoising the Touché 2020 corpus and adding post-hoc judgments, the effectiveness of all neural models substantially improves. However, BM25*

	BM25	CITADEL+	SPLADEv2	DRAGON+	Contriever	TAS-B
w/ title	99.6	75.3	60.6	39.2	41.8	35.2
w/o title	99.5	79.1	68.2	39.5	40.8	38.9

Table 3.6: Agreement (in %) with the length normalization axiom LNC2 when retrieving with (w/) or without (w/o) the title on Touché 2020. BM25 agrees perfectly with LNC2.

remains more effective than all neural models after filteration and post-hoc judgments.

3.2.4 Axiomatic Error Analysis on Touché 2020

Axiomatic analysis in information retrieval formalizes desirable properties of ranking functions as a set of constraints, or *axioms*, that a well-behaved retrieval model should satisfy [38]. These axioms encode intuitive retrieval heuristics, such as preferring documents with higher query term frequency (e.g., TFC1), appropriate document length normalization (e.g., LNC1), or stronger semantic similarity (e.g., STMC1). The central premise, often referred to as the axiomatic relevance hypothesis, is that retrieval models satisfying more of these constraints are more likely to perform well empirically. We advise the reader to refer to Bondarenko et al. [38] for formal definitions of individual IR axioms.

To contrast our previous empirical evaluation of neural retrieval models on Touché 2020 with well-grounded theoretical foundations of information retrieval, we investigate if we can observe similar trends using axiomatic analysis. Therefore, we measure the agreement of the neural models under investigation with information retrieval axioms. A higher agreement indicates that a retrieval model fulfills the theoretical constraint introduced in the axiom. These axioms can highlight the problems in neural models, and fixing these problems can improve the model's effectiveness [38], even when there is no strong correlation between axioms and relevance judgments [50]. While retrieval axioms can increase the effectiveness of neural retrieval models (e.g., when used for regularization [285]), dedicated axioms for neural retrieval models are still missing [346].

Setup and Background. We conduct our axiomatic analysis using the `ir_axioms` framework [38].[7] Because most axioms require theoretical preconditions that are rarely met in real-world datasets (e.g., requiring document pairs retrieved for the same query of identical length) [38], we first use synthetic document pairs derived from real documents and subsequently use real document pairs with the default length relaxation from `ir_axioms`.

[7] https://github.com/webis-de/ir_axioms

Model	Term Frequency			Doc. Length		Semantic Sim.	
	TFC1	TFC3	M-TDC	LNC1	TF-LNC	STMC1	STMC2
BM25 (O)	61.6	100.0	51.8	37.8	58.5	48.3	54.9
BM25 (+D)	68.5	100.0	55.6	32.8	57.4	48.4	50.9
CITADEL+ (O)	59.2	88.9	56.6	54.3	60.7	50.7	57.9
CITADEL+ (+D)	62.6	72.7	47.6	56.1	57.1	51.0	57.8
Contriever (O)	59.7	100.0	46.5	52.7	55.9	52.5	59.1
Contriever (+D)	59.4	80.0	51.4	52.5	57.7	52.6	54.3
DRAGON+ (O)	61.1	100.0	50.6	55.3	59.0	52.1	58.2
DRAGON+ (+D)	63.2	92.3	54.7	53.1	55.4	52.2	54.5
SPLADEv2 (O)	59.8	50.0	57.1	47.8	56.2	50.8	56.6
SPLADEv2 (+D)	62.9	91.7	53.0	51.5	57.8	51.3	55.2
TAS-B (O)	60.1	33.3	55.4	50.3	55.8	52.3	60.5
TAS-B (+D)	62.2	33.3	50.6	54.0	53.1	52.4	54.2

Table 3.7: Agreement (in %) with the term frequency, document length, and semantic similarity axioms for all models on the original (O) and the denoised (+D) data.

Given more than 20 previously proposed retrieval axioms [38], we include a subset of all axioms related to document length, term frequency, and semantic similarity in our analysis. We focus on document length axioms following our observation that document length plays an important role in Touché 2020, while we include term frequency and semantic similarity because they are the specialty of lexical and neural retrieval models. In all cases, we report the agreement in the percentage of the model under investigation with the preferences of an axiom as implemented in `ir_axioms`.

Axiomatic Analysis on Synthetic Document Pairs. Table 3.6 shows the agreement of the tested models with the document length normalization axiom LNC2 that (somewhat artificially) states that the relevance score of an m-times self-concatenation of a document should not be lower than the original document's relevance score [101]. We synthetically create document pairs that fulfill this precondition by randomly sampling 250 query–document pairs from the top-10 ranked results by all models under investigation. For each query–document pair, we create pairs for $m = 1$, 2, 3, and 4. We observe that BM25 almost perfectly agrees with the LNC2 axiom (agreement above 99%), whereas neural models substantially violate LNC2, with TAS-B having the highest disagreement, which is an expected shortcoming of TAS-B as all documents are, independent of their length, represented by vectors of the same length.

Axiomatic Analysis on Real Document Pairs. Table 3.7 shows the results of our

axiomatic analysis on all document pairs from the top-50 ranked results for each test query on both the original (O) and the denoised (+D) Touché 2020 corpus. We report the term frequency axioms TFC1 [100], TFC3 [101] (we leave out TFC2 [101] because this axiom can only be applied on synthetic documents), and TDC [101], the document length axioms LNC1 [101] and TF-LNC [101], and the semantic similarity axioms STMC1 [102] and STMC2 [102]. We observe that BM25 has the highest agreement with the term frequency axioms TFC1 and TFC3 which are more frequently violated by the other neural models. For the M-TDC, LNC1, and TF-LNC axioms, BM25 achieves only mediocre agreement. Similarly, BM25 does not agree well with the semantic similarity axioms STMC1 and STMC2, where neural models outperform BM25, for which this could be expected (BM25 alone suffers from vocabulary mismatch in contrast to neural models), which indicates that those axioms play a subordinate role on Touché 2020.

Answer to RQ4: *IR axioms on the Touché 2020 data find that all neural models violate the document length normalization axiom LNC2, which is well-supported by BM25.*

3.3 Summary

In this chapter, we revisited the BEIR subset Touché 2020, an argument retrieval task. We addressed the question of why neural models are subpar, compared to BM25 on Touché 2020. To this end, we conducted a systematic error analysis and found that neural models often retrieve short and non-relevant arguments.

To alleviate this issue, we enhanced data quality by filtering out noisy and short arguments in Touché 2020 and by including post-hoc judgments to fill up holes and enable a fair evaluation of all tested models. Although these amendments improved the effectiveness of neural models by up to a margin of 0.52 in nDCG@10, they continue to lag behind BM25. Coupled with our theoretical analysis, we highlighted that all neural models violate the document length normalization LNC2 axiom, which is intuitively explained by the fact that documents are mapped to equal-size vectors.

In the next chapter, we will examine how to construct a retrieval benchmark that is useful for accelerating multilingual retrieval across a wide variety of languages.

Chapter 4

Benchmarking Multilingual Retrieval Across a Continuum of Languages

We have recently witnessed many efforts to build resources for non-English languages, for example, CLIRMatrix [319], XTREME [132], MKQA [204], mMARCO [41], TyDi QA [71], XOR-TyDi [23], and Mr. TyDi [397]. These initiatives complement multilingual retrieval evaluations from TREC, CLEF, NTCIR, and FIRE which focus on specific language pairs. Nevertheless, there remains a paucity of resources for languages beyond English. Existing datasets are far from sufficient to fully develop information access capabilities for the 7000+ languages spoken on our planet [151].

In this chapter, we take a small step toward addressing these issues. To stimulate further advances in multilingual retrieval, we build the MIRACL dataset on top of Mr. TyDi [397], comprising of human-annotated passage-level relevance judgments on Wikipedia for 18 languages, totaling over 726k query–passage pairs for 78k queries. These languages are written in 11 distinct scripts, originate from 10 different language families, and collectively encompass more than three billion native speakers around the world. Compared to existing datasets in Table 4.1, MIRACL provides more thorough and robust annotations and broader coverage of the languages, including both typologically diverse and similar language pairs. We ask the following three important research questions in this chapter:

RQ1 *How to construct a reliable, high-quality and large-scale monolingual retrieval dataset?*
RQ2 *How do unsupervised and supervised multilingual retrievers perform on such a dataset?*
RQ3 *Do we observe cross-language transfer in monolingual retrieval within such a dataset?*

Dataset Name	Natural Queries	Natural Passages	Human Labels	# Lang	Avg # Q	Avg # Labels/Q	Total # Labels	Training?
NeuCLIR [172]	✓	✓	✓	3	160	32.74	5.2k	×
MKQA [204]	×	✓	✓	26	10k	1.35	14k	×
mMARCO [41]	×	×	✓	13	808k	0.66	533k	✓
CLIRMatrix [319]	×	✓	×	139	352k	693	34B	✓
Mr. TyDi [397]	✓	✓	✓	11	6.3k	1.02	71k	✓
MIRACL (our work)	✓	✓	✓	18	4.3k	9.23	726k	✓

Table 4.1: 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 training data for neural models.

4.1 Dataset Overview

MIRACL is a multilingual retrieval dataset that spans 18 different languages and focuses on the monolingual retrieval task. In total, we gather over 726k manual relevance judgments (i.e., query–passage pairs) for 78k queries across Wikipedia in these languages, where all assessments are performed by native speakers. MIRACL provides a substantial number of examples that can support the training and evaluation of neural retrieval models. Detailed statistics for MIRACL are shown in Table 4.2.

Among the 18 languages in MIRACL, 11 are existing languages that are already 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 generate new queries for all languages; however, since we inherit Mr. TyDi queries, fewer new queries are generated for the 11 existing languages as these languages already have 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 [397], 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 are introduced in Section 4.2.2.

- **Training and development sets:** For the 11 existing Mr. TyDi languages, 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 4.2: Descriptive statistics for all languages in MIRACL, organized by split. # **Q**: number of queries; # **J**: number of human judged pairs (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.

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 comprise entirely new queries that have never been released before (compared to test-A sets, whose queries ultimately draw from TyDi QA 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

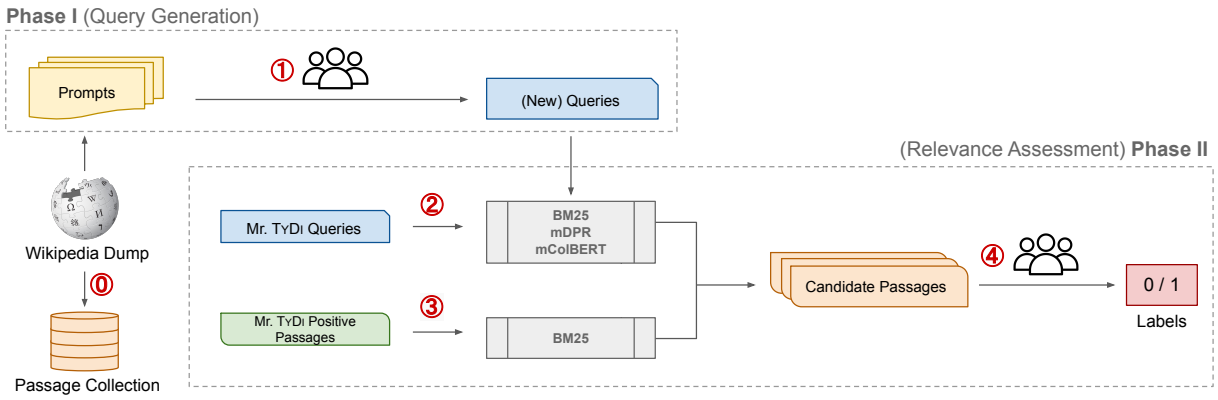


Figure 4.1: Diagram illustrating the MIRACL annotation workflow.

relevant passages for some queries from Mr. TyDi; we refer to these as "invalid" queries and remove them from the corresponding MIRACL splits (details in Section 4.4.2). These are still useful queries and utilized in Chapter 10 for relevance assessment in multilingual RAG. To evaluate the system outputs, we use standard IR metrics, nDCG@10 and Recall@100.

4.2 Dataset Construction

To build MIRACL, we hire native speakers as annotators to provide high-quality queries and relevance judgments. At a high level, our workflow comprises two phases: First, the annotators are asked to generate well-formed queries based on "prompts" [71] (details in Section 4.2.2). Then, they are 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 the dataset is *not* constructed via crowd-sourced workers, unlike other previous efforts such as SQuAD [269]. Instead, we hire 31 annotators (both part-time and full-time) across all languages. All annotators were recruited via a LinkedIn job posting, and required to be physically located in Canada for taxation purposes. Each annotator was interviewed prior to being hired and was verified to be a native speaker of the language they work in. Our team created an consistent onboarding process that included carefully crafted training sessions. Our team began interviewing annotators in mid-April 2022; dataset construction began in earnest in late April 2022 and continued until the end of September 2022.

Throughout the annotation process, we constantly check randomly sampled data to

monitor annotation quality (see Section 4.3). Whenever issues were detected, we promptly communicate with the annotators to clarify the problems so that they are resolved expeditiously. Constant interaction with the annotators helps minimize errors and misunderstandings. We believe that this design yields higher-quality annotations than what could be obtained by crowd-sourcing.

In total, MIRACL represents over 10k hours of assessor effort, or around five person-years. We offer annotators an hourly rate of $18.50 per hour (converted into USD). For reference, the local minimum wage is $11.50 USD/hr during the period of April to September 2022.

4.2.1 Corpora Preparation

For each MIRACL language, we prepare a pre-segmented passage corpus from a raw Wikipedia dump. For the existing languages in Mr. TyDi, we use exactly the same raw Wikipedia dump as Mr. TyDi and TyDi QA from early 2019. For the new languages, we use releases from March 2022. We parse the Wikipedia articles using WikiExtractor[1] and segment them into passages based on natural discourse units using two consecutive newlines in the wiki markup as the delimiter. Query–passage pairs form the basic annotation unit (Step ⓪ in Figure 4.1).

4.2.2 Annotation Workflow

MIRACL is created using a two-phase annotation process adapted from TyDi QA, which is in turn built on best practices derived from Clark et al. [71]. The two phases are *query generation* and *relevance assessment*.

Query Generation. In Phase I, annotators are shown "prompts" (distinct from LLM prompting), comprising the first 100 words of randomly selected Wikipedia articles that provide contexts to elicit queries. Following Clark et al. [71], the prompts are designed to help annotators write queries for which they "seek an answer." Annotators are allowed to ask queries potentially requiring retrieval of multiple passages to answer them.

To generate high-quality queries, we ask annotators to avoid generating queries that are directly answerable by the prompts themselves. They are asked to generate well-formed natural language queries likely (in their opinion) to have precise, unambiguous answers. Using prompts also alleviates the issue that queries may be overly biased toward their

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

personal experiences. We also give annotators the option of skipping any prompt that does not "inspire" any queries. This corresponds to step ① in Figure 4.1.

Relevance Assessment. In the second phase, for each query from the previous phase, we ask annotators to judge the binary relevance of the top-k candidate passages ($k = 10$) from an ensemble retrieval system that combines three separate models, which corresponds to step ② in Figure 4.1: (1) **BM25** [281]: We use the implementation in Anserini [383], which is based on the open-source Lucene search library, with default parameters and the corresponding language-specific analyzer in Lucene if it exists. If not, we simply use the white space tokenizer. (2) **mDPR** [156, 397]: This model is trained using the Tevatron toolkit [111] starting from an mBERT checkpoint and then fine-tuned using the training set of MS MARCO Passage. (3) **mColBERT** [159, 41]: This model is trained using the authors' official repository.[2] Same as mDPR, the model is initialized from mBERT and fine-tuned on MS MARCO Passage.

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

We rectify the inconsistent passage segmentation in Mr. TyDi by re-segmenting the Wikipedia articles of TyDi QA. 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 augment the set of passages to be assessed with "projected" relevant passages. To accomplish this, we use 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 in content, the top retrieved passages are likely to have substantial overlap with the annotated passages in Mr. TyDi (and hence are also likely to be relevant). This corresponds to Step ③ in Figure 4.1.

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 corresponds to Step ④ in Figure 4.1. 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 is only labeled by one annotator, which we believe is a better use of limited resources, compared to the alternative of multiple judgments over

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

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 follow the same workflow. After the annotation process concludes, we divide MIRACL into training sets, development sets, test-A sets, and test-B sets, as described in Section 4.1. 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 from the newly generated queries. For the new (known) languages, all generated queries are 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.

4.3 Quality Control

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

4.3.1 Automatic Verification

Automatic heuristic verification is applied to both phases daily, using language-specific heuristics to flag "obviously bad" annotations. In Phase I, we flag the generated query if it is: (1) left empty; (2) similar to previous queries in the same language;[3] (3) missing interrogative indicators;[4] (4) overly short or long.[5] In Phase II, we flagged the label if it is 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) are removed from the dataset, while minor issues are flagged but the data is retained. For both

[3]We measure similarity using Levenshtein distance. A query is flagged if its similarity score to any previous query is greater than 75%.

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

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

cases, whenever the metrics drop below a reasonable threshold,[6] we schedule a meeting with the annotator in question to discuss the issues. This process allows us to minimize the number of obvious errors while giving and receiving constructive feedback.

4.3.2 Manual Assessment

Manual assessment is also applied to both phases. To accomplish this, we hire another group of native speakers (graduate-level students at the University of Waterloo) of each language as reviewers (with minor overlap). Similarly, we provide consistent onboarding training to each reviewer and compensated $18.50 USD per hour.

Phase I. In this phase, reviewers are given both the prompts and the generated queries, and asked to apply a checklist to determine whether the queries meet our requirements. The criteria include the examination of the query itself (e.g., spelling or syntax errors, fluency, etc.) and whether the query can be answered directly by the prompt, which we wish to avoid (see Section 4.2.2).

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

How related are the queries to the prompts? We measure 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 Clark et al. [71]. Overlaps primarily occur in entities or stopwords. We thus conclude that the generated queries are reasonably different from the given prompts.

Phase II. In this phase, reviewers are provided the same guidance as annotators performing the relevance assessment. They are asked to (independently) label a randomly sampled 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 [348], real-world annotators disagree due to a number of "natural" reasons. To further understand why, we randomly sample five query–passage pairs per language for closer examination. After some iterative sense-making, we arrive at the conclusion that

[6] We set this threshold to 0.8 in terms of the percentage of questionable annotations in each submission.

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 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) A partially relevant query–passage pair in MIRACL.

Figure 4.2: Examples of disagreements between annotators and reviewers.

disagreements come from both mislabeling and partially relevant pairs. We show examples of the two cases in Figure 4.2a and Figure 4.2b.

Figure 4.2a 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 4.2b, 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 convince us that the generated queries and annotations are of high quality.

Answer to RQ1: *By hiring multiple native speakers as annotators and employing an intensive two-phase annotation process, coupled with automatic verficiation and quality checks, we construct MIRACL, as a reliable and high-quality monolingual retrieval dataset.*

4.4 MIRACL Analysis

4.4.1 Question Words

We first analyze question types, following Clark et al. [71]. Table 4.3 shows the distribution of the question word in the English dataset of MIRACL, along with SQuAD [269] as a point of reference. The MIRACL statistics are grouped by splits in Table 4.3. The test-B column shows the distribution of the new queries generated with the workflow described in Section 4.2.2, whereas the other columns correspond to Mr. TyDi queries.

Category	Train	Dev	Test-A	Test-B	SQuAD
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 4.3: Query distribution of each split in English MIRACL, compared to SQuAD.

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.

4.4.2 Queries with No Relevant Passages

Recall that since the query generation and relevance assessment phases are decoupled, for some fraction of queries in each language, annotators are not able to identify any relevant passage in the pairs provided to them. As the annotator uses the "prompt" as an inspiration to generate a question in step ①, they do not have a constraint during generation, i.e., the question could likely not be answerable with information present in Wikipedia. Because we aim to have at least one relevant passage per query, these queries—referred to as "invalid" for convenience—are discarded from the final dataset.

For each language, we randomly sample five invalid queries and spot-check them using a variety of tools, including interactive searching on the web. From this, we identify 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)

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 4.4: Baseline results on the MIRACL dev set, where "K. Avg." and "S. Avg" indicate 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)

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 cannot 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. However, we discuss later in Chapter 10, how we utilize these "invalid" queries for relevance assessment when evaluating answer abstention or relevance assessment in multilingual RAG.

4.5 Experimental Results

4.5.1 Baselines

We make several baselines available in the popular Pyserini IR toolkit [193] to serve as foundations that others can build on. Baseline scores for these retrieval models are shown in Table 4.4 in terms of the two official retrieval metrics of MIRACL. The baselines include the three methods used in the ensemble system introduced in Section 4.2.2 (BM25, mDPR, mColBERT) plus the following approaches:

- **Hybrid** combines the scores of BM25 and mDPR results. 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** [138] adopts additional pretraining with contrastive loss based on unsupervised data prepared from CCNet [368], which demonstrates improved effectiveness in downstream IR tasks. We used the authors' released multilingual checkpoint, where the model is further fine-tuned on English MS MARCO after the additional pretraining.
- **In-language fine-tuned mDPR** follows the same model configuration as the mDPR baseline, but we fine-tune each model on the MIRACL training set of the target language rather than on 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 [368], the dataset used by mContriever for additional pretraining. Code, documentation, and instructions for reproducing these baselines are been released together with the MIRACL dataset and can be found on the MIRACL website.

From results in Table 4.4, we observe 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 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`).

Answer to RQ2: *We observe that BM25–mDPR hybrid provides a strong zero-shot baseline that even outperforms in-language fine-tuned mDPR on most languages, except for those that are comparatively under-represented in mBERT pretraining (e.g., `sw`, `th`, `hi`).*

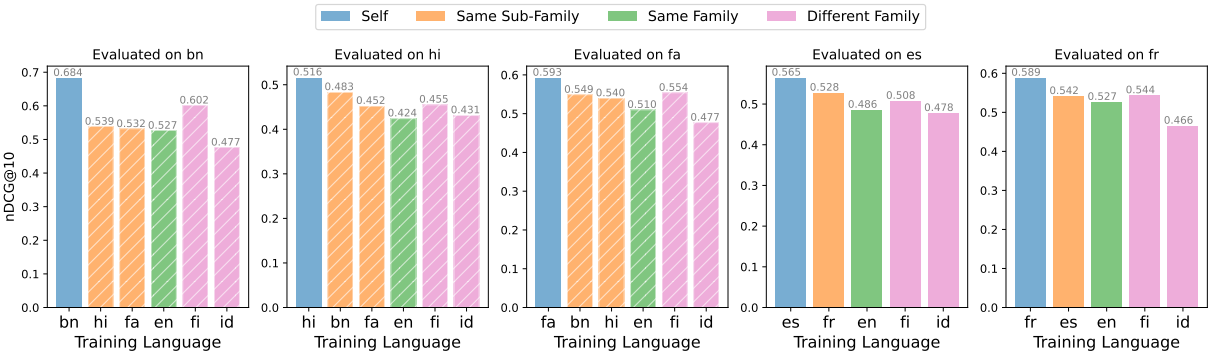


Figure 4.3: Case study of how language kinship affects cross-lingual transfer. Results are grouped according to the target language and each bar indicates a source language (used for fine-tuning mDPR). Within each panel, the relationship between the source and target languages moves from closer to more distant from left to right (self, same language sub-family, same language family, different language family). Striped bars denote that the source language has a different script from the target language. The exact nDCG@10 score is shown at the top of each bar.

4.5.2 Cross-Lingual Transfer Effects

MIRACL enables the exploration of the linguistic factors influencing multilingual transfer; in this section, we present a preliminary study. Specifically, we evaluate cross-lingual transfer effectiveness on two groups of *target* languages: (A) {`bn`, `hi`, `fa`} and (B) {`es`, `fr`}, where the models are trained on different source languages drawn from (with respect to the target language): (1) different language families, (2) same language family but different sub-families, and (3) same sub-family. We add the "self" condition as an upper-bound, where the model is trained on data in the target language itself.

The evaluation groups are divided based on the script of the language: in group (A), all the source languages are in a different script from the target languages, whereas in group (B), all the source languages are in the same script as the target language (Latin script in this case). In these experiments, we reuse checkpoints from the "in-language fine-tuned mDPR" condition in Section 4.5.1. For example, when the source language is `hi` and the target language is `bn`, we encode the `bn` query and the `bn` corpus using the `hi` checkpoint; this corresponds to the `hi` row, **in-L.** column in Table 4.4.

The results are visualized in Figure 4.3, where each panel corresponds to the target language indicated in the header. Each bar within a panel represents a different source language and the y-axis shows the zero-shot nDCG@10 scores. Within each panel, from

left to right, the kinship relationship between the source and target languages moves from close to distant, where the source languages that are at the same distance to the target languages are colored the same (self, same language sub-family, same language family, different language family). Striped bars denote that the source language is in a different script from the target language.

We find that languages from the same sub-families show better transfer capabilities in general, regardless of the underlying script. Among the five languages studied, those from the same sub-families achieve the best transfer scores (i.e., the orange bars are the tallest). The only exception appears to be when evaluating `bn` using `fi` as the source language (the leftmost panel). Interestingly, the source language being in the same family (but a different sub-family) does not appear to have an advantage over the languages in different families (i.e., the green bars versus the pink bars). This suggests that transfer effects only manifest when the languages are closely related within a certain degree of kinship.

We also observe that transfer effects do not appear to be symmetrical between languages. For example, the model trained on `bn` achieves 0.483 and 0.549 on `hi` and `fa`, respectively, which are over 90% of the "self" score on `hi` and `fa` (first orange bar in the second and third panels). However, models trained on `hi` and `fa` do not generalize on `bn` to the same degree, scoring less than 80% of the "self" score on `bn` (two orange bars in the first panel).

Answer to RQ3: *We make two observations about cross-lingual transfer. First, languages from the same sub-families show better transfer capabilities in general. Second, transfer effects do not appear to be symmetrical between languages.*

4.6 Summary

In this chapter, we presented MIRACL, a new high-quality multilingual retrieval dataset that represented approximately five person years of annotation effort. We provided baselines and presented initial explorations that demonstrated the potential of MIRACL for studying interesting scientific questions. While MIRACL represented a significant stride towards equitable information access, further efforts are necessary to accomplish this objective.

One obvious future direction is to extend MIRACL to include more languages, especially low-resource ones. Another possibility is to augment MIRACL with cross-lingual retrieval support. However, pursuing these avenues demands additional expenses and manual labor.

In the next chapter, we will examine a strategy to automatically construct retrieval benchmarks that can be expanded at scale without requiring human annotators.

Chapter 5

Building Realistic Benchmarks on Technical Documents

In the previous chapter, we saw how to conduct post-hoc judgments in argument retrieval and construct a multilingual benchmark by employing multiple human annotators. However, this procedure is rather expensive, cumbersome and time-consuming to manage annotators. In addition, retrieval benchmarks are not well-suited for evaluation, as they are potentially outdated and highly limited. In particular, we observe three major issues in existing retrieval and RAG benchmarks:

- **Lack of realistic, open-ended questions**: Existing datasets contain purely extractive short answers (e.g., Natural Questions [165], TriviaQA [150]) or crowd-sourced questions (e.g., HotPotQA [385]). A limited number of datasets capture "natural" human-asked questions, i.e., MS MARCO [29] or Natural Questions [165], but unfortunately, brief and straightforward questions are inserted into a search box, failing to represent the complex questions that real users might pose to modern-day search systems.

- **Artificially easy**: RAG represents a *solution* for grounded question answering, not a standalone *problem*. Consequently, datasets constructed by design to be solvable via retrieval often fail to encode challenges faced in real-world RAG applications.

- **Static and unspecialized**: After sourcing questions and answers, a benchmark becomes at the risk of (1) *contamination*, if current LLMs are trained on the same set of documents or questions, (2) *overfitting*, when systems inevitably saturate due to repeated internal or external leaderboarding (e.g., BEIR [329]), and (3) *staleness*, when questions or answers are not refreshed and become outdated.

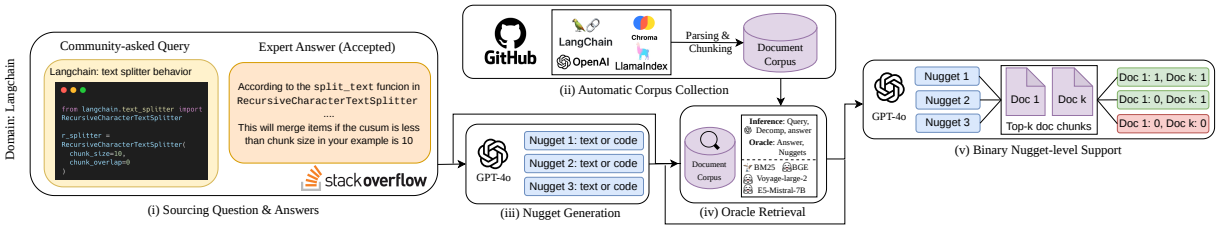


Figure 5.1: The FreshStack framework: (1) Stack Overflow questions and answers are sourced for recent and niche domains. (2) GitHub repository documents are collected and chunked to form the corpus (for each domain). (3) Nuggets or atomic facts within the question and answer are generated with GPT-4o. (4) Ensemble techniques and models retrieve documents, which construct our document judgment pools. (5) GPT-4o evaluates support for every document-nugget pair as a binary judgment.

In this chapter, we introduce **FreshStack**, a holistic framework for constructing realistic datasets in niche and challenging domains. We argue that using technical documentation as the corpus, such as code repositories, which are highly dynamic, provide a particularly suitable foundation for realistic RAG evaluation, seeking to avoid contamination due to (perpetual) recency. Using FreshStack, we construct an evaluation benchmark on five niche domains sourced from community-asked questions and answers on Stack Overflow, together with a corpus containing code snippets and technical documents from public GitHub repositories. We investigate three research questions in this Chapter:

RQ1 *How can we construct challenging evaluation datasets from real user-asked questions?*
RQ2 *How do LLMs perform as assessors for nugget generation on community-asked questions and answers, and for nugget-level support using retrieved documents?*
RQ3 *How do state-of-the-art retrieval models and rerankers perform on retrieval evaluation benchmarks generated with FreshStack?*

5.1 The FreshStack Framework

The framework involves five major stages to construct an evaluation dataset (as highlighted in Figure 5.1). FreshStack includes three major design choices:

1. A general framework that can be extended to different domains without manual effort.
2. Adding recent and niche domains that are actively discussed in computer programming communities such as Stack Overflow.
3. Sourcing community-asked questions and answers to make our evaluation challenging, requiring domain expert knowledge to answer them correctly.

Stack Overflow is an online question answering platform for computer programmers. Users ask questions under particular tags and provide descriptions (often including code snippet with the error message). Questions and answers are also tagged, allowing for easy retrieval of tag or domain-specific questions.

5.1.1 Stack Overflow Domain and Question Selection

For FreshStack, we target niche and recent domains (or topics) only introduced on Stack Overflow from 2023 onward (to reduce pametric knowledge leakage in LLMs), each containing a minimum of 50 posts. We sort all domains by the total number of posts and curate five most-discussed recent domains starting from the highest frequency (LangChain) to the lowest frequency (Yolo v7 & v8), covering different areas and are sufficiently distinct from each other. Each area represents a unique domain, e.g., Computer Vision (CV).

Questions & Answers. We extract relevant posts and answers from the Stack Overflow XML data archive (dated October 2024).[1] We scan the archive to select all relevant posts as questions containing the required tag and further filter answer posts for these questions. Finally, we keep only questions with an *accepted answer*, prioritizing precision over quantity by retaining questions with high-quality answers that are accepted and upvoted by the Stack Overflow community.

5.1.2 Automatic Corpus Collection with GitHub Documents

Answering a higher percentage of questions requires a robust set of corpora drawn from *multiple sources*. For instance, addressing issues in LangChain may require ChromaDB GitHub documentation to resolve errors related to its usage. In our work, we build a separate document corpus per domain by combining multiple repositories as sources.

Stack Overflow Tags. We analyze tag frequencies on Stack Overflow to select relevant GitHub repositories to include in the document corpus. This involves identifying the top-k co-occurring tags, where k is a threshold that balances question coverage with indexing costs. Some tags are generic, such as `Python`, whereas others are specific, such as `LangChainJS`. Filtering tags to keep only a subset of repositories for a domain does not degrade the dataset quality. We manually verify each GitHub repository for each tag, with plans to automate this procedure in the future.

[1]The Stack Overflow XML data archive (CC BY-SA license) is updated once every quarter: https://meta.stackexchange.com/questions/401324/announcing-a-change-to-the-data-dump-process

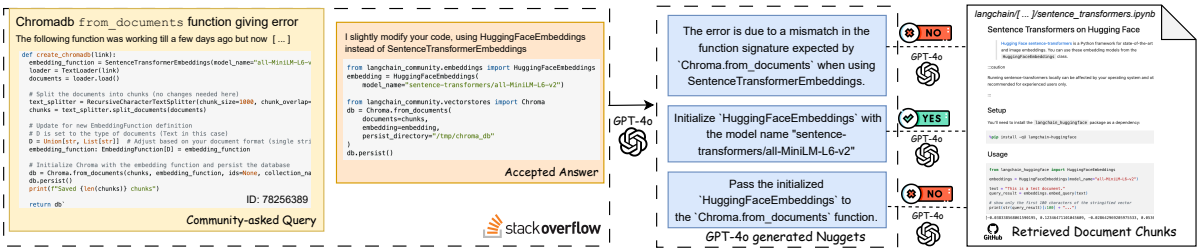


Figure 5.2: A data instance example from LangChain generated with FreshStack.

Chunking & Indexing. We clone the latest branch of the GitHub repository (as of October 22, 2024) and parse files into a tree structure. Each file (either a text document or a code snippet) is chunked into small sections containing at most **2048 tokens** (large enough to contain a large code snippet and fit within the context of dense retriever models), skipping non-text formats.[2] The GitHub filepath serves as the document identifier, with additional chunk details encoded in the identifier. Finally, we combine all chunks into a single corpus, prefixing all document identifiers with the repository name to uniquely distinguish common files (e.g., `langchain/LICENSE`, `langchainjs/LICENSE` and `chroma/LICENSE`).

5.1.3 Nugget Generation with Stack Overflow Question & Answer

A nugget is a core concept or an atomic fact essential in a system's response. The term nugget was informally referred to as SCU (summary content units) as clauses appearing in model summarization [234] and later formalized as "information nugget" for evaluating long-form answers [350, 189, 190, 253]. *Nugget generation* refers to the procedure of constructing nuggets from information-dense text (e.g., Initialize '`HuggingFaceEmbeddings`' with the model name '`sentence-transformers/all-MiniLM-L6-v2`'). The procedure decomposes a verbose answer into key atomic facts or essential components, aiding evaluation. The nugget-generation methodology was first coined two decades ago in the TREC-QA 2003 track for evaluating answers to free-form questions [350]. Back then, human annotators manually wrote "information nuggets" or atomic facts required to be present in the answer. More recently, with the advent of RAG, nugget-based evaluation has renewed interest with LLMs for factual accuracy assessment in long answers, where automatic nugget generation is explored, i.e., using LLMs to automatically create nuggets [18, 218, 103, 258, 259, 260].

Nugget Generation Setting. We automatically generate nuggets from Stack Overflow question–answer pairs using GPT-4o [247], avoiding the cumbersome procedure of manual

[2]We skip indexing images, videos, .bin, .csv, and audio files or unrecognized file formats.

nugget construction [258]. LLM-based nugget generation is explored in the TREC 2024 RAG track [258, 259, 260] and in multiple other works [88, 103]. Separately, we experiment with prompting techniques and find that grading notes style prompts [202] provide parseable and high-quality nuggets in our experiments. An example of nugget generation from a query–answer pair in LangChain is shown in Figure 5.2. Unfortunately, due to legal and copyright issues, we cannot release the exact prompt used for nugget generation.

5.1.4 Retrieval: Oracle & Inference Setting

A RAG evaluation dataset requires questions, answers, and a corpus with documents that can support crucial facts present in the answers. Pooling [407, 46] is a predominant technique in IR for selecting a subset of documents to be assessed for relevance, instead of assessing every document in the collection. We retrieve a list of potentially relevant (unjudged) documents from the corpus and construct judgment pools. Since, we are constructing an *evaluation dataset* and we have *curated answers* for the questions, we retrieve documents using two methods: (1) **Inference**, which relies on the question and automatic approaches, and (2) **Oracle**, which relies on the gold answer or the list of nuggets, to construct judgment pools for the support (or relevance) judgment task in the next stage.

Retrieval Setting. We experiment with multiple systems to increase diversity in our judgment pools. First, we experiment with two techniques in the *inference setting*: (i) GPT-4o Sub-Questions: we decompose the original question and generate synthetic sub-questions with GPT-4o, following Rosset et al. [284], concatenate them, and use the result to retrieve documents (sub-questions are empirically observed to be independent or not chained with each other), and (ii) GPT-4o Closed Book Answer: we generate a closed-book answer for the original question with GPT-4o, similar to HyDE [110], and use the closed-book answer to retrieve documents. Next, in the *oracle setting*, we experiment with: (i) Stack Overflow Answer: we use the curated Stack Overflow answer as the question to retrieve documents, and (ii) Stack Overflow Nuggets: we use the list of GPT-4o-generated nuggets (Section 5.1.3), concatenated as the question, to retrieve documents.

Retrieval Models. We experiment with five different code and text-aware retrieval models: (i) **BM25**, a strong lexical baseline in BEIR [329]. We utilize the default implementation available in Pyserini [193]. (ii) **BGE (Gemma-2)** [59] a dense retriever model[3] fine-tuned on the backbone architecture of Gemma-2 (9B) [280] with an embedding size of 3584 and 8K context length. (iii) **E5 Mistral (7B)** [359] is a dense retriever model[4] based

[3]BGE Gemma-2: https://huggingface.co/BAAI/bge-multilingual-gemma2
[4]E5 Mistral 7B: https://huggingface.co/intfloat/e5-mistral-7b-instruct

Domain	Area	Dataset Count				Avg. Length		% Containing Code		Relevance Judgments	
		#Q	#Docs	#GH	Avg. N/Q	Query	Answer	Query	Answer	Rel. Docs/N	Rel. Docs/Q
LangChain	Machine Learning (ML)	203	49,514	10	3.1	473.4	233.8	83.3%	62.1%	5.7	10.9
Yolo v7 & v8	Computer Vision (CV)	57	27,207	5	3.5	497.1	191.7	70.2%	71.9%	3.9	7.4
Laravel 10 & 11	Backend Development	184	52,351	9	3.0	474.4	155.5	43.5%	51.1%	3.2	6.0
Angular 16, 17 & 18	Front-end Development	129	117,288	4	3.2	463.3	215.1	69.8%	57.4%	4.4	8.7
Godot4	Game Development	99	25,482	6	3.3	350.4	263.0	52.5%	52.5%	2.9	5.9

Table 5.1: FreshStack dataset statistics; Dataset count measures the number of queries, documents, GitHub repositories, and average nuggets per query; Avg. length measures the average text lengths; % containing code measures the fraction of queries and answers with code snippets; Relevance judgments measure relevant documents per nugget and per query.

fine-tuned on the backbone of Mistral 7B [145] with 32 layers and embedding size of 4096. (iv) **Voyage-large-2**[5] is a proprietary and general-purpose embedding model optimized for retrieval quality, with a context length of 16K tokens and embedding size of 1536. (v) **Ensemble Fusion** is defined as the process of combining multiple search techniques (or models) to improve the overall relevance and accuracy of retrieved results [164]. We use a hybrid retrieval strategy that combines all four models (i–iv), normalizes and sums their individual similarity scores, and retrieves up to 100 documents from each model.

5.1.5 Nugget-Level Support Assessment with Retrieved Documents

Traditionally, relevance judgments are conducted on selected pools of retrieved documents, i.e., a human assessor judges the relevance of a question with respect to each provided document. Due to computational costs, recent studies experiment with using an LLM judge (instead of a human assessor) for relevance judgments in IR [97, 331, 340, 267]. Questions in existing IR datasets are traditionally short, making it easier to judge document relevance. In contrast, questions in the FreshStack dataset are long and elaborate (between 350–500 tokens in length), containing a mixture of text, code snippets, or outputs, which makes it challenging to judge question–document relevance directly [84]. For instance, a document may answer the main problem posed in the question, address only part of it, or contain relevant references, and we need to translate these cases into a relevance score.

Nugget-level Support Setting. Instead of relying on traditional relevance assessments, we simplify the judgment procedure for GPT-4o and evaluate whether a document supports the information (or contains) provided by a nugget.[6] Recall that a nugget highlights an

[5]Voyage-large-2: https://docs.voyageai.com/docs/embeddings

[6]This is analogous to relevance judgment in traditional IR, but we use the term "support" to assess whether a document sufficiently supports the nugget information, rather than relevance.

essential fact from the Stack Overflow question or answer. Judging document relevance at the nugget level is effective as nuggets are factual and concise information snippets, which reduces the ambiguity often encountered in traditional relevance judgments. To reduce computational costs, we evaluate top-k documents (a maximum $k = 20$) together with the list of all nuggets for a question in a single inference call ($n + k$). We perform support judgment with GPT-4o as the judge using a chain-of-thought prompt [367]. An example of nugget-level support between nuggets and a retrieved document in LangChain is shown in Figure 5.2. Unfortunately, due to legal and copyright issues, we cannot release the exact prompt used for nugget-level support assessment in our work.

Answer to RQ1: *We can automatically construct realistic benchmarks by (i) scraping real-user Stack Overflow queries with verified answers and GitHub repositories as the corpora, (ii) generating nuggets from the Stack Overflow question and answer, (iii) performing retrieval in the oracle setting to retrieve potential candidate documents and (iv) conducting a nugget-level support assessment of the retrieved candidate documents.*

5.2 Dataset Statistics & Evaluation

After completing the previous stages, we employ two additional post-processing steps to ensure high-quality question–answer pairs remain in the dataset, at the cost of reducing the overall dataset size. In the first step, we remove fully unsupported questions, i.e., questions that do not have even a single relevant document; this removes, on average, **11.8%** of the total questions. In the second step, we apply an aggressive filter by removing questions that contain at least one unsupported nugget, i.e., a nugget that is not supported by any document, which reduces the dataset by an additional **34.2%** of the total questions.

5.2.1 Dataset Statistics

The FreshStack dataset covers five domains for programmers: machine learning, computer vision, backend, front-end, and game development, all listed in Table 5.1. Some Stack Overflow domains such as LangChain are introduced in 2023, whereas others, such as Laravel or Angular, include questions on the latest versions (e.g., Laravel 10 & 11). Each domain contains at least 50 questions, all asked between January 2023 and June 2024. The corpus contains at least 25K documents sourced from 4–10 GitHub repositories. The questions are even longer than answers, containing 350–500 tokens (computed using GPT-4o tokenizer), and at least 50% of the questions and answers contain code snippets. GPT-4o generates around 3–4 nuggets for each question. Each nugget is supported by at least 3 relevant documents, resulting in 5–6 relevant documents per question across all domains.

5.2.2 Cost Comparsion: FreshStack vs. Human Judgments

The cost of automatically constructing FreshStack with GPT-4o is approximately 260-275 USD. Below, we provide a rough estimate of the costs involved: (1) *nugget generation*: $O\big((\alpha_i + \alpha_o) \cdot n\big)$, where n is the number of queries; For all domains it costs about 10–15 USD with GPT-4o to generate nuggets (2–3 USD per domain). (2) *nugget-level support*: $O\big((\beta_i + \beta_o) \cdot n \cdot m\big)$ cost, where m is the depth of retrieved documents for each query; For all domains it costs about 250–260 USD with GPT-4o to assess document support with query nuggets (40–75 USD per domain), where α and β are constants mapping to expected token count, with subscripts i and o corresponding to input and output. Overall, the total computational cost is around 260–275 USD for five domains.

In constrast, manually constructing FreshStack with human annotators is expensive. (1) *nugget generation*: There are 1,149 queries in total, and assuming a human annotator requires 2–3 minutes to generate nuggets per query, requiring 50–60 hours to complete the task. (2) *nugget-level support*: Assuming at least 50 documents need to be labeled for each query, leading to 57,450 pairwise comparisons (query-document pairs). Assuming a single human annotator can annotate 1 pair in a minute (to read all the nuggets and document carefully and mark nuggets that are sufficiently supported by the document), completing all judgments would require roughly 950–1000 hours. Assuming annotators are paid 20 USD per hour, the cost of manually annotating FreshStack is approximately 19–20K USD.

5.2.3 Retrieval Evaluation Metrics

IR evaluation traditionally follows the Cranfield paradigm [347], focusing on individual document relevance, independent of other documents. This paradigm is used to construct standard test collections, such as BEIR [329] and TREC datasets, including the Deep Learning (DL) track [75, 76, 78]. However, diversity in search [51, 298, 371] penalizes information redundancy among retrieved documents to enrich information content and improve efficiency. Therefore, we evaluate retrieval systems with three metrics with relevance judgments at the nugget-level: α-nDCG@10 for diversity and relevance, Coverage@20 for nugget coverage, and Recall@50 for traditional relevancy.

5.3 Document Pooling & Judgment

In this section, we attempt to answer **RQ2** by evaluating methods that retrieve documents contributing to the judgment pools. We first evaluate the retrieval baselines during nugget-

level support judgment (or sampling pools) with both inference and oracle settings. In FreshStack, we are constructing a *test evaluation dataset*. Therefore, we can use the Stack Overflow answer or its variants when constructing judgment pools, as discussed previously in Section 5.1.4. We pool and sample documents from different systems and techniques, similar to how existing question answering datasets are constructed, such as Natural Questions [165] or XOR-TyDI [23], which assess document-level relevance by calculating answer overlap within documents.

Experimental Settings. We perform retrieval with four techniques and baselines (as explained in Section 5.1.4) and with an ensemble fusion of the top 100 documents, where each model's score is normalized and summed up. The evaluation metrics include α-nDCG@10, Coverage@20, and Recall@50. We use GPT-4o with a temperature setting of 0.1[7] for both automatic stages. Nugget generation uses a grading notes prompt with the question and answer, and support assessment uses a chain-of-thought prompt [367], judging up to a maximum of 20 documents simultaneously with the list of nuggets generated for each question. Finally, we sample and judge the top 20 fusion documents from each technique and setting (including the original question) to avoid sampling holes, highlighting the importance of document diversity in our judgment pools.

5.3.1 Document Judgment Pooling Results

We outline the results achieved on document judgment pools during FreshStack construction using techniques from both inference and oracle settings in Table 5.2. Key takeaways and findings are discussed below:

Overall Highlights. Table 5.2 reveals three key findings: (1) Techniques in the oracle setting significantly outperform their counterparts in the inference setting. (2) Stack Overflow answer and nuggets techniques are crucial in pooling potential relevant documents for a question, and (3) fusion outperforms all individual models, highlighting the value of diversity in model choice, aiding in the construction of our judgment pools.

Within the inference setting, we observe that GPT-4o Sub-Questions achieves the best pooling results for four domains (except Yolo v7 & v8), showing that decomposing the question into smaller sub-questions is useful for retrieving relevant documents. Stack Overflow Nuggets achieves the best results (except Laravel 10 & 11) in the oracle setting, showing that breaking down the answer into facts or nuggets is crucial. Among the individual

[7]Separately, we test temperatures of 0.1 and 0.7 and observe an identical downstream retrieval accuracy during FreshStack construction.

Method	Model	LangChain			Yolo v7 & v8			Laravel 10 & 11			Angular 16, 17 & 18			Godot4		
		αN@10	C@20	R@50	αN@10	C@20	R@50	αN@10	C@20	R@50	αN@10	C@20	R@50	αN@10	C@20	R@50
***Inference Setting**: Using a variant of the Stack Overflow question for retrieval of documents within the corpus*																
GPT-4o Sub - Questions	BM25	0.228	0.495	0.249	0.150	0.427	0.328	0.349	0.656	0.464	0.307	0.666	0.378	0.154	0.326	0.211
	BGE (Gemma-2)	0.220	0.561	0.324	0.220	0.554	0.367	0.407	0.727	0.585	0.360	0.707	0.459	0.240	0.532	0.382
	E5 Mistral (7B)	0.262	0.613	0.362	0.266	0.593	0.484	0.306	0.643	0.528	0.305	0.617	0.397	0.220	0.461	0.349
	Voyage-large-2	0.270	0.563	0.329	0.213	0.526	0.370	0.366	0.687	0.552	0.344	0.69	0.449	0.260	0.594	0.473
	Fusion (4 models)	**0.322**	**0.708**	**0.475**	0.305	0.665	0.489	**0.478**	**0.763**	**0.662**	**0.428**	**0.817**	**0.584**	**0.290**	**0.598**	**0.526**
GPT-4o Closed Book Answer	BM25	0.256	0.520	0.273	0.286	0.554	0.431	0.376	0.655	0.495	0.293	0.542	0.332	0.241	0.473	0.349
	BGE (Gemma-2)	0.181	0.467	0.263	0.271	0.599	0.473	0.360	0.694	0.539	0.242	0.525	0.338	0.187	0.454	0.358
	E5 Mistral (7B)	0.198	0.471	0.277	0.239	0.511	0.364	0.188	0.458	0.384	0.179	0.430	0.267	0.151	0.318	0.237
	Voyage-large-2	0.220	0.500	0.301	0.247	0.557	0.495	0.317	0.658	0.524	0.227	0.461	0.338	0.253	0.510	0.454
	Fusion (4 models)	0.275	0.630	0.432	**0.356**	**0.686**	**0.578**	0.420	0.738	0.641	0.290	0.582	0.470	0.288	0.538	0.492
***Oracle Setting**: Using the Stack Overflow answer directly or its variants for retrieval of documents within the corpus*																
Stack Overflow Answer	BM25	0.461	0.726	0.428	0.481	0.756	0.574	0.511	0.774	0.588	0.469	0.751	0.521	0.325	0.565	0.397
	BGE (Gemma-2)	0.290	0.625	0.367	0.390	0.815	0.604	0.472	0.814	0.675	0.346	0.690	0.481	0.341	0.718	0.561
	E5 Mistral (7B)	0.331	0.671	0.430	0.315	0.683	0.509	0.260	0.634	0.488	0.291	0.570	0.412	0.277	0.546	0.434
	Voyage-large-2	0.385	0.700	0.432	0.405	0.703	0.589	0.439	0.791	0.641	0.371	0.680	0.477	0.371	0.626	0.541
	Fusion (4 models)	0.484	0.821	0.619	0.546	0.854	0.788	0.564	**0.892**	**0.820**	0.470	0.805	0.695	0.449	0.741	0.683
Stack Overflow Nuggets	BM25	0.467	0.739	0.445	0.519	0.796	0.657	0.540	0.840	0.654	0.485	0.787	0.536	0.428	0.680	0.489
	BGE (Gemma-2)	0.308	0.667	0.405	0.461	0.784	0.572	0.448	0.806	0.666	0.393	0.756	0.536	0.335	0.664	0.555
	E5 Mistral (7B)	0.323	0.684	0.432	0.437	0.737	0.554	0.287	0.631	0.533	0.346	0.670	0.470	0.292	0.596	0.494
	Voyage-large-2	0.419	0.763	0.508	0.430	0.845	0.675	0.409	0.791	0.624	0.406	0.733	0.533	0.353	0.715	0.590
	Fusion (4 models)	**0.519**	**0.881**	**0.655**	**0.601**	**0.876**	**0.825**	**0.566**	0.888	0.818	**0.544**	**0.881**	**0.756**	**0.476**	**0.814**	**0.719**

Table 5.2: Pooling results by retrieval baselines (including fusion) in inference or oracle settings during FreshStack construction. α-N@10 denotes α-nDCG@10, C@20 denotes Coverage@20 and R@50 denotes Recall@50. Stack Overflow Answer & Nuggets both rely on the gold answer for retrieval (oracle setting), whereas other methods do not rely on the gold answer for retrieval (inference setting). The best result is highlighted in **bold**.

models, BM25 achieves the best α-nDCG@10 on all domains, underlining the importance of lexical approaches in judgment pool construction.

5.4 Qualitative Analysis

In our work, a crucial component is the automatic construction of nuggets and nugget-level document judgments with GPT-4o. To assess GPT-4o's accuracy, we calibrate against an expert human assessor (PhD-level graduate student; the author of this thesis) on a subset of LangChain, evaluating the quality of generated nuggets and nugget-document support labels for 60 randomly sampled questions.

5.4.1 Nugget Quality Evaluation

For nugget quality evaluation, we ask the human assessor to answer the following questions (A, B, and C) after reading the Stack Overflow question, answer, and the list of nuggets.

Nugget Quality		Judgment Quality	
Precision	90.1 %	Relevant	71.7 %
Recall	96.6 %	Partially Relevant	11.7 %
Groundedness	96.4 %	Non-Relevant	16.6 %

Table 5.3: Expert human evaluation by the author of this thesis for nugget quality and nugget-document relevance judgments on 60 randomly sampled queries in LangChain.

(1) A: Does the nugget contain hallucinated content? (requires a boolean response). (2) B: Is the information provided in the nugget minor or redundant? (also requires a boolean response). After completing A and B, we ask (3) C: How many additional nuggets are required to cover all key ideas? (requires an integer response). The cognitive burden to estimate A, B, and C varies across components. Judgments for A and B are binary. In contrast, judgments for C require a holistic assessment of the Stack Overflow answer.

Evaluation Metrics. We measure the nugget quality using three metrics covering different aspects: (1) *precision* (P), which measures whether nuggets generated are accurate, (2) *Recall* (R), which measures whether the nuggets cover the key aspects of the answer; and (3) *Groundedness* (G), which measures whether nuggets contain non-hallucinated content, i.e., stay within the scope of the answer. More formally, we define each metric as follows:

$$\mathbf{P} = \frac{|Nuggets| - \text{sum}(B)}{|Nuggets|}, \quad \mathbf{R} = \frac{|Nuggets| - \text{sum}(B)}{|Nuggets| - \text{sum}(B) + C}, \quad \mathbf{G} = \frac{|Nuggets| - \text{sum}(A)}{|Nuggets|},$$

where $|Nuggets|$ denotes the count of nuggets for a given question.

Experimental Results. As shown in Table 5.3, the generated nuggets achieve above 90% in precision and 96% in recall and groundedness, indicating that GPT-4o can generate high-quality nuggets required by the FreshStack framework. Most nuggets are well-grounded, i.e., they do not produce hallucinated content (3.6% error), and cover the key aspects of the answer in terms of recall (3.4% error). Precision errors are higher (9.9% error), indicating that nuggets may contain either minor or repeated information. Within these errors, the last positioned nugget is not informative in almost 50% of all error cases, and either the first or second positioned nugget in the remaining error cases.

5.4.2 Relevance Judgment Quality Evaluation

We assess relevance between nuggets and documents in nugget-level support. Since judging all documents (including negatives) for each nugget is cumbersome, we qualitatively check

Model	LangChain			Yolo v7 & v8			Laravel 10 & 11			Angular 16, 17 & 18			Godot4		
	αN@10	C@20	R@50	αN@10	C@20	R@50	αN@10	C@20	R@50	αN@10	C@20	R@50	αN@10	C@20	R@50
Inference Setting: Retrieving documents using only the Stack Overflow (SO) query.															
BM25	0.230	0.475	0.261	0.137	0.342	0.337	0.319	0.602	0.441	0.259	0.551	0.340	0.144	0.268	0.200
BM25 + Reranker	0.322	0.587	0.294	0.337	0.590	0.424	0.414	0.729	0.509	0.346	0.647	0.385	0.251	0.407	0.244
BGE (Gemma-2)	0.216	0.548	0.337	0.258	0.547	0.430	0.348	0.699	0.574	0.323	0.571	0.378	0.199	0.479	0.419
BGE (Gemma-2) + Reranker	0.349	0.662	0.387	0.388	0.666	0.459	0.306	0.646	0.571	0.296	0.595	0.387	0.324	0.576	0.471
E5 Mistral (7B)	0.304	0.654	0.393	0.243	0.552	0.394	0.250	0.565	0.470	0.262	0.548	0.368	0.217	0.444	0.359
E5 Mistral (7B) + Reranker	0.385	0.701	0.439	0.364	0.628	0.468	0.305	0.613	0.510	0.306	0.601	0.375	0.315	0.566	0.426
Voyage-large-2	0.246	0.528	0.309	0.270	0.570	0.453	0.345	0.701	0.543	0.304	0.625	0.427	0.282	0.522	0.458
Voyage-large-2 + Reranker	0.345	0.648	0.355	**0.418**	0.670	0.514	0.302	0.653	0.529	0.300	0.600	0.414	**0.342**	0.598	0.511
Fusion (4 models)	0.337	0.700	0.477	0.304	0.627	0.534	**0.426**	**0.748**	**0.646**	**0.385**	**0.719**	**0.532**	0.265	0.550	0.505
Fusion (4 models) + Reranker	**0.397**	**0.729**	**0.501**	0.416	**0.733**	**0.592**	0.319	0.671	0.614	0.318	0.641	0.488	0.340	**0.627**	**0.545**
Best Scores in the Oracle Setting taken from Table 5.2: Upper Baselines on the FreshStack dataset															
SO Answer: Fusion (4 models)	0.484	0.821	0.619	0.546	0.854	0.788	0.564	0.892	0.820	0.470	0.805	0.695	0.449	0.741	0.683
SO Nuggets: Fusion (4 models)	0.519	0.881	0.655	0.601	0.876	0.825	0.566	0.888	0.818	0.544	0.881	0.756	0.476	0.814	0.719

Table 5.4: Document retrieval results on FreshStack with retrieval and reranker baselines (including fusion). Best scores or upper baselines in the oracle setting are taken from Table 5.2. The reranker is the Voyage AI rerank-2 model [9] reranking the top 100 documents. If the reranker improves upon the retrieval model, it is highlighted in **green** else **red**. We highlight the best result in **bold**.

precision by evaluating only the relevant pairs. The annotator labels one positive document per question on a three-level scale: relevant, partially relevant, or non-relevant.

Experimental Results. As shown in Table 5.3, 71.7% of the judged nugget–document pairs are relevant, with an additional 11.7% labeled as partially relevant, indicating a high precision in GPT-4o support judgments. On the other hand, GPT-4o makes an incorrect judgment for 16.6% of the total questions. This discrepancy arises from several factors: some documents are relevant to only part of the nugget's information, leading to mislabeling; ambiguity within the nugget content can cause misjudgments; and occasionally, literal grounding of a document in the nugget does not translate to semantic relevance for answering the question.

Answer to RQ2: *Expert evaluation of GPT-4o as the judge indicates that most nuggets are well-grounded and cover the key aspects of the answer. Similarly, GPT-4o shows high precision in nugget-level support judgments, although the remains room for improvement with respect to ambuiguity in nugget content and cases of literal grounding.*

5.5 Main Experiments

In this section, we evaluate retrievers and rerankers in document retrieval on the constructed FreshStack datasets, addressing **RQ3** posed in our introduction. All models are evaluated

using only the Stack Overflow question to retrieve documents in the inference setting, and do not include any information about the Stack Overflow answer or nuggets, ensuring a fair assessment.

Experimental Settings. We evaluate the same retrieval models used as baselines during pooling in FreshStack: BM25, BGE (Gemma-2), E5 Mistral 7B, Voyage-large-2, and Fusion. In addition, we evaluate the Voyage AI rerank-2 [9] as the reranker with a 16K context length, reranking the top 100 documents retrieved from each retrieval system and fusion. Evaluation metrics are defined in Section 5.2.3: α-nDCG@10, Coverage@20, and Recall@50.

5.5.1 Document Retrieval Results

Accuracy gap between oracle indicates plenty of headroom. From Table 5.4, we observe techniques from the oracle setting (using Stack Overflow answers or nuggets) achieve a substantially higher α-nDCG@10, Coverage@20, and Recall@50 compared to all inference setting models, including ensemble fusion and reranking with VoyageAI rerank-2. This highlights the complexity of answering FreshStack questions and demonstrates substantial headroom for improving existing code-mixed retrieval models to close the gap between retrieval models at inference and oracle approaches.

Ensemble fusion outperforms individual models. Individual retrieval models demonstrate limited success on the FreshStack dataset; in contrast, the ensemble fusion of four retrieval models outperforms each individual model across all metrics (α-nDCG@10, Coverage@20, and Recall@50) and all five domains, except α-nDCG@10 on Godot4. This highlights a crucial point: a compound retrieval system [391], developed as an ensemble of retrieval models or something similar, is currently required to retrieve documents for niche and challenging domains. However, fusion is inefficient at inference time, as it aggregates multiple model inferences, requiring more efficient alternatives.

Opportunities to improve reranking. When using a weak first-stage retriever, neural rerankers typically improve document ranking [329], although it has been recently shown that this is not always the case when a strong first-stage retriever is used [392, 140]. Consistent with these recent observations, reranking provides benefits over BM25 for all domains in the FreshStack dataset. However, for dense retrievers, reranking provides a clear benefit on some but not all datasets. Specifically, while the reranker enhances α-nDCG@10, Coverage@20, and Recall@50 for LangChain, Yolo v7 & v8, and Godot4, it reduces these metrics on Laravel 10 & 11 and Angular 16, 17 & 18 for BGE (Gemma-2), Voyage-large-2 and fusion. We suspect the reranker performs better for certain programming languages such as Python, and we leave a deeper analysis of the limitations to future work [140].

Answer to RQ3: *Hybrid fusion of BM25 and dense retrievers outperforms each individual retrieval system on FreshStack. In addition, rerankers provides clear benefits in some domains but not in all. Finally, the accuracy gap between oracle and inference techniques indicates substantial headroom for improvement in FreshStack.*

5.5.2 FreshStack Discussion

Generalization Assumption. While our approach to creating FreshStack can, in principle, be generally applied to any domain, in this work we apply it to questions on technical documents. The key assumptions for FreshStack to generalize to a new domain are: (1) requires at least 50 queries (50 is the typical minimum number of queries used in TREC, e.g., the TREC Deep Learning (DL) 2020 Track used 54 queries [77]). that are human-generated for a lower variance in retrieval performance; (2) answers to the queries (preferably human-generated) that can be linked using information available in documents; and (3) a collection of at least 5–10K documents (or chunks) with rich information.

Data Contamination. Fully eliminating data contamination is infeasible, as both Stack Overflow and GitHub are publicly available. However, by focusing on temporally recent and niche domains and carefully considering the knowledge cutoff dates of LLMs, we can potentially reduce the likelihood of data contamination during LLM pre-training and supervised fine-tuning.

RAG results. In this chapter, we have omitted out the RAG results, which are present in the original publication. Using the GPT-4o generated nuggets from Stack Overflow queries and gold answers, we evaluate the naively generated LLM responses by feeding the query and top 20 retrieved oracle or retrieved documents as context. We evaluate using the nugget recall metric (A_{strict}), introduced in Pradeep et al. [260].

5.6 Summary

In this chapter, we introduced a holistic framework, FreshStack, to automatically construct challenging datasets for realistic evaluation of retrieval systems. We sourced real user questions and answers from Stack Overflow and built a document corpus using technical documents from public GitHub repositories. Using FreshStack, we constructed datasets in five niche domains and evaluated four frontier retrieval models and a reranker model in the document retrieval setting. The accuracy gap observed between the retrieval models and the oracle approaches indicated substantial headroom for improvement, and we identified cases that may motivate future research in reranking.

Chapter 6

Scaling Multilingual Retrieval with Synthetic Training Data

Until now, we explored the construction of retrieval benchmarks (for argument retrieval, multilingual settings, and technical domains). To improve accuracy on retrieval benchmarks, a crucial component lies in the training data. Typically, a combination of supervised and synthetic training data plays a key role during fine-tuning of SoTA retrievers, such as Qwen-3 embeddings [399]. In this chapter, we will learn an effective technique for large-scale and high-quality synthetic training data generation for multilingual retrieval (terminology defined in Section 2.1 back in Chapter 2).

6.1 Overview

There is limited exploration of multilingual retrieval models, due to uneven and sparse distribution of human-supervised training data for languages other than English [276, 104, 369]. Collecting human annotations for training data generation is not scalable, as it is cumbersome to search and hire native speakers, verify their language proficiency, and train them. Additionally, human annotators are expensive, requiring a large annotation budget to obtain a sufficient amount of training pairs (as seen previously in Chapter 4).

In this chapter, we use PaLM 2 [17], a recent multilingual LLM (successor of PaLM 540B [69]) for query generation. The generated query, paired with the original passage from Wikipedia, is used to construct the **SWIM-IR** dataset, containing both monolingual and cross-lingual retrieval training pairs. It provides synthetic training (query–passage) pairs

Dataset	Q Gen.	Cross.	Mono.	# L	Domain	# Train
NeuCLIR	Human	EN→L	L→L	3	News (hc4)	×
MKQA	Human	L→EN	×	26	Wikipedia	10K
mMARCO	Translate	×	L→L	13	MS MARCO	533K
Mr.TyDI	Human	×	L→L	11	Wikipedia	49K
MIRACL	Human	×	L→L	18	Wikipedia	726K
JH-POLO	GPT-3	EN→L	×	3	News (hc4)	78K
SWIM-IR	**PaLM 2**	**L→EN**	**L→L**	**33**	**Wikipedia**	**28M**

Table 6.1: We construct SWIM-IR, a "synthetic" multilingual dataset with 28 million PaLM 2 generated training pairs across 33 languages in our work; (Q Gen.) denotes the query generation technique; (Cross. and Mono.) denotes the retrieval task and (query→document) language pair; (# L and # Train) denotes the language count and available training pairs.

for improving dense retrieval models without requiring any human supervision. The dataset spans 33 diverse languages, including both high and very-low resource languages and is one of the largest multilingual (i.e., monolingual and cross-lingual) synthetic training dataset with 28 million training pairs (cf. Table 6.1). We ask the following three research questions:

RQ1 *Can we construct a large multilingual retrieval dataset automatically and at scale?*
RQ2 *How do unsupervised multilingual retrievers perform against supervised counterparts?*
RQ3 *What is the cost of creating such a large-scale dataset automatically, compared to human judgments?*

6.2 SWIM-IR Dataset Overview

In our dataset overview, we first describe the summarize-then-ask prompting technique for multilingual query generation (Section 6.2.1), data construction details (Section 6.2.2), and finally discuss human validation and content filtration (Section 6.2.3).

6.2.1 Summarize-then-Ask Prompting (SAP) Design Formulation

Multilingual query generation is not a trivial task as it requires a deep understanding of the passage content and its own translations across different languages [356]. In addition, passages are often lengthy and contain information on multiple topics. Using the entire passage can potentially cause hallucinations in models by generating non-meaningful queries, which negatively affects retrieval performance [117].

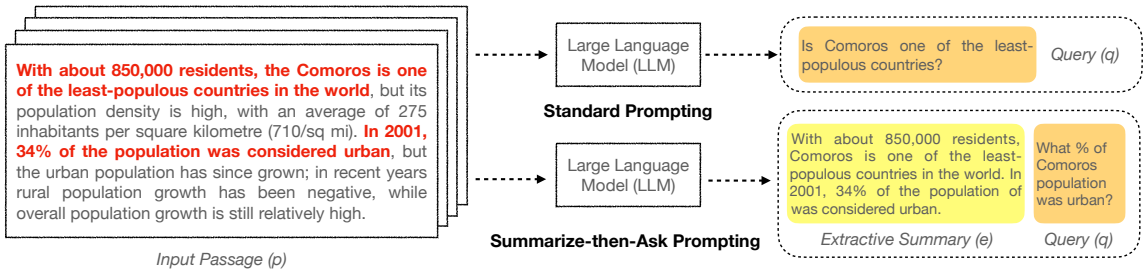


Figure 6.1: An illustration of SAP (*Summarize-then-Ask Prompting*) versus standard prompting for English query generation on English Wikipedia. SAP assists the LLM in improving the query generation quality (orange box) by identifying the relevant sections of the input passage (highlighted in red) via the extractive summarization (yellow box) as an intermediate reasoning step.

To reduce the complexity of multilingual query generation and improve query quality, we implement summarize-then-ask prompting (SAP). As shown in Figure 10.1, we identify the relevant information within a passage by asking the LLM to generate an extractive summary and use it as an intermediate step for generating informative queries [367]. The procedure is described in more detail below:

(i) Summary Extraction. The LLM constructs an extractive summary e_s of the input passage p_s, where s denotes the source language. The summary captures the most relevant information contained within the passage p_s (which occasionally may be long) and acts as a useful intermediate signal for the LLM to generate a multilingual query in the later stage. We denote the first stage as $e_s = \text{LLM}(p_s; \theta^1, \cdots, \theta^k)$, where $(\theta^1, \cdots, \theta^k)$ denotes the k few-shot prompt exemplars[1] containing the passage, the summary in the source language s and the query in the target language t. Given a few-shot exemplars demonstrating the task, we empirically observe that PaLM 2 can extract summaries, rather than generating them.[2]

(ii) Query Generation. Next, the LLM combines the summary e_s generated previously with the original input passage p_s, highlighting the relevant information for composing the query q_t in the target language t. We denote this stage as $q_t = \text{LLM}(e_s, p_s; \theta^1, \cdots, \theta^k)$, where the extractive summary e_s, the input passage p_s and the k-shot exemplars from the first stage all appear as input.

[1]Multilingual query generation requires few-shot prompt exemplars. We internally observe that zero-shot prompting often generates unparseable outputs with PaLM 2.

[2]In our work, we do not use abstractive summarization, as LLMs have notoriously known to hallucinate and generate factual inconsistencies in their outputs [219, 199].

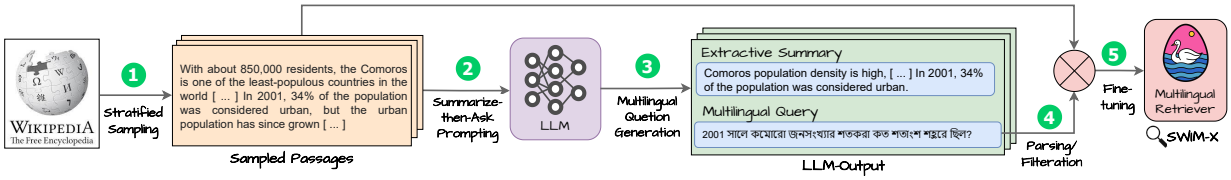


Figure 6.2: An illustration of the cross-lingual SWIM-IR dataset construction procedure.

6.2.2 SWIM-IR Dataset Construction

To construct SWIM-IR, we require only an unlabeled corpus of passages and a set of few-shot exemplars. An overview of the cross-lingual generation procedure is shown in Figure 6.2.

Cross-lingual. The goal is to generate a query in the target language t using an input passage in English (the source language s). We use a stratified sampling algorithm to sample up to 1 million passages for each target language t from the English Wikipedia corpus used in XOR-Retrieve [71, 23] or XTREME-UP [288]. Next, we construct five prompt exemplars and manually create both the summary and query for each exemplar in English. Further, we use Google Translate to translate the exemplar queries into other languages. Finally, we construct the prompt, by explaining the query generation task as an instruction, specifying the target language, and including the 5-shot exemplars as input to the LLM with SAP.

Monolingual. The goal is to generate a query in the same language as the input passage ($s = t$). We follow the setting similar to the cross-lingual task. We first sample up to 1 million passages (if available) from each language-specific Wikipedia corpus in MIRACL [398].[3] Next, we carefully select three training pairs as our prompt exemplars.[4] For languages with no training split, we manually construct the prompt exemplars. Further, we then use Google Bard to generate exemplar summaries in the target language. Finally, we construct the prompt by describing the query generation task and including the 5-shot exemplars with SAP.

6.2.3 Human Validation & Content Filtration

Human Validation. The goal of our query generation is to generate an adequate and fluent query given a passage [262]. To evaluate the intrinsic query quality, we conduct a

[3]MIRACL contains a training split except for two languages: German (`de`) and Yoruba (`yo`).

[4]As language-specific passages consume more tokens (e.g., Telugu), to save computational budget, we use only on 3-shot exemplars (instead of five) for the monolingual task.

human validation study on SWIM-IR using a subset of five languages, where each author in the paper is a native speaker of an individual language for this human validation study. Among these languages, three are high-resource (English, Spanish and Chinese), one medium-resource (Hindi) and one low-resource (Bengali). For each language, we randomly sample a fixed amount of query-passage pairs resulting in a total of 500 evaluation pairs that are human-validated across all languages.

We compute the query quality on a three-level rating scheme (0/1/2) based on three evaluation criteria: fluency, adequacy, and language. (i) *Fluency* measures the coherence of the generated query, i.e., whether the query is understandable and readable by the user and contains no spelling or grammatical mistakes. (ii) *Adequacy* measures the relevancy of the query with passage, used for query generation. (iii) *Language* detects whether the generated query is in the required or specified language prompted by the user.

Validation Statistics. Table 6.2 reports the human validation statistics. For fluency, we observe major mistakes in Hindi (12%) and Chinese (7%), where the sampled passages from MIRACL [398] can be too short (only 2–3 words), which leads to the exact duplication of the passage text in the query. For adequacy, we observe that in Chinese (30%) of the queries are not relevant to the passage. Similar to fluency, low adequacy is observed in cases when either a query is generated for a short passage or when the query addresses a related topic that is not directly referenced in the passage. Finally for language, annotators report between 99–100% for all languages, indicating that PaLM 2 reliably generates queries in the intended target language. Importantly, we observe technical entities remain in English forming code-switched queries (e.g., Bengali), which are natural and expected in our evaluation.

Lang. (ISO)	Fluency (↑)			Adequacy (↑)			Language (↑)		
Rating	0	1	2	0	1	2	0	1	2
English (en)	2%	3%	95%	2%	13%	85%	0%	0%	100%
Spanish (es)	1%	10%	89%	14%	12%	74%	1%	0%	99%
Chinese (zh)	7%	19%	74%	7%	30%	63%	0%	0%	100%
Hindi (hi)	12%	5%	83%	6%	19%	75%	0%	0%	100%
Bengali (bn)	6%	4%	90%	10%	14%	76%	1%	0%	99%

Table 6.2: Human validation statistics on SWIM-IR. Annotators rate fluency, adequacy, and language correctness on a 0/1/2 scale.

Content Filtration. LLMs have been shown to generate undesirable content, particularly under conditions that prime the model with material targeted at drawing out any negative patterns or associations in the training data [114, 32]. To mitigate this, we use the Google Cloud Natural Language content classification categories[5] to filter harmful content from the SWIM-IR training pairs. We discard samples with a high content classification of either `/Adult` or any of the `/Sensitive Subjects` labels.

[5] https://cloud.google.com/natural-language/docs/categories

Model	PLM	PT	Finetune (Datasets)	Success@5kt Avg.	Ar	Bn	Fi	Ja	Ko	Ru	Te
Existing Supervised Baselines (Prior work)											
Dr. DECR [185]	XLM-R	WikiM	NQ + XOR*	73.1	70.2	85.9	69.4	65.1	68.8	68.8	83.2
mDPR [23]	mBERT	—	XOR	50.2	48.9	60.2	59.2	34.9	49.8	43.0	55.5
mBERT + xQG [406]	mBERT	—	XOR	53.5	42.4	54.9	54.1	33.6	52.3	33.8	52.5
Google MT + DPR [23]	BERT	—	NQ	69.6	69.6	82.2	62.4	64.7	68.8	60.8	79.0
OPUS MT + DPR [23]	BERT	—	NQ	50.6	52.4	62.8	61.8	48.1	58.6	37.8	32.4
Zero-shot baselines (English-only supervision)											
mContriever	mT5	mC4	—	38.9	35.9	33.9	43.6	34	35.1	45.1	44.5
mDPR-EN	mT5	—	MS MARCO	39.3	34.3	35.5	45.2	40.2	36.5	43.9	39.5
mContriever-EN	mT5	mC4	MS MARCO	44.0	37.5	38.2	50.6	41.1	37.2	49.8	53.8
Supervised Baselines (Cross-lingual supervision)											
mDPR-X	mT5	—	XOR	53.6	51.5	63.5	52.5	45.6	52.3	43.0	66.8
mContriever-X	mT5	mC4	XOR	55.3	52.1	68.1	54.5	47.7	50.5	50.2	64.3
mDPR-X	mT5	—	MS MARCO + XOR	58.2	55.3	70.1	56.7	49.8	55.8	50.6	69.3
mContriever-X	mT5	mC4	MS MARCO + XOR	59.6	54.7	73.4	57.0	53.1	56.5	51.5	71.0
Synthetic Baselines (Our work)											
SWIM-X (500K)	mT5	—	SWIM-IR	59.0	54.0	67.4	59.2	52.7	55.1	54.4	70.2
SWIM-X (500K)	mT5	mC4	SWIM-IR	63.0	57.0	71.1	61.8	56.8	60.7	63.3	70.2
SWIM-X (7M)	mT5	—	SWIM-IR	65.1	57.9	75.0	65.6	59.3	58.9	64.6	74.4
SWIM-X (7M)	mT5	mC4	SWIM-IR	66.7	61.2	77.0	65.0	62.2	62.8	65.4	73.5

Table 6.3: Experimental results showing Success@5kt for cross-lingual retrieval on XOR-Retrieve dev [23]; (PLM) denotes the pre-trained language model; (PT) denotes the pre-training dataset; (*) Dr.DECR is fine-tuned in a complex training setup across more datasets (§6.3.3); WikiM denotes WikiMatrix [299]; XOR denotes XOR-Retrieve; SWIM-X (ours) is fine-tuned on 500K and 7M synthetic data.

Biases in Synthetic Query Generation. Synthetic query generation, while scalable, introduces known biases: First, LLM-generated queries are often well-formed and less ambiguous than real user queries. Second, since queries are generated conditioned on passages, they may be overly aligned with the source content, potentially leading to artificially easier retrieval scenarios. Third, synthetic queries may underrepresent diverse information needs, as LLMs focus on salient and explicitly stated content. In our work, we view synthetic query generation as a mechanism for scaling training data, rather than replacing human-centered evaluation.

Answer to RQ1: *Utilzing a large multilingual LLM (such as PaLM 2) to generate multilingual queries from Wikipedia passages using SAP prompting is an effective strategy. The generated queries are fluent, adequate and in the required target language.*

Model	Avg.	ar	bn	en	es	fa	fi	fr	hi	id	ja	ko	ru	sw	te	th	zh	de	yo
Existing Supervised Baselines (Prior work)																			
BM25	38.5	48.1	50.8	35.1	31.9	33.3	55.1	18.3	45.8	44.9	36.9	41.9	33.4	38.3	49.4	48.4	18.0	22.6	40.6
mDPR	41.8	49.9	44.3	39.4	47.8	48.0	47.2	43.5	38.3	27.2	43.9	41.9	40.7	29.9	35.6	35.8	51.2	49.0	39.6
Hybrid	56.6	67.3	65.4	54.9	64.1	59.4	67.2	52.3	61.6	44.3	57.6	60.9	53.2	44.6	60.2	59.9	52.6	56.5	37.4
Cohere-API	54.2	66.7	63.4	50.1	50.7	48.4	67.5	44.3	57.3	50.5	51.6	54.6	47.7	54.3	63.8	60.6	38.9	41.4	62.9
Zero-shot baselines (English-only supervision)																			
mDPR-EN	39.8	49.7	50.1	35.4	35.3	39.3	48.2	31.3	37.4	35.6	38.9	44.1	36.1	33.8	49.2	50.6	34.7	32.1	34.4
mContriever-EN	37.8	49.1	48.4	32.7	33.3	37.1	48.4	27.0	35.9	32.7	34.1	40.2	35.1	44.5	46.2	45.0	27.5	29.7	33.7
Supervised Baselines (Monolingual supervision)																			
mDPR-X	39.6	52.8	57.1	30.2	24.7	37.6	46.1	26.4	27.8	37.3	42.9	38.3	34.9	53.7	68.4	58.2	34.9	19.2	22.2
mContriever-X	55.4	66.4	68.4	44.2	42.8	48.9	65.2	46.2	45.0	45.8	56.8	58.8	51.2	67.7	79.0	70.7	49.4	42.3	48.4
Synthetic Baselines (Our work)																			
SWIM-X (180K)	46.4	60.2	57.1	34.7	33.4	36.3	40.6	64.3	33.0	39.5	40.8	43.3	49.7	40.0	55.9	56.3	63.3	50.2	36.5

Table 6.4: Experimental results for monolingual retrieval on MIRACL dev [398]. All scores denote **nDCG@10**; (Hyb.) denotes Hybrid retriever with ranked fusion of three retrievers: mDPR, mColBERT and BM25; BM25, mDPR and Hybrid scores taken from [398]; Cohere-API is used as a reranker on top of 100 BM25 results, taken from [153]. SWIM-X (ours) is fine-tuned on 180K synthetic training pairs.

6.3 Experiments

6.3.1 Datasets and Metrics

We evaluate on three multilingual retrieval benchmarks: (i) **XOR-Retrieve** [23], (ii) **MIRACL** [398] and (iii) **XTREME-UP** [288]. XOR-Retrieve and XTREME-UP are cross-lingual, while MIRACL is monolingual. Following prior work, we evaluate models at Success@5kt on XOR-Retrieve, nDCG@10 on MIRACL and MRR@10 on XTREME-UP.

6.3.2 Experimental Methods

Baseline Categories. Following common practice across all datasets, we evaluate three categories of baselines: (i) *Zero-shot baselines*: where the model denoted by "EN" (model-EN) is fine-tuned using supervised English-only training data such as MS MARCO [29] or NQ [165]. (ii) *Supervised baselines*: where the model denoted by "X" (model-X) is fine-tuned on human-supervised, i.e., multilingual training data. (iii) *Synthetic baselines*: where the model denoted by "SWIM-X" is fine-tuned without any supervision, relying purely on synthetic multilingual training data. Additionally, we report the amount of synthetic pairs used, e.g., SWIM-X (500K) is fine-tuned on 500K training pairs.

Model Choices. For our dense retrieval models, we adapt DPR [156] to the multilingual setting with the mT5-base [379] language model with 580M parameters. Next, we include

mContriever [138], which adopts an additional pre-training stage with contrastive loss based on unsupervised data prepared from pairwise sentence cropping in mC4 [379]. For query generation, we use PaLM 2 (S) [17] for efficient generation due to its low-cost and latency.

Existing Baselines. For XOR-Retrieve, we include Dr. DECR [185], a cross-lingual ColBERT [159] fine-tuned on a large amount of supervised data in a computationally expensive setup involving knowledge distillation with English ColBERTv2 [296]. xQG [406] performs cross-language query generation and concatenates the queries with the passage representation. We also include two-stage translation baselines, Google Translate and Opus-MT, from Asai et al. [23]. For MIRACL, we include the official BM25, mDPR and Hybrid (combining BM25, mDPR and mColBERT) baselines [398], and we use the Cohere-API as a reranker on the top-100 BM25 results [153].

6.3.3 Training Methodology

Zero-shot & Supervised Baselines. We replicate the mContriever and mDPR zero-shot baselines by initializing from an mT5-base checkpoint [379] and further fine-tuning on MS MARCO, following a setup similar to Ni et al. [236]. Similarly, mContriever-X and mDPR-X are additionally fine-tuned on training split available for each dataset. As mContriever includes an additional pre-training stage, we set the batch size to 8192, learning rate to $1e^{-3}$ and pre-train for 600K steps on mC4 [379].

Synthetic Baselines. For SWIM-X, we pre-train the mT5-base checkpoint on mC4 [379] for 600K steps using a contrastive loss function objective, similar to Contriever [138]. Next, we fine-tune the pre-trained mT5-base model on SWIM-IR with in-batch negatives and a contrastive loss function [343]. During fine-tuning, we set the batch size to 4096, learning rate to $1e^{-3}$ and fine-tune for 5–50K training steps, depending on the training dataset size.

6.3.4 Experimental Results

XOR-Retrieve. Table 6.3 shows that SWIM-X (7M), fine-tuned on 7M synthetic pairs (max. of 1M per language) outperforms the best supervised baseline, mContriever-X, by 7.1 points Success@5kt. Without mC4 pre-training, the accuracy of SWIM-X (7M) drops by only 1.6 points. We also evaluate SWIM-X (500K), a limited-budget baseline fine-tuned on 500K training pairs, which outperforms mContriever-X by 3.6 points. Few existing baselines outperform SWIM-X, however, the comparison is not fair. For instance, Dr. DECR is a multilingual ColBERT [159] model, which is computationally expensive at inference [329].

Model	Avg.	as	bho	brx	gbm	gom	gu	hi	hne	kn	mai	ml	mni	mr	mwr	or	pa	ps	sa	ta	ur
Zero-shot baselines (English-only supervision)																					
mDPR-EN	6.3	2.6	6.4	0.4	7.2	1.3	8.6	13.3	5.2	10.4	6.4	12.3	0.2	8.9	5.8	0.4	6.0	5.6	5.2	10.2	10.0
mContriever-EN	7.9	7.9	3.2	7.8	0.3	9.7	2.2	11.1	15.2	8.2	10.6	8.6	15.6	0.4	10.7	8.5	1.1	10.3	3.3	5.7	12.9
Supervised Baselines (Cross-lingual supervision)																					
mDPR-X	8.4	6.7	9.9	4.8	10.0	8.7	8.8	9.1	9.4	9.0	10.0	10.5	4.8	7.8	9.6	6.9	8.6	7.4	8.5	8.1	9.1
mContriever-X	12.4	9.8	15.7	6.7	14.0	11.7	13.3	15.5	13.9	13.6	13.9	16.9	6.5	12.0	13.8	7.5	13.4	9.8	12.4	13.0	14.1
mContriever-X$^{\heartsuit}$	13.5	11.6	15.4	8.0	16.9	12.3	15.2	16.7	15.7	14.7	15.6	17.4	7.0	14.2	14.7	9.1	13.2	10.1	14.8	12.1	14.9
Synthetic Baselines (Our work)																					
SWIM-X (120K)^{MT}	26.1	25.2	29.5	2.1	30.8	22.1	31.5	35.8	31.5	28.7	32.2	34.6	2.2	32.7	27.7	14.8	30.7	21.0	28.2	30.6	29.2
SWIM-X (120K)	25.2	24.4	27.7	4.3	28.3	25.4	29.4	32.4	28.8	30.1	31.8	34.4	5.1	30.7	25.7	15.8	29.6	20.6	26.1	27.9	26.1

Table 6.5: Experimental results for cross-lingual retrieval on XTREME-UP test [288]. ($^{\heartsuit}$) denotes the mContriever-X model fine-tuned without MS MARCO [29]; Two variants of SWIM-X considered, both fine-tuned on 120K synthetic data: (1) SWIM-X (120K)^{MT} fine-tuned using Google Translate, i.e., translated prompt exemplars for 15 languages, whereas (2) SWIM-X (120K) is fine-tuned using prompt exemplars sampled from XTREME-UP training split for all languages.

Similarly, Google MT + DPR relies on a Google Translate system for the translation of cross-lingual queries to English.

MIRACL. Table 6.4 shows that the SWIM-X (180K) model is competitive on MIRACL. SWIM-X (180K) outperforms the best zero-shot model by 6.6 points nDCG@10. However, SWIM-X underperforms mContriever-X on MIRACL, which is fine-tuned on 90K human-labeled training pairs with up to four hard negatives available in MIRACL, by 9.0 points nDCG@10. This highlights the difficulty in the monolingual retrieval task, where models still benefit substantially from human-supervision. A few existing baselines outperform SWIM-X; however, the comparison is not fair. The Hybrid baseline aggregates information from three models, and for Cohere-API, the underlying model information is unknown.

XTREME-UP. Table 6.5 shows the results on XTREME-UP. SWIM-X (120K) is fine-tuned by randomly selecting 5 exemplars from the XTREME-UP training dataset (human-labeled queries) for all languages, whereas the MT variant reuses XOR-Retrieve prompt exemplars with translated summaries and queries for 15 languages.[6] SWIM-X (120K)^{MT} outperforms the best supervised baseline, mContriever-X$^{\heartsuit}$ (fine-tuned without MS MARCO) by a huge margin of 12.6 points MRR@10, but performs minimally worse than the MT version by 0.9 points. Interestingly, none of the evaluated baselines perform well on two extremely low-resource languages, Boro (`brx`) and Manipuri (`mni`).

[6]We could not translate prompt exemplars for five languages due to lack of support in Google Translate: Boro (`brx`), Garhwali (`gbm`), Chhattisgarhi (`hne`), and Marwari (`mwr`). Manipuri (`mni`) is available only in the Meitei script, whereas XTREME-UP uses the Bengali-Assamese script.

Answer to RQ2: *SWIM-X, an retrieval model fine-tuned only with synthetic training data in SWIM-IR, outperforms supervised baselines by a large margin on both cross-lingual retrieval benchmarks: XOR-Retrieve by 7.1 points at Success@5kt and XTREME-UP by 11.7 points at MRR@10. On MIRACL, SWIM-X underperforms mContriever-X by 9.0 points at nDCG@10, which indicates room for future improvement.*

6.3.5 Effectiveness of Summarization & K-shot Exemplars

Summarization Effectiveness. In our work, we use the SAP prompting technique, in which we employ extractive summarization as a rationale for PaLM 2 to generate informative multilingual queries. To evaluate the effectiveness of summarization, we assess both settings (i.e., contrasting with and without summarization) on cross-lingual retrieval using Success@5kt on XOR-Retrieve. We additionally evaluate different PaLM 2 model sizes to examine a relationship between retrieval model performance and LLM size, (i.e., in terms of model parameters). To ensure consistency, we adopt the experimental setup utilized in SWIM-X (500K) for all models. Our results are shown in Figure 6.3 (left). We infer two insights: (i) Increasing the LLM size yields diminishing returns in terms of Success@5kt on XOR-Retrieve. (ii) SAP outperforms standard prompting by at least 0.6 points consistently with all various PaLM 2 generators on XOR-Retrieve, with a maximum improvement of up to 3.2 points at Success@5kt. We also observe that PaLM 2 models with large sizes (sizes $>$ S) are inherently able to generate coherent queries without requiring SAP, leading to diminishing improvements in SAP versus standard prompting.

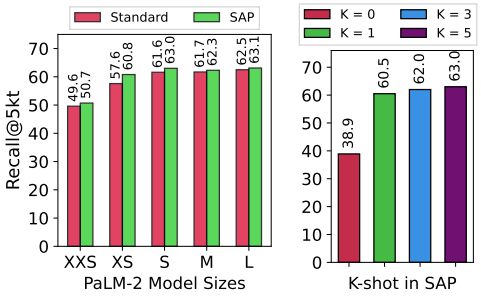


Figure 6.3: SAP versus Standard prompting and K-shot ablations.

Optimal Value of K-shot Exemplars. We investigate the optimal number of few-shot exemplars required by PaLM 2 and the resulting variation in retrieval performance on XOR-Retrieve.[7] From Figure 6.3 (right), we observe roughly a linear improvement in Success@5kt as K increases. Best Success@5kt is observed with K = 5. SAP is unable to perform well in the zero-shot setting (i.e., K = 0) due to the complexity of the multilingual query generation, which requires few-shot exemplars to learn the task.

[7]We limit K = 5 to fit within a context length of 4096 tokens. For additional exemplars, PaLM 2 would need a longer context length and signficantly increase the computational cost.

6.3.6 How much Synthetic Data to Generate?

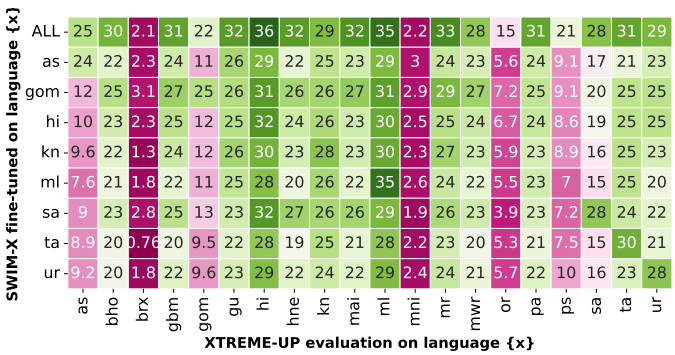
We analyze the optimal amount of synthetic training data required to fine-tune SWIM-X. Figure 6.4b shows the relative improvement in Success@5kt on XOR-Retrieve. SWIM-X performance increases gradually and starts to saturate after 500K synthetic training pairs. Our first observation is that with only 2K training pairs, SWIM-X (2K) achieves 49.1 Success@5kt on XOR-Retrieve, already outperforming the best zero-shot (English-only) baseline. The break-even point occurs at 200K pairs, where SWIM-X (250K) achieves 60.5, outperforming mContriever-X, which achieves a 59.6 Success@5kt on XOR-Retrieve.

6.3.7 Indo-European Language Transferability

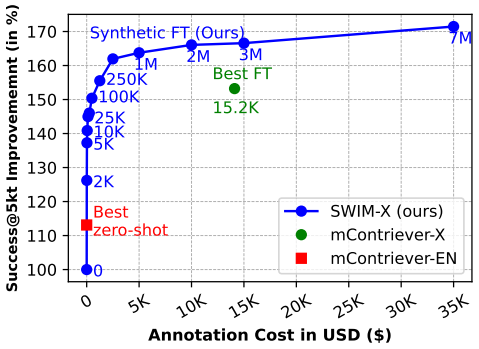
We investigate the language transfer capabilities within the Indic split (Indo-European language family) in SWIM-IR. We fine-tune individual SWIM-X models for eight selected languages and evaluate them on XTREME-UP. From Figure 6.4a, we observe that SWIM-X models fine-tuned for Konkani (`gom`) or Hindi (`hi`) transfer best across all languages in XTREME-UP (rows 3 and 4), whereas fine-tuning on Tamil (`ta`) transfers worst overall (row 8). Assamese (`as`), Konkani (`gom`), Odia (`or`), Pashto (`pa`) and Sanskrit (`sa`) exhibit the lowest zero-shot capabilities with SWIM-X, highlighting the importance of in-language synthetic data. Hindi (`hi`), Kannada (`kn`) and Malayalam (`ml`) demonstrate strong zero-shot transfer capabilities across all Indic languages.

6.4 Cost Comparison

Generating synthetic training data is relatively inexpensive, though it is not free. The cost depends on the length of the prompt, input, and output generated by the LLM. The costs also linearly increase with each additional language pair. At the time of writing, PaLM 2 and similar LLMs cost about 0.0005 USD for 1000 characters in the input and output text. Our prompts on average contain about 8–9K characters in the prompt input and generate about 1–2K characters in the output. The relative performance improvement associated with annotation cost in XOR-Retrieve is shown in Figure 6.4b. Generating 200K synthetic training pairs in SWIM-IR costs roughly $1K USD. SWIM-X (200K) performs comparably to the best supervised baseline (mContriever-X), trained on 15.2K human-annotated pairs. Producing these annotations requires roughly 14× more cost, i.e., $14.1K USD to annotate, if we pay an hourly rate of $18.50 USD/hr for annotators (local minimum wages is $11.50 USD/hr) following [398], assuming an annotation time of 3 minutes per example [288].



(a) Heatmap showing MRR@10 scroes denoting language-based transfer ability of SWIM-X (120K) across Indo-European languages in XTREME-UP [288]. (ALL) denotes fine-tuning on all languages.



(b) Success@5kt improvement on XOR-Retrieve versus annotation cost (in USD). The number of training pairs are shown besides each point.

Figure 6.4: Language transfer patterns of SWIM-X (left) and the impact of annotation cost on retrieval performance (right).

Answer to RQ3: *Using synthetic data, approximately 200K training pairs in SWIM-IR achieve performance comparable to 15K supervised training pairs in XOR-Retrieve, while being 14× cheaper to generate synthetic data than relying on human judgments.*

6.5 Summary

In this chapter, we presented SWIM-IR, a synthetic multilingual retrieval training dataset with 28 million training pairs across 33 diverse languages. SWIM-IR enabled inexpensive synthetic fine-tuning of multilingual dense retrieval models without human supervision. The dataset was constructed using SAP (*summarize-then-ask prompting*), which assisted the LLM in identifying the relevant sections of the input passage, and improved the quality of the generated multilingual queries.

Our evaluation across three multilingual retrieval benchmarks assessed the quality of our dataset. We found that SWIM-X, fine-tuned on SWIM-IR (while keeping model training parameters unchanged) outperformed the best supervised cross-lingual baseline by 7.1 points Success@5kt on XOR-Retrieve and 11.7 points MRR@10 on XTREME-UP, while remaining competitive in monolingual retrieval on MIRACL.

In the next chapter, we will revisit supervised training datasets and examine how to curate them to improve their quality through dataset pruning and hard-negative mining.

Chapter 7

Revisiting Supervised Training Data Quality for Robust Retrieval

As established in Chapter 1, a combination of synthetic and supervised training data is used to train SoTA retrieval models, such as Qwen-3 Embeddings [399]. In the previous chapter, we examined the problem of data scarcity in multilingual resources and examined techniques to generate synthetic training data to alleviate the issue of uneven and low distribution of human-supervised training data in non-English languages.

In this chapter, we shift our focus to supervised (i.e., human-annotated) training datasets, and examine issues within them, and remedy them for large training collections using: (1) dataset-level pruning and (2) relabeling of false negatives (at a data sample level).

7.1 Overview

State-of-the-art (SoTA) retrieval or embedding models are typically fine-tuned on lots of supervised training pairs [399]. While the general notion is that more training data is better, in accordance with scaling laws [59, 181, 231], we show the contrary: fine-tuning on a select few datasets is rather crucial. For example, removing ELI5 [98] surprisingly improves nDCG@10 on 7 out of 14 of the BEIR datasets [329] and the average nDCG@10 by 0.6 points. A similar observation is also made on other training datasets: by pruning 8 out of the 15 datasets in the BGE training collection [181],[1] the $E5_{base}$ retrieval model improves by 1.0 point nDCG@10 on BEIR (as shown later in Figure 7.4).

[1]The pruned dataset contains only 42.5% training pairs of the original dataset, making it 2.35× smaller.

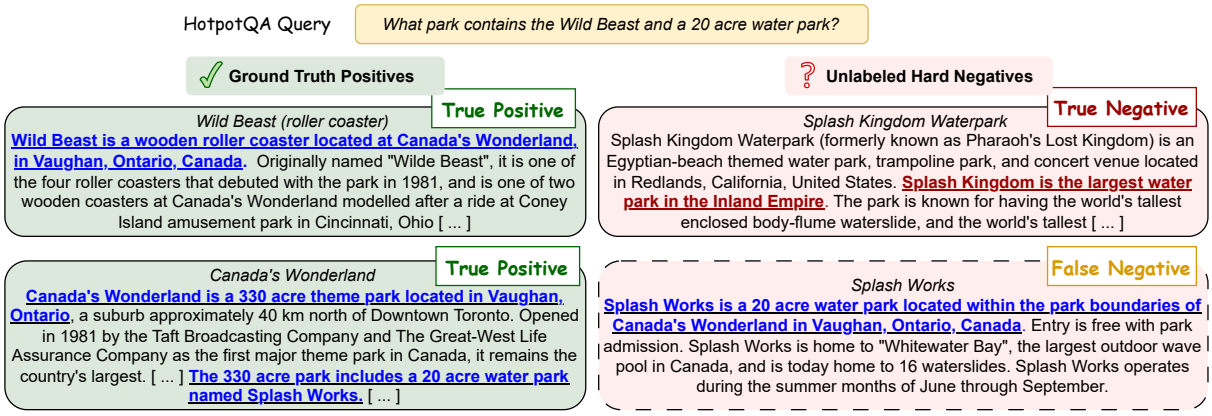


Figure 7.1: Example of a training instance (query, ground truth positives, and unlabeled hard negatives) with detected false negatives taken from HotpotQA. The false negative passage (*Splash Works*) is **mislabeled** as it is relevant in answering the user's query. The relevant parts of the text useful in answering the query are highlighted in **blue**.

The above observation reveals a non-negligible amount of "false" or mislabeled data is mixed in the current training datasets, caused likely due to *pooling biases* [49], which adds unnecessary training cost and hurts model training. *How can the "false" data be eliminated?* We approach from the perspective of *false negatives* (example in Figure 7.1), specifically, by proposing **RLHN** (ReLabeling Hard Negatives) utilizing a cost-effective framework with large language model (LLM) cascading [61] to accurately identify and relabel false negatives (at a data sample level). We choose to look at false negatives since it is a systematic pitfall from how retrieval training data is constructed:[2] As long as there are *unjudged* documents used as negative examples, the issue of false negative persists, which is especially severe for big sparsely-annotated datasets, such as MS MARCO [29] or NQ [165].

The issue of false negatives has been noticed for long — Qu et al. [263] distill knowledge from a cross-encoder to alleviate their impact. Moreira et al. [228] filter potential false negatives based on relevance score to the query. However, the former solution does not curate or clean the training datasets and is based on the assumption that the cross-encoder is more robust to false negatives than retrieval models. As we will show in Section 9.5, albeit smaller, inferior training data also negatively affect cross-encoders. The latter solution is based on the assumption that the relevance scores of false negatives are systematically higher than 95% of the positive scores, which does not consider score variance at the level of a data instance.

[2] In contrast to "false positives", that only results from mistakes of human annotators in training datasets.

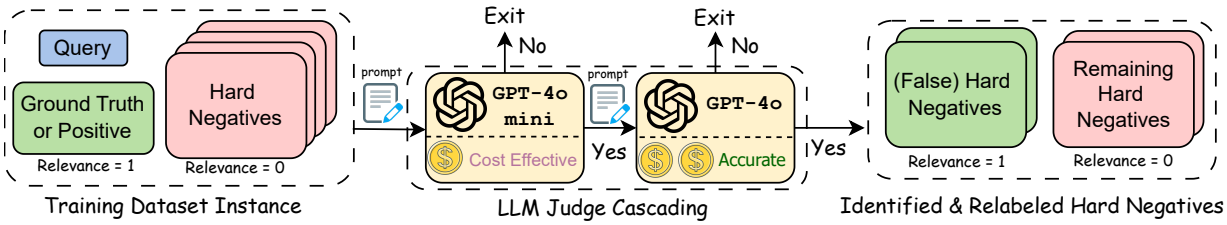


Figure 7.2: Flowchart for **RLHN** (**R**e**L**abeling **H**ard **N**egatives): (1) Provide the query, ground-truth or positive passages, and hard negative passages from a training instance as input, (2) Prompt a cost-effective LLM judge (e.g., GPT-4o-mini) and evaluate whether any hard negative is misclassified, (3) If yes, repeat the prompt with an accurate LLM judge (e.g., GPT-4o) (4) Output the relabeled hard negative passages (which are found relevant) and either remove them or relabel them as ground-truth passages in our experiments.

In this chapter, we ask the following four research questions that guide this work:

RQ1 *How prevalent are false hard negatives in popular retrieval training datasets?*
RQ2 *Does scaling supervised retrieval data monotonically improve retrieval accuracy?*
RQ3 *Does curating training data help improve retrieval model generalization?*
RQ4 *How does data curation compare to existing false negative mitigation strategies?*

7.2 LLM Judge Cascading Framework

We adopt a simple and cost-effective approach that uses cascaded LLM judges (shown in Figure 7.2), inspired by Chen et al. [61], to identify false hard negatives datasets at scale. The cascading framework is spot-checked by human assessors, achieving a decent agreement (results in Section 7.5). As the framework relies solely on LLM judges, there can be technical debt: model behavior, availability, and cost may change over time. Further, we discuss causes for error in false negative detection in Section 7.6. The framework involves state-of-the-art LLM judges for assessment in two stages:

1. **Cost-effective judge (GPT-4o-mini):** In the first stage, we prompt GPT-4o-mini [247], a cost-effective LLM judge, to improve **recall** by identifying potential pairs with false negatives through independent evaluation of each training pair in a dataset.
2. **Accurate judge (GPT-4o):** In the second stage, we prompt GPT-4o [247], a more reliable but expensive judge[3] to re-evaluate the candidate pairs with potential false negatives identified by GPT-4o-mini in order to improve **precision**.

[3]GPT-4o-mini and GPT-4o pricing (as of May 15th, 2025) is 0.6$ and 5.0$ for 1M input tokens and

Dataset	#Train Pairs	Avg. GT/Q	Avg. HN/Q	RLHN Stage 1	RLHN Stage 2
MS MARCO	485,823	1.1	25.0	391,965	326,301
HotpotQA	84,516	2.0	20.0	11,268	4,756
NQ	58,568	1.0	98.5	32,184	19,199
FEVER	29,096	1.3	20.0	7,764	3,577
ScidocsRR	12,655	1.6	19.7	2,068	351
FiQA-2018	5,500	2.6	15.0	3,632	1,833
ArguAna	4,065	1.0	13.6	0	0

(a) BGE training collection statistics [59]. Avg. GT/Q denotes the average ground truth passages per query, and Avg. HN/Q denotes the average hard negative passages per query.

Dataset	*Cascading Stage 1* # Pairs	*Cascading Stage 1* GPT-4o-mini	*Cascading Stage 2* # Pairs	*Cascading Stage 2* GPT-4o
MS MARCO	485,823	180.40 USD	391,965	2431.98 USD
HotpotQA	84,516	43.35 USD	11,268	97.26 USD
NQ	58,568	37.41 USD	32,184	345.08 USD
FEVER	29,096	22.67 USD	7,764	103.99 USD
ScidocsRR	12,655	9.07 USD	2,068	24.81 USD
FiQA-2018	5,500	3.60 USD	3,632	40.17 USD
ArguAna	4,065	0.00 USD	4,065	0.00 USD
Total Costs		**∼300 USD**		**∼3000 USD**

(b) Cost estimates for relabeling false negatives in RLHN using GPT-4o-mini and GPT-4o. We avoid relabeling hard negatives in ArguAna due to its inherent complex task.

Table 7.1: Training dataset statistics (left) and RLHN relabeling cost analysis (right).

7.2.1 Training Dataset Modification

After identifying false negatives using the RLHN framework, we consider three operations for handling the detected false negatives:

- **Remove**: Discard the entire training instance due to low quality, even if it contains at least one false negative as judged by cascading GPT-4o-mini and GPT-4o.[4]
- **Remove HN**: Discard only the detected false negatives from the hard negative subset, while keeping the instance with the surviving hard negatives.
- **Relabel HN (RLHN)**: Relabel only the detected false negatives from the hard negative subset, by adding them to the ground truth subset, while keeping the instance with the surviving hard negatives.

7.2.2 Dataset Postprocessing & Statistics

In Table 7.1a, we report the training dataset statistics for the BGE training collection. MS MARCO contains the largest number of training pairs, followed by HotpotQA. All datasets contain training pairs with 1–3 ground-truth passages and 13–25 hard negatives (except NQ, which contains 98–100 hard negatives).

2.4$ and 20.0$ for 1M output tokens, respectively.

[4]We lose the instance completely in the "remove" technique.

False negatives. From Table 7.1a, we observe that the majority of detected false negatives occur in MS MARCO (91.6% of all detected pairs). Up to 67% of all training pairs in MS MARCO contain false negatives, compared to a low of about 3% in SciDocsRR.[5] From Figure 7.3, we observe that in 58% of all detected false negative pairs, only a single false positive was detected, and 19% with two false negatives, and less than 1% with eight or more false negatives. If we detect any training pair with detected false negatives over a certain threshold k ($k = 7$ emperically fixed in our experiments), we exclude the instance entirely in RLHN, as the query is likely ambiguous and thus might not be a useful training instance (e.g., *what color is amber urine?*).

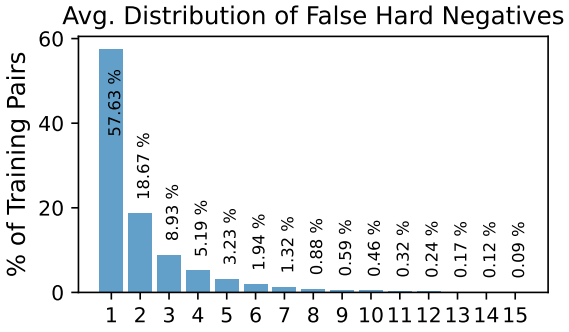


Figure 7.3: Distribution of training pairs (with at least one false negative) across detected false hard negatives. 58% have one false negative, 19% have two, and so on.

Cost estimates. We report the maximum costs incurred in RLHN (accurate input tokens and estimated 2048 output tokens on average) for both judges at each cascading stage: GPT-4o-mini and GPT-4o in Table 7.1b. Overall, running RLHN with GPT-4o-mini in Stage 1 costs approximately 300 USD, and using GPT-4o in Stage 2 costs approximately 3000 USD.

Answer to RQ1: *False hard negatives are prevalent in retrieval training datasets. Their occurence varies substantially across datasets: MS MARCO accounts for 91.6% of all detected cases, whereas SciDocsRR exhibits much lower rates (3%). These findings highlight that false negatives are a prevalent issue with large numbers of sparse judged documents.*

7.3 Experimental Setting

BGE training collection. We utilize the original BGE training collection[6] [181], a comprehensive collection that includes training datasets for retrieval (e.g., NQ, MS MARCO), clustering (e.g., TwentyNewsgroups), and classification (e.g., AmazonReviews) tasks. Many

[5] We avoid relabeling ArguAna due to its inherent complex task, which doesn't measure directly for argument similarity, but rather counter arguments given an argument. Therefore, we keep the original dataset in fine-tuning without relabeling.

[6] https://huggingface.co/datasets/cfli/bge-full-data

Backbone	#Params	#Layers	Dim.	Pool	LR	Batch Size	Time Taken	T-COVID	NFC.	FiQA	SciFact
$E5_{large}$ (unsupervised) [358]	330M	24	1024	*mean*	$1e-5$	128 x 8 x 4	$\sim$ 36 hours	<u>0.712</u>	**0.383**	**0.475**	**0.747**
ModernBERT-base [366]	149M	22	768	*mean*	$2e-4$	256 x 8 x 4	$\sim$ 12 hours	0.560	0.279	0.440	0.602
$E5_{base}$ (unsupervised) [358]	110M	12	768	*mean*	$2e-5$	256 x 8 x 4	$\sim$ 18 hours	**0.731**	<u>0.381</u>	<u>0.444</u>	<u>0.728</u>
$E5_{small}$ (unsupervised) [358]	33M	12	384	*mean*	$3e-5$	256 x 8 x 4	$\sim$ 13 hours	0.667	0.349	0.420	0.698
Qwen-2.5-0.5B [381]	500M	24	896	*last*	$1e-5$	96 x 8 x 4	$\sim$ 36 hours	0.503	0.356	0.417	0.692
SmolLM2-360M [13]	360M	32	960	*last*	$1e-5$	96 x 8 x 4	$\sim$ 33 hours	0.670	0.336	0.355	0.635
SmolLM2-135M [13]	135M	30	576	*last*	$1e-5$	128 x 8 x 4	$\sim$ 24 hours	0.668	0.327	0.304	0.608

Table 7.2: Performance of encoder (top) and decoder (bottom) backbones <= 500M parameters, fine-tuned on the full BGE training collection for 3 epochs with above mentioned hyperparameters, and evaluated on four datasets on BEIR (T-COVID is TREC-COVID, NFC. is NFCorpus). The best score is **highlighted** and the second best is <u>underlined</u>.

of these training datasets are used to fine-tune popular retriever models such as E5-Mistral [359], GRIT-LM [231], Linq [67], LLM2Vec [31], CDE [229], or NV-Embed [173]. Since our work focuses on the *retrieval task*, we remove all training datasets from clustering and classification tasks, resulting in 15 datasets focused on the retrieval task, comprising a total of 1.6M training pairs, originally released under the MIT license.

LLM judges. In our work, we use GPT-4o-mini (version `2024-07-18`) and GPT-4o (version `2024-11-20`) as the judge using the Azure OpenAI service in the *batch* setting. We follow a temperature setting of 0.1 and use a chain-of-thought prompt setting [367]. The prompt is not provided in this thesis, but can be found in the original publication. The prompt first evaluates the relevance between every hard negative passage and the question, and compares them with the ground truth to identify potential false negatives. We prompt up to a maximum of 25 hard negative passages per query in a single API call [200].

Evaluation benchmarks. We evaluate the retrieval and reranker accuracy of the models fine-tuned on datasets with false negatives either removed or relabeled with RLHN on the BEIR benchmark [329] and AIR-Bench [58]. Both benchmarks evaluate retrieval accuracy in nDCG@10. BEIR contains human-constructed datasets, and AIR-Bench contains datasets automatically generated by LLMs without human intervention. In BEIR, we drop Quora and CQADupstack and evaluate on 16 datasets. In AIR-Bench (version `24.05`), we evaluate five specific domains in English-only: Arxiv, Finance, Healthcare, Law, and News.

Backbone models. We use the $E5_{base}$ unsupervised[7] [358, 359], a BERT-based encoder, due to its high accuracy on BEIR (preliminary backbone results in Table 7.2), the inclusion of a pre-training stage, and lower training complexity. $E5_{base}$ contains 110M parameters, 12 layers, and a 768 embedding dimension with mean pooling. Also, we use an LLM-

[7] `https://huggingface.co/intfloat/e5-base-unsupervised` on Hugging Face.

BEIR Dataset	*No Filtering*	*Baselines*		*Cascading Stage 1*			*Cascading Stage 2*			*No Filtering*	*Cascading Stage 2*	
	Default	**TopK-PercPos**	**CE Distill**	**Rem.**	**Rem. HN**	**RLHN**	**Rem.**	**Rem. HN**	**RLHN**	**Default**	**Rem. HN**	**RLHN**
Backbone	$E5_{base}$	$E5_{base}$	$E5_{base}$	$E5_{base}$	$E5_{base}$	$E5_{base}$	$E5_{base}$	$E5_{base}$	$E5_{base}$	Qwen2.5-7B	Qwen2.5-7B	Qwen2.5-7B
TREC-COVID†	0.783	0.789	0.793	0.786	0.793	0.798	0.794	0.785	0.809	0.797	0.771	0.815
NFCorpus†	0.378	0.377	0.363	0.378	0.380	0.381	0.380	0.382	0.390	0.389	0.389	0.391
NQ	0.595	0.601	0.624	0.593	0.592	0.602	0.573	0.598	0.591	0.597	0.602	0.623
HotpotQA	0.737	0.734	0.741	0.737	0.736	0.739	0.741	0.736	0.735	0.704	0.702	0.729
FiQA-2018	0.439	0.434	0.417	0.443	0.440	0.444	0.441	0.445	0.448	0.453	0.461	0.465
ArguAna	0.701	0.697	0.725	0.702	0.706	0.700	0.700	0.700	0.692	0.554	0.550	0.560
Touché-2020†	0.256	0.286	0.305	0.255	0.271	0.268	0.218	0.265	0.266	0.221	0.211	0.230
DBPedia	0.438	0.444	0.446	0.439	0.437	0.442	0.433	0.441	0.447	0.443	0.456	0.472
SCIDOCS	0.242	0.243	0.216	0.243	0.243	0.244	0.245	0.243	0.242	0.245	0.243	0.252
FEVER	0.878	0.878	0.889	0.875	0.876	0.877	0.881	0.876	0.871	0.863	0.857	0.872
Climate-FEVER	0.391	0.386	0.377	0.388	0.385	0.391	0.382	0.384	0.367	0.370	0.373	0.360
SciFact	0.735	0.735	0.727	0.741	0.731	0.733	0.744	0.735	0.740	0.755	0.755	0.767
TREC-NEWS†	0.465	0.466	0.458	0.470	0.466	0.473	0.464	0.473	0.484	0.494	0.480	0.487
Robust04†	0.442	0.451	0.452	0.448	0.452	0.471	0.447	0.458	0.497	0.501	0.501	0.540
Signal-1M (RT)†	0.275	0.272	0.271	0.279	0.275	0.275	0.274	0.270	0.274	0.275	0.268	0.280
BioASQ†	0.378	0.375	0.413	0.382	0.385	0.392	0.384	0.384	0.394	0.408	0.412	0.438
Avg. 16 (All)	0.508	0.511	0.514	0.510	0.511	0.514	0.506	0.511	**0.515**	0.504	0.502	**0.518**
Avg. 7 (OOD)	0.425	0.431	0.436	0.428	0.432	0.437	0.423	0.431	**0.445**	0.441	0.433	**0.454**

Table 7.3: Retrieval results measuring nDCG@10 on 16 datasets in the BEIR benchmark by fine-tuning retrieval models on 7 datasets from the BGE training collection with relabeled false negatives. The seven unseen (or out-of-domain) datasets during fine-tuning are highlighted with † and their average scores are provided in Avg. 7; Rem. denotes Remove.

based decoder model with Qwen2.5-7B [381] with 7.61B parameters, 28 layers, and a 3584 embedding dimension with the `[EOS]` token pooling as the retrieval models. In addition, we use the Qwen2.5-3B model [381] for the reranker.

Fine-tuning details. All models are fine-tuned using 7 hard negatives, 1 positive, and random in-batch negatives (128 total) per batch, optimized with the InfoNCE loss function [343] using the Tevatron repository[8] [111, 210] for up to 4–5 epochs, with a learning rate of 2e-5, and a maximum sequence length of 350 tokens (512 tokens during inference). We append a "`query: `" and "`passage: `" prefix. $E5_{base}$ models are fine-tuned using 4×L40S GPUs, and Qwen2.5-7B and Qwen2.5-3B using a maximum of 2×H200 GPUs.

Baselines. To evaluate the impact of relabeling hard negatives using RLHN, we include two baselines: (1) *hard-negative mining*: Top-95% TopK-PecPos sampling [228] on the default training dataset, using similarity scores computed for all hard negatives with the `bge-reranker-v2-gemma` reranker, and (2) *cross-encoder distillation*: we compute the normalized similarity scores for all query and hard negatives and positive pair on the default training dataset with the `bge-reranker-v2-gemma` reranker. We then fine-tune $E5_{base}$ using knowledge distillation from the cross-encoder scores, with 1 positive, 15 hard negatives, and no in-batch negatives, using Tevatron.

[8] https://github.com/texttron/tevatron

7.4 Experimental Results

7.4.1 Preliminary Results: Dataset Pruning

False datapoint can hurt the training of retriever models. We assess the individual dataset contribution by evaluating multiple model variants by leaving one dataset out and fine-tuning on the rest. As we need to fine-tune many models, i.e., one for each removed dataset, we limit these experiments to $E5_{base}$. The summarized results in Figure 7.4 demonstrates that training datasets (highlighted in green) can hurt the model retrieval accuracy, such as ELI5, removing which improves the nDCG@10 on BEIR (0.519 → 0.525). Also, it shows that certain datasets (highlighted in red) are crucial for model accuracy.

Based on findings in Figure 7.4 and selecting necessary datasets for individual task-based performances in BEIR, we prune the original 16 retrieval datasets in the BGE collection and select seven datasets (marked as $**$), reducing the training dataset size from 1.6M to 680K training pairs in our experiments. The average nDCG@10 score of $E5_{base}$ improves from 0.519 → 0.529 on 14 datasets on average in BEIR, despite fine-tuning on almost 2.35× smaller dataset (1.6M → 680K).

Answer to RQ2: *Fine-tuning retrieval models on larger training collections, or simply scaling supervised data, does not monotonically improve retrieval accuracy. Our experiments with* $E5_{base}$ *show that certain datasets–such as ELI5–systematically degrade nDCG@10 on BEIR. Conversely, a few datasets are critical for maintaining a strong retrieval accuracy.*

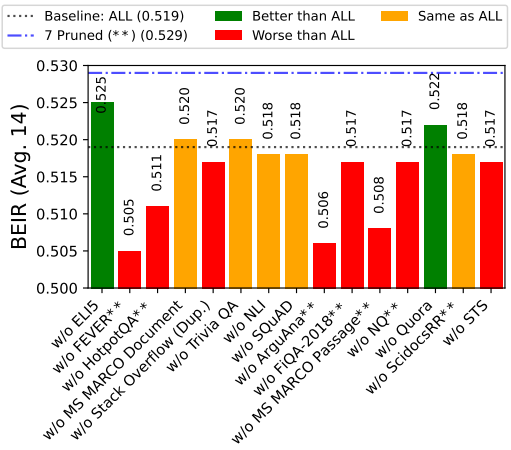


Figure 7.4: Dataset pruning by leaving one dataset out during fine-tuning $E5_{base}$ on the BGE-training collection.

7.4.2 Main Results: Relabeling False Negatives

This section presents the results of models with $E5_{base}$ and Qwen2.5-7B backbones fine-tuned on the variants of the training dataset described in Section 7.2 and 7.2.1 with relabeled false negatives, while keeping the rest of the model training parameters unchanged.

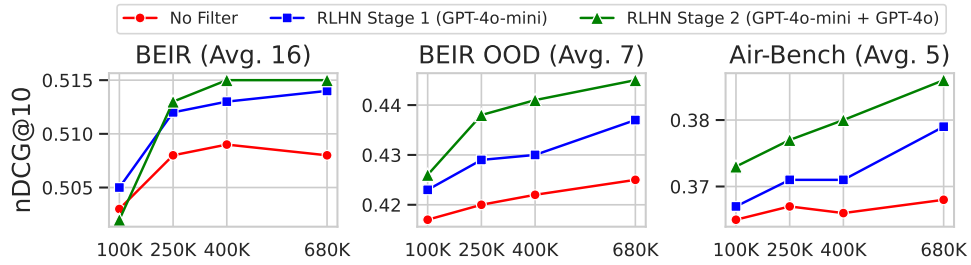


Figure 7.5: nDCG@10 scores on BEIR (Avg. 16 and Avg. 7) and AIR-Bench (Avg. 5) by fine-tuning $E5_{base}$ on a subset of the 100K, 250K, 400K, and 680K training pairs using the "RLHN" technique for both stages.

BEIR benchmark. Results in Table 7.3 show that for both $E5_{base}$ and Qwen2.5-7B, the RLHN technique achieves the best overall average nDCG@10 of 0.515 and 0.518 on 16 datasets on BEIR, outperforming models trained with the default setting and other remove techniques. The relabeled data in RLHN improves model generalization, with improvements strongly visible in seven out-of-domain (OOD) datasets in BEIR. Stage 1 (RLHN) outperforms the Default setting by 2.0 points and Stage 2 (RLHN) by 3.2 points in nDCG@10. Overall, relabeling false negatives improves the data quality, which is reflected in model generalization across out-of-domain settings in BEIR.

AIR-Bench. In addition to BEIR, AIR-Bench provides a zero-shot setting to evaluate on challenging domains, such as Law. Table 7.4 shows the average nDCG@10 on five specialized domains. The improvements in model generalization are consistent to what we observed in BEIR. Stage 1 (RLHN) improves the Default setting by 1.1 points in nDCG@10, and Stage 2 (RLHN) further improves by 2.1 points. Overall, without changing the model or training parameters, mitigating false negatives in training datasets with RLHN enables the model generalize better to specialized domains in AIR-Bench.

Backbone	Technique	Arxiv	Finance	Health.	Law	News	Avg. 5
$E5_{base}$	**Default**	0.345	0.401	0.521	0.117	0.455	0.368
$E5_{base}$	**TopK-PercPos**	0.348	0.418	0.529	0.119	0.464	0.376
$E5_{base}$	**CE Distill**	0.372	0.430	0.536	0.168	0.498	**0.401**
	Stage 1: GPT-4o-mini						
$E5_{base}$	**Remove**	0.346	0.407	0.526	0.118	0.452	0.370
$E5_{base}$	**Remove HN**	0.344	0.406	0.522	0.118	0.459	0.370
$E5_{base}$	**RLHN**	0.362	0.421	0.522	0.123	0.465	**0.379**
	Stage 2: GPT-4o-mini + GPT-4o						
$E5_{base}$	**Remove**	0.341	0.403	0.514	0.125	0.438	0.364
$E5_{base}$	**Remove HN**	0.346	0.411	0.525	0.124	0.464	0.374
$E5_{base}$	**RLHN**	0.356	0.440	0.521	0.138	0.476	**0.386**
Qwen2.5-7B	**Default**	0.325	0.391	0.479	0.115	0.430	0.348
	Stage 2: GPT-4o-mini + GPT-4o						
Qwen2.5-7B	**Remove HN**	0.335	0.384	0.487	0.111	0.423	0.348
Qwen2.5-7B	**RLHN**	0.330	0.418	0.494	0.133	0.450	**0.365**

Table 7.4: Retrieval results measuring nDCG@10 on five specialized domains in AIR-Bench dev (version `24.05`) after fine-tuning $E5_{base}$ and Qwen2.5-7B.

Robustness of RLHN across varying training data subsets. As training datasets can be large, relabeling all training pairs using the LLM cascading pipeline can be computationally prohibitive. From Figure 7.5, we demonstrate that RLHN remains robust

and maintains similar accuracy gains, even when applied to smaller randomly sampled subsets of the training dataset. To evaluate this, we use four random subsets (100K, 250K, 400K, and 680K) of the training datasets. Overall, we have two main findings: (i) the $E5_{base}$ model fine-tuned on RHLN Stages 1 and 2 training data, with false hard negatives relabeled as positives, *consistently* outperforms the Default setting, and (ii) the steeper slope in nDCG@10 demonstrates *continual improvement* across zero-shot domains, as the amount of training data increases, especially as observed in AIR-Bench. While we do not report formal statistical tests here, the consistency of improvements on both BEIR and AIR-Bench supports the reliability of the RLHN curated training datasets.

Answer to RQ3: *Curating training data by identifying and correcting false negatives improves retrieval model generalization, particularly on out-of-domain retrieval settings. RLHN trained* $E5_{base}$ *and Qwen2.5-7B in BEIR (OOD) and AIR-Bench, consistently outperform models trained precariously on original datasets with false negatives. These improvements are persistent even when applied to smaller subsets of the training dataset.*

Comparison with baselines. Results in Table 7.3 and Table 7.4 show that carefully avoiding sampling very hard negatives using Top-95%-PercPos outperforms the Default model, but still underperforms compared to the RLHN strategy with $E5_{base}$. Next, the `bge-reranker-v2-gemma` cross-encoder, used as the distillation teacher is a strong baseline. It slightly underperforms RLHN on BEIR but outperforms RLHN on AIR-Bench. However, we want to reiterate that our core motivation is to *identify* and *relabel* false negatives in training datasets to enhance data quality. Distillation-based fine-tuning requires on a strong, domain-focused cross-encoder reranker. Similarly, RLHN is particularly valuable for fine-tuning cross-encoders when teacher supervision is not viable.

Reranker results. Training data with improved quality also benefits cross-encoder rerankers. Table 7.5 shows the result comparison on the BEIR benchmark, where we rerank the top-100 results from the fine-tuned $E5_{base}$ in the Default setting. Training rerankers with data fixed on RLHN Stages 1 and 2

BEIR Dataset	Default	Stage 1	Stage 2
TREC-COVID†	0.836	0.861	0.862
NFCorpus†	0.401	0.414	0.415
NQ	0.730	0.739	0.736
HotpotQA	0.863	0.861	0.861
FiQA-2018	0.517	0.521	0.519
ArguAna	0.740	0.730	0.763
Touché-2020†	0.275	0.308	0.313
DBPedia	0.532	0.536	0.538
SCIDOCS	0.278	0.273	0.270
FEVER	0.941	0.939	0.936
Climate-FEVER	0.457	0.468	0.430
SciFact	0.786	0.793	0.794
TREC-NEWS†	0.507	0.513	0.527
Robust04†	0.531	0.548	0.589
Signal-1M†	0.292	0.276	0.274
BioASQ†	0.510	0.505	0.500
Avg. 16 (All)	0.575	0.580	**0.583**
Avg. 7 (OOD)	0.479	0.489	**0.497**

Table 7.5: Reranker results measuring nDCG@10 on 16 datasets in BEIR by fine-tuning reranker models (based on Qwen2.5-3B) on RHLN (Stage 1 and Stage 2) variants.

progressively increases nDCG@10 on BEIR datasets by 0.5 points and 0.8 points, respectively. This improvement is most prominent on the seven OOD datasets, consistent with the above observation on retrievers: the data correction on the two stages improves the averaged OOD results by 1.0 and 1.8 points, respectively. We note that the scale of the improvement on cross-encoders is not as large as on retrievers, which may indicate that cross-encoder rerankers are comparatively more robust to false negatives. However, albeit small, cross-encoders still benefit from training data of higher quality, especially when generalizing to unseen domains.

Answer to RQ4: *RLHN for data curation directly tackles the problem by curating the training dataset as a reusable community resource. RLHN outperforms hard-negative mining and is competitive with cross-encoder distillation, which relies on computationally expensive SoTA rerankers for offline labeling.*

7.5 Human Validation

We conducted a validation study with three human assessors (the authors from the original publication) using Label Studio[9] for data annotation. The assessors were briefed on the relevance task, and then independently evaluated a total of 670 query–hard negative pairs. The hard negatives were randomly sampled, each containing at least one false negative. During the assessment, all annotators worked independently and were not exposed to the LLM predictions.

Table 7.6 reports Cohen's Kappa (κ) measuring agreement between each LLM's predictions and the human labels. The κ scores are consistent with prior work reporting similar levels of human–LLM agreement [19]. GPT-4o shows substantially higher agreement with human annotators compated to GPT-4o-mini. This finding aligns with our empirical results, where relabeling with GPT-4o shows consistent gains over GPT-4o-mini in training retrieval and reranker models.

Metric	GPT-4o-mini	GPT-4o
Cohen's Kappa (κ)	0.320	0.390

Table 7.6: Cohen's κ agreement of GPT-4o-mini and GPT-4o with judgments on 670 query–negative pairs with three human assessors (the authors from the original publication).

[9] https://github.com/HumanSignal/label-studio/

Query	Ground Truth or Positive Passages		False Negatives (Detected by RLHN)	
(Q1) Which is a food magazine, Latin Mass Magazine or Saveur?	**Latin Mass Magazine**: A Journal of Catholic Culture, commonly referred to as Latin Mass Magazine, is an American Catholic magazine published quarterly, with a traditionalist Catholic viewpoint. [...]	**Saveur**: Saveur is a gourmet, food, wine, and travel magazine that specializes in essays about various world cuisines. Its slogan—"Savor a World of Authentic Cuisine"—signals the publication's focus on enduring culinary traditions [...]	**Food & Wine**: Food & Wine is a monthly magazine published by Time Inc. It was founded in 1978 by Ariane and Michael Batterberry. It features recipes, cooking tips, travel information, restaurant reviews, chefs, wine pairings and seasonal content [...]	**Cocina (magazine)**: is a Colombian-based monthly magazine published by Publicaciones Semana S.A.. It features recipes, cooking tips, culinary tourism information, restaurant reviews, chefs, wine pairings and seasonal holiday content [...]
(Q2) What year was the premier professional ice hockey league in the world established?	**2016–17 Minnesota Wild season**: The 2016–17 Minnesota Wild season was the 17th season for the National Hockey League franchise that was established on June 25, 1997.	**National Hockey League**: The National Hockey League (NHL; French: "Ligue nationale de hockey—LNH") is a professional ice hockey league currently comprising 31 teams [...]	**History of the National Hockey League (1917–42)**: History of the National Hockey League (1917–42) The National Hockey League (NHL) was founded in 1917 following the demise of its predecessor league, the National Hockey Association (NHA). [...]	
(Q3) name meaning yin and yang	**Yin and yang**: In Chinese philosophy, yin and yang (also, yin-yang or yin yang) describes how apparently opposite or contrary forces are actually complementary, interconnected, and interdependent in the natural world, and how they give rise to each other as they interrelate to one another.		**Yin and yang**: Yin and Yang are ancient Chinese philosophical terms, with the Yin Yang Theory being a fundamental part of Feng Shui. It is a Chinese theory on the perspective of continuous change and balance. [...]	**Yin Yang Symbols and Their Meanings**: In a nutshell, Chinese yin yang symbols represent perfect balance. A great deal of Chinese philosophy stems from the concept of yin and yang - opposites interacting [...]
(Q4) Charles, Prince of Wales is patron of numerous other organizations.	**Charles, Prince of Wales**: Charles, Prince of Wales (born 14 November 1948) is the eldest child and heir apparent of Queen Elizabeth II [...] Charles's interests encompass a range of humanitarian and social issues: he founded The Prince's Trust in 1976, sponsors The Prince's Charities, and is patron of numerous other charitable and arts organisations. [...]		**Julia Cleverdon Dame**: Julia Charity Cleverdon [...] served for 16 years as Chief Executive of Business in the Community, one of the Prince's Charities of Charles, Prince of Wales.	**The Prince's Trust**: The Prince's Trust is a charity in the United Kingdom founded in 1976 by Charles, Prince of Wales, and Frederick John Pervin to help young people. [...]

Table 7.7: Qualitative analysis showcasing the different varieties of false negatives detected by RLHN. The first two questions are taken from HotpotQA, the third from MS MARCO, and the last from FEVER. The text supporting the query is highlighted in **green**, partially supporting in **orange**, and not supporting with **red**.

7.6 Qualitative Analysis of False Negatives

We qualitatively analyze the labeling accuracy of our LLM cascading framework by manually spot-checking a few training instances. As shown in Table 7.7, we observe a variety of false negatives, which fall into the following scenarios:

1. Detected false negatives are incorrect or not relevant. GPT-4o can sometimes detect a false negative that is not relevant to the query. E.g., (Q1) query asks which is a food magazine between *Latin Mass* or *Saveur*, however, the detected false negatives identify different food magazines such as *Food & Wine* or *Cochina*, which are both incorrect.

2. The ground truth may be incorrectly labeled. In a few queries, we observe that the ground truth passage can contain conflicting information with the false negative, resulting in incorrect labeling. E.g., the correct answer to the (Q2) query, which asks about the professional ice hockey establishment is 1917 (present in the false negative). However, the ground truth incorrectly states 1997.

3. The query may be too generic or ambiguous. In a substantial amount of training pairs in MS MARCO, we find that the training query is rather ambiguous, leading to many false negatives being detected. E.g., for the (Q3) query, all passages–including both

the ground truth and false negatives–are relevant, as they each correctly define "yin and yang" but with different interpretations.

4. False negatives can be partially correct. Not all detected false negatives are entirely non-relevant to the query. E.g., one false negative is partially relevant to (Q4), which asks about organizations associated with Charles, the Prince of Wales.

7.7 Summary

In this chapter, we emphasized the importance of clean training datasets for fine-tuning retrieval models. First, we observed a departure from the conventional notion of data scaling laws, as we showed that certain datasets can negatively impact model effectiveness when fine-tuned within a large collection containing many training pairs. Empirical dataset pruning removed 57.5% (8 out of 15 datasets in the BGE training collection), improved model accuracy on BEIR by up to 1.0 point, and reduced the dataset size by 2.35×, thereby lowering training time.

Next, after pruning, we observed the issue of false hard negatives in the remaining datasets, where passages in the hard-negative lists were misclassified and were actually relevant to the query. We presented RLHN, an effective cascading LLM-based (or frugal) approach for relabeling hard negatives as ground truth positives.

Using RLHN, we identified ∼356K out of 680K training pairs with at least one false negative across six datasets in the BGE training collection and relabeled them. Fine-tuning retrievers and rerankers on RLHN-curated training data consistently improved model generalization on both BEIR and zero-shot AIR-Bench benchmarks, with continued gains as the amount of clean training data increased. RLHN using proprietary judges such as GPT-4o-mini and GPT-4o cost under USD 3.3K, which was substantially less than the cost of having human assessors rejudge the relevance sets from scratch.

RLHN-based data curation outperformed several comparable baselines, including hard-negative mining, and was comparable to cross-encoder distillation, which requires a computationally expensive state-of-the-art reranker. Finally, in terms of limitations, RLHN ultimately relied on LLMs as judges, which can still produce incorrect labels or conflict with the ground truth.

Chapter 8

A Framework for Benchmarking Retrieval-Augmented Generation

As the previous chapters have established, accelerating research progress in information retrieval requires the development of benchmarks, data and training strategies, as well as evaluation methodologies. Until Chapter 7, we focused exclusively on retrieval benchmarks, data, and training setups. From this chapter onward, we shift our attention to retrieval-augmented generation (RAG) benchmarking and evaluation.

8.1 Overview

Retrieval-Augmented Generation (RAG) [119, 180, 139, 42] has emerged as a popular technique to augment large language model (LLM) generation for knowledge-intensive tasks such as open-domain question answering or fact verification [254]. Using the top-k retrieved segments from a suitable retrieval system, RAG systems output an answer summary grounded on the relevant context.

However, there is a lack of a unified, open-source framework that integrates SoTA components and provides the necessary resources for rigorous and reproducible RAG evaluation. Without such a foundation, comparing different systems becomes an unreliable process. This chapter introduces Ragnarök, a reusable, end-to-end RAG framework for reproducible baselines. Ragnarök is used to provide official baselines in the TREC 2024 and 2025 Retrieval-Augmented Generation (RAG) tracks, serving as a unified platform also for participants to submit their baselines and systems. In this chapter, we curate

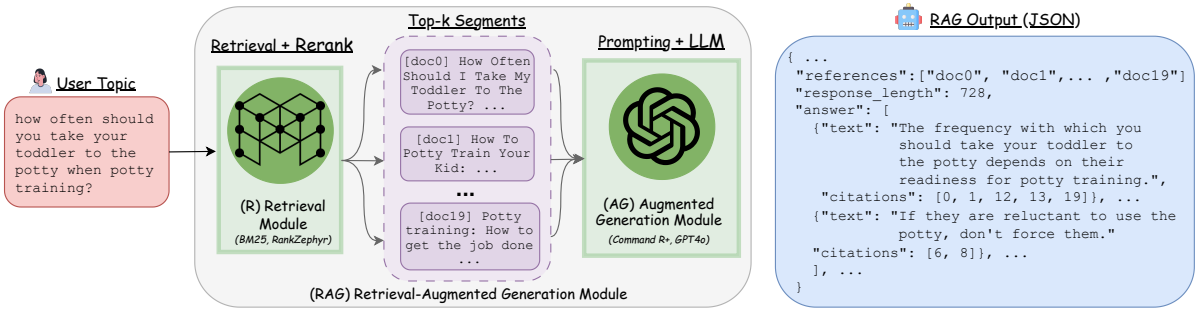


Figure 8.1: Schematic diagram of the Ragnarök. Given a user topic (left), the process consists of two steps: (1) (R) retrieval (+ rerank), where the topic yields the top-k relevant segments from our document collection (e.g., potty training articles); and (2) (AG) augmented-generation, where the retrieved segments with a suitable prompt template are fed to the large language model (LLM) to generate the post-processed answer response (JSON) containing individual sentence-level citations.

and provide necessary data sources for RAG evaluation, e.g., deduplicated data collections and non-factoid topics requiring long-form answers, necessary for benchmarking long-form answers generated by RAG systems. We ask the following research questions in this chapter:

RQ1 *How can we design a modular RAG framework for benchmarking systems?*
RQ2 *How do we construct a reliable large-scale document collection and diverse topic sets?*
RQ3 *What is the effectiveness of different retrieval pipelines on this new benchmark?*
RQ4 *How do generation models compare in answer characteristics and citation patterns?*

By answering these questions, this chapter provides detail on the creation of a comprehensive benchmarking ecosystem. We describe the Ragnarök framework, the curation of the MS MARCO V2.1 collection, the development of the TREC-RAGgy 2024 and TREC-Researchy 2024 topic sets, and the establishment of strong RAG baseline systems. Ragnarök provides the essential framework and baselines using which the evaluation methodologies discussed ahead in Chapter 9 build upon.

8.2 Ragnarök Framework Description

Ragnarök is an open-source, reproducible, and reusable framework implementing an end-to-end retrieval-augmented generation (RAG) pipeline, comprising two modules applied sequentially: (1) (R) retrieval and (2) (AG) augmented generation. Through Ragnarök, we

provide several baselines to all participants at TREC RAG. An overview of the framework is provided in Figure 8.1. We first describe both modules and expand on the I/O specifications.

Retrieval Module. This module retrieves the segments relevant for a user topic as the input. It supports (i) first-stage lexical retrieval models such as BM25 [281] or dense retrieval models such as GTE-L or ArcticEmbed-M [186, 222, 388] and (ii) reranking models such as RankZephyr [257]. The retrieval system searches for potentially relevant segments (definition of a segment is provided in Section 8.3) in the document collection and retrieves the top-100 segments further reranked by the reranker model to filter out the top-20 relevant segments for the next stage.

Augmented Generation Module. This module takes in the user topic and the top-20 retrieved segments (from the retrieval module) as the input and a prompting strategy to the LLM to generate the answer response with in-context citations for the topic. The complete answer response is divided into individual sentences using spaCy sentence segmentation (`en_core_web_trf`) [129]. Each sentence within the answer contains text and is grounded on retrieved documents provided as references to the LLM (if possible). We support most state-of-the-art propietary models (GPT-4o. Gemini 2.5, etc.) through OpenRouter,[1] and open-source models (Command R+, LLaMA-3.1-70B-Instruct, etc.) through vLLM [166].

8.2.1 RAG Input/Output Definitions

RAG Input. The input specifications are straightforward as the user can formulate any question they wish to ask, provide the user topic as input, and call Ragnarök.

RAG Output. The user receives a JSON output in response to their topic from Ragnarök. The first key in the output JSON schema, `references`, provides a list of segment IDs that are referenced in the answer. These segments are selected from among the top-20 results returned by the retrieval module. Next, `answer`, provides the LLM-generated RAG answer to the user topic, presented as a top-to-bottom list of sentence-level texts (spaCy sentence segmentation with `en_core_web_trf`) with corresponding segment citations. All citations are zero-based indexed, indicating the exact position of the segment ID from the `references` list. Finally, `response_length`, provides the total count of the whitespace-separated words present in the output RAG answer.

Answer to RQ1: *Ragnarök is a two-stage modular framework for providing long-form answers enabling REST APIs. The retrieval module supports first-stage retrievers (BM25,*

[1] https://openrouter.ai/

Collection	Version V2	Version V2.1 (Ours)
MS MARCO Document	11,959,635	10,960,555
MS MARCO Segment	124,131,414	113,520,750

Table 8.1: Comparison of document and segment counts between versions V2 and V2.1 (our version after removing near-duplicates) of the MS MARCO collection.

GTE-L, etc.) and rerankers (monoT5-3B, RankZephyr). The augmented generation module supports both proprietary and open-sourced LLMs. Ragnarök provides a standardized I/O definition with a JSON output with references used for citations for each answer sentence.

8.3 Document Collection

The MS MARCO V2 document collection, previously used in the TREC-DL tracks, contains a substantial overlap of near-duplicates (documents with sufficiently similar text information) [75, 78]. When left intact, these near-duplicates degrade the downstream retrieval accuracy and reduce the diversity of the collected documents, potentially impacting the effectiveness of RAG systems. Documents in the existing collection tend to be verbose, containing extensive information about individual topics. Chunking, which breaks down a long verbose document into smaller compact representations is a key challenge, as the retrieved chunk representations correlate with the RAG answer quality [200].

MS MARCO V2.1 Document Collection. We conduct a deduplication strategy in the MS MARCO V2 document collection to avoid near-duplicates in two stages. In the first stage, we establish an equivalence class of the documents using Locality Sensitive Hashing (LSH) with MinHash [44] and 9-gram shingles. Next, we select a representative document for each equivalence class for our refined MS MARCO V2.1 document collection, reducing the duplicates in the MS MARCO V2 document collection by 8.35% as shown in Table 8.1.

MS MARCO V2.1 Segment Collection. We chunk the MS MARCO V2.1 document collection into overlapping segments (or chunks) and develop the MS MARCO V2.1 segment collection, with more than 113 million text segments (Table 8.1). We utilize a sliding window technique to generate the segments, by fixing the sliding window size of 10 sentences and a stride of 5 sentences to create each segment, roughly on average, between 500–1000 characters long. To easily map each segment back to the document, every segment contains the document ID within the segment ID. Further, two new fields: `start_char` and `end_char`, which indicate the start (inclusive) and the end position character (exclusive) of where the segment begins and ends in the mapped MS MARCO V2.1 document collection.

TREC-RAGgy 2024

Topic Category	%	First Word	%
Aggregation	24.2	What	37.5
Simple w/ cond.	23.3	How	27.5
Set	20.8	Why	3.3
Simple	10.0	Is	2.5
Comparison	6.7	When	1.7

TREC-Researchy 2024

Intrinsic Attributes	%	First Word	%
Knowledge-Intensive	79.8	How	41.0
Multi-Faceted	75.7	Why	25.5
Reasoning-Intensive	75.5	What	15.0
Subjective	48.5	Is	5.2
Assumptive	25.7	Should	2.2

Table 8.2: TREC-RAGgy 2024 and TREC-Researchy 2024 topic distribution; top-5 categories in topic classification for TREC-RAGgy 2024 (each topic classified into a single category), intrinsic attributes for TREC-Researchy 2024, and the first word in all topics.

8.4 Topic Collection

Topics, i.e., user queries, are crucial for robust evaluation of RAG systems. Traditionally, popular retrieval and traditional QA benchmarks primarily consist of factoid queries, where answers are typically found within a single sentence or paragraph. However, these topics lack complexity, leading to short answers that can be easily memorized by LLMs. For instance, MS MARCO [29] surprisingly contains up to 55% factoid queries [34, 284]. To avoid short-form answers in RAG, we utilize two collections containing non-factoid topics covering diverse information and requiring long-form answers. We describe them below:

TREC-RAGgy 2024. We develop TREC-RAGgy 2024, a collection with topics filtered from TREC Deep Learning 2021-2023 tracks [75, 76, 78], based on topic category and generated-answer classification. We classify each available topic into seven categories taken from CRAG [384] and filter out a subset of topics that either have a long-form answer or require information aggregation across multiple sources of information. Out of the 210 original topics available, we filter and include 120 topics (57.1%) in the TREC-RAGgy 2024 collection. From Table 8.2, we observe 24.2% of the topics included are "aggregation", indicating RAG systems should chain information through entities and set relations from multiple segments (e.g., example of what a family advocate does) to generate an accurate long-form answer. Similarly, 65% of the topics start with "what" or "how" questions.

The TREC-RAGgy 2024 collection includes mapped document-level relevance judgments from previous TREC Deep Learning tracks. These relevance judgments, originally associated with the full documents, have been mapped to the new deduplicated corpus, enabling effective evaluation of the retrieval systems. However, since most of our work operates at the segment level, prior to the evaluation, we perform a MaxP step — leveraging the score of the most relevant segment as the representative score of the document.

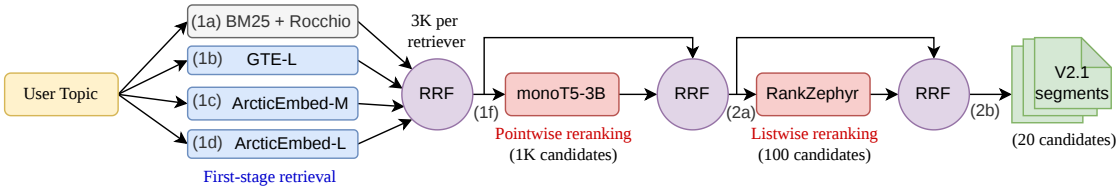


Figure 8.2: An illustration of the retrieval and reranking pipeline in Ragnarök.

TREC-Researchy 2024. Researchy Questions, introduced in Rosset et al. [284], contains 102k non-factoid topics with long-form answers. These topics were curated from Bing Search logs and evaluated by GPT-4 on a scale of 0–10 based on eight intrinsic attributes, such as subjectivity and multifacetedness. Notably, unlike TREC-RAGgy 2024, these queries lack relevance judgments. To curate a smaller development subset for a faster evaluation of RAG systems, we employ a sampler designed to maximize diversity based on the eight intrinsic attributes. This is achieved by iteratively selecting the query with the highest l_1 norm in the intrinsic attribute space (of all eight dimensions) relative to the already-sampled set. We refer to the resultant topic set as TREC-Researchy 2024. From Table 8.2, about 80% of the topics are Knowledge-Intensive, and about 76% are Multi-Faceted, highlighting the need for effective RAG systems. Additionally, 66.5% of topics start with "how" or "why", emphasizing explanatory questions. These distributions suggest that TREC-Researchy 2024 prioritizes complex and multi-dimensional topics.

Answer to RQ2: *We deduplicate and segment the MS MARCO V2 document collection to produce the MS MARCO V2.1 segment collection. For topics, we manually curate TREC DL 2021-2023 to form TREC-RAGgy 2024, filter questions with a potential long-form answer, and incorporate a sampling technique to maximize diversity based on eight intristic attributes in TREC-Researchy 2024.*

8.5 TREC RAG Baselines

8.5.1 Retrieval

Our retrieval module integrates both first-stage retrievers and rerankers. We provide an illustration of the pipeline used in the retrieval module in Figure 8.2.

Lexical Retrieval. For traditional lexical retrievers, we employ BM25 and BM25 + Rocchio, available in Anserini [383] and retrieve the top 3000 segments for a given topic. BM25 is

proved to be effective as a first-stage retriever due to its capability to capture lexical overlap, with Rocchio introducing relevance feedback [120] to further refine the search results based on initial BM25 retrieval.

Dense retrieval. We further extend our retrieval evaluation by incorporating GTE-L, a dense retrieval model from Alibaba-NLP [186]. GTE-L is their larger variant comprising 434M parameters, with embeddings of dimensionality 1024, and support for a maximum sequence length of 8196 tokens. This model is suited for long-context retrieval tasks, enabling it to effectively capture relationships across larger text segments. GTE-L is a strong baseline demonstrating state-of-the-art effectiveness in the MTEB [232] benchmark, when it was introduced. Additionally, we evaluate two other dense retrieval models from Snowflake: ArcticEmbed-M and ArcticEmbed-L [222, 388]. ArcticEmbed-M is a 137M parameter model, generating embeddings with a dimensionality of 768, while ArcticEmbed-L is a larger model with 335M parameters and an embedding dimensionality of 1024. Both models have shown state-of-the-art performance in retrieval tasks, particularly on the BEIR benchmark [329], where they excel in diverse retrieval settings. ArcticEmbed-M and ArcticEmbed-L provide efficient and scalable solutions for semantic retrieval, complementing our first-stage retrieval.

Reciprocal Rank Fusion (RRF). For each retrieval method (BM25 + Rocchio, GTE-L, ArcticEmbed-M, and ArcticEmbed-L), we retrieve the top 3000 segments. To leverage the strengths of multiple retrieval strategies, we employ Reciprocal Rank Fusion (RRF) as a hybrid approach. The RRF technique fuses the retrieval results from BM25 + Rocchio, GTE-L, ArcticEmbed-M, and ArcticEmbed-L, with the aim of producing a better candidate set for reranking, as hybrid approaches improve retrieval effectiveness [170].

Reranking. Following the RRF fusion, we explore neural reranking models for first-stage reranking. We leverage monoT5-3B, a pointwise reranker with 3B parameters, that demonstrates strong effectiveness in reranking by assigning relevance scores to individual document-query pairs [240]. Note that we fuse these results with the results from the prior stage. Next, we incorporate RankZephyr, a highly effective listwise reranker in Pradeep et al. [257], which takes as input a query and a list of passages and outputs a reordered list based on relevance. RankZephyr is particularly effective in scenarios where contextual relationships between documents are useful for ideal reranking. In our pipeline, we use RankZephyr to rerank the top 100 resultant candidates from monoT5-3B. Note that we fuse the RankZephyr results with the results from the prior stage.

Both monoT5-3B and RankZephyr and also others such as LRL [212], RankGPT [320] and LiT5 [322], are available through the `rank_llm` package, which we leverage as part of our reranking pipeline in Ragnarök. While budget constraints prevent us from testing every

Model	nDCG@10	MAP@100	Recall@100
Lexical & Dense Retrievers			
(1a) BM25	0.4227	0.1561	0.2807
(1b) BM25+Rocchio	0.4188	0.1818	0.3141
(1c) GTE-L	0.5682	0.2162	0.3512
(1d) ArcticEmbed-M	0.5749	0.2349	0.3692
(1e) ArcticEmbed-L	0.5776	0.2277	0.3623
(1f) RRF(1b,1c,1d,1e)	0.6064	0.2592	0.3990
Rerankers			
(2a) RRF(1f, monoT5-3B)	0.6175	0.2708	0.4208
(2b) RRF(2a, RankZephyr)	**0.6357**	**0.2770**	**0.4208**

Table 8.3: Results on the Document Ranking Task of the TREC-RAGgy 2024.

reranking model, support for this functionality continues through Ragnarök. Finally, the top-20 reranked segments are passed on to the next stage, i.e., augmented generation.

Design Choices. While the overall architecture in Figure 8.2 follows a standard multi-stage retrieval and reranking design, we emphasize several key design choices. For fusion, we adopt Reciprocal Rank Fusion (RRF) due to its effectiveness in combining retrieval scores without requiring score normalization, as it performed better than normalizing each score and summing them up. Similarly, our reranking strategy incorporates two rounds, with pointwise and listwise reranking models, which is used to provide the optimal subset of ranked segments for a given topic. We did not consider closed-source alternatives to RankZephyr and monoT5-3B, such as RankGPT [320] or Voyage [9], in our work.

Retrieval Results. Table 8.3 presents the nDCG@10, MAP@100, and recall@100 for the primary retrieval baselines under consideration. The traditional lexical retrievers BM25 (1a) and BM25+Rocchio (1b) demonstrate comparable scores with only marginal differences. Dense retrievers (1c, 1d, 1e) exhibit superior effectiveness compared to traditional lexical methods (1a, 1b). Among the dense models, the ArcticEmbed variants (1d, 1e) achieve slightly higher effectiveness metrics than GTE-L (1c). Notably, the application of Reciprocal Rank Fusion across these models (1f) yields substantial improvements compared to any individual retriever (1a–e).

Reranker Results. The integration of the pointwise reranker monoT5-3B (2a) enhances retrieval effectiveness across all metrics. Subsequent application of listwise reranking models with RankZephyr (2b) yields additional effectiveness gains by improving precision atop the pairwise reranking with monoT5-3B.

```
System:  This is a chat between a user and an artificial intelligence assistant.
The assistant gives helpful, detailed, and polite answers to the user's questions
based on the context.  The assistant should also indicate when the answer cannot be
found in the context.

INSTRUCTION: Please give a complete answer to the question.  Cite each context
document that supports your answer within brackets [] using the IEEE format.

QUESTION: {query}

CONTEXTS:
[1] {Passage title}:  {Passage text}
[2] {Passage title}:  {Passage text}
...
[20] {Passage title}:  {Passage text}

INSTRUCTION: Please give a complete answer to the question.  Cite each context
document that supports your answer within brackets [] using the IEEE format.
```

Figure 8.3: ChatQA prompt template [203] used for RAG generation with in-text citations with LLaMA3.1-70B and GPT-4o in our Ragnarök framework.

Answer to RQ3: *We find that dense retrievers substantially outperform lexical methods, while the hybrid fusion of first-stage retrieval achieves the best results. Additionally, the multi-stage ranking pipeline, with monoT5-3B and RankZephyr further improves the scores.*

8.5.2 Augmented Generation

Our generation module is designed to support a wide range of both open-source and proprietary models, enabling flexible integration for various use cases. At the core of our open-source support is vLLM [166], an optimized framework for efficient large language model (LLM) inference. vLLM excels in handling resource-intensive models by leveraging tensor parallelism and dynamic batching, ensuring scalability and performance across distributed systems. This makes it well-suited for open-source LLMs, where efficient resource management is critical. In addition, vLLM provides a wide support for text-based LLMs including custom models fine-tuned with LoRA [131].

Our focus is on three popular LLMs: (i) GPT-4o is the latest GPT version from OpenAI [247]; (ii) Command R+ is Cohere's open-source instruction following LLM developed for complex RAG pipelines [73]; (iii) LLaMA-3.1-70B-Instruct is Meta's flagship open-source instruction tuned generative model optimized for multilingual dialogue use cases [92]. Given

that Command R+ cites in a span level, we map the citations to their parent sentences (segmented using spaCy [129]). For GPT-4o and LLaMA-3.1-70B, we follow the ChatQA prompt template [203] and generate the complete answer. During post-processing, we segment the answer into sentences using spaCy sentence segmentation [129], and cite relevant segments within the text (in-line) using the IEEE format. An example of the prompt template used for RAG generation is shown in Figure 8.3.

8.5.3 RAG-Bench Evaluation

Evaluating different RAG answers is challenging as multiple factors within the output response are crucial for effectiveness evaluation. To combat this, recent works rely on an LLM-as-a-judge setup [403], where strong LLM assessors judge the RAG-generated output in a pairwise evaluation style (side-by-side) in a head-on tournament. In our work, we briefly evaluate three baseline techniques: Command R+, GPT-4o and LLaMA3.1-70B using human evaluators (the authors from the original publication).

Answer to RQ4: *The Command R+ baseline outputs shorter answers and cites more relevant segments, whereas, the GPT-4o baseline outputs longer and more detailed answers and cites fewer segments. Therefore, for topics in both TREC-RAGgy 2024 and TREC-Researchy 2024, GPT-4o intuitively is the better choice for RAG answer generation. LLaMA3.1-70B model produces long and detailed answers, but occasionally generates a references section (from internal memory), from which it cites.*

8.6 Ragnarök System Arena

Heavily inspired by the success of Chatbot Arena [64, 403], a crowdsourcing benchmark WebUI featuring anonymous battles, we extend the concept to multi-stage configurable RAG pipelines with Ragnarök. In the arena, users interact with two unblinded/blinded RAG systems simultaneously, issuing the same topic to both. The participants evaluate and select the pipeline that delivers their most preferred response, with the identities of the modules in the end-to-end pipeline revealed after the voting process in the blinded case. We leverage Gradio [3] to build the WebUI for Ragnarök. Each step of the pipeline uses REST APIs for intercommunication, enabling easy module switching within the pipeline. This modular design simplifies the integration of different retrieval and LLM configurations, enhancing scalability and maintainability.

Figure 8.4 illustrates an example topic "what inspired pink floyd's the wall?" processed by two different pipelines: Pipeline A, comprising BM25 → RankZephyr → GPT-4o (left), and

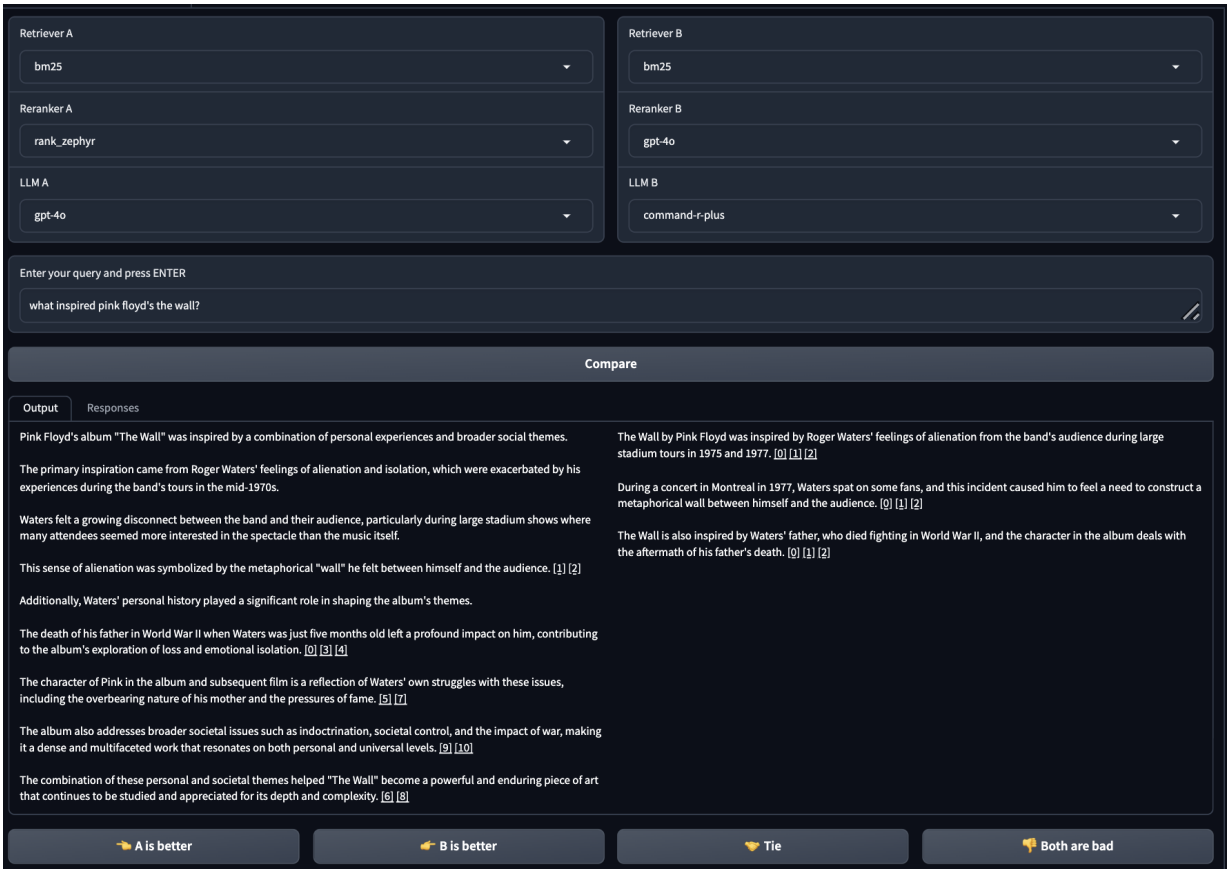


Figure 8.4: WebUI showcasing the Ragnarök System Arena and the user query, "what inspired pink floyd's the wall?", with answers from two pipelines side-by-side comparing GPT-4o (left) and Command R+ (right).

Pipeline B, comprising BM25 → RankGPT-4o → Command R+ (right) in the unblinded tab. The outputs generated by each pipeline are compared, allowing users to discern which system provided a more satisfactory answer. Note that when the user hovers the mouse over a citation, they can preview the cited segment. We also provide a similar blinded pairwise evaluation interface along with the ability to view the responses in the JSON form (a tab) in the WebUI for Ragnarök.

After reading both outputs, the user has to pick one of the four choices for preference scoring: (i) A is better (ii) B is better (iii) Tie, or (iv) Both are bad. These results from preference scoring are organized into three separate ELO leaderboards, leveraging an SQLite database to track module effectiveness. These leaderboards evaluate language models

(LLMs) for augmented generation-only (AG), retrieval models (R), and retrieval-augmented generation pipelines (RAG), providing a comprehensive view of their contributions.

8.7 Summary

The emergence of retrieval-augmented generation (RAG) has revolutionized modern search systems by allowing real-time data incorporation into large language models (LLMs). This chapter presented Ragnarök, a reusable and open-source end-to-end framework designed to provide reproducible baselines and a WebUI that served as a RAG battle arena for the retriever, reranker, and generation models.

In addition to introducing the MS MARCO V2.1 collection, we carefully curated topics from TREC-DL 2021-2023 and Researchy Questions. The TREC-RAGgy 2024 subset and the corpus can be collectively viewed as an ad hoc retrieval collection on which we evaluated several state-of-the-art dense retrievers and rerankers. Our work also defines I/O specifications to assist users in the RAG paradigm. By tightly integrating with popular retrieval frameworks such as Pyserini and `rank_llm`, Ragnarök ensures a seamless usage.

We identified key baselines from industry, including OpenAI's GPT-4o, Cohere's Command R+ and Meta's LLaMA3.1-70B, and provided a qualitative analysis of these baselines on the development topics. Our framework also provided a WebUI for head-to-head RAG system comparisons, inspired by recent work such as Chatbot Arena, to facilitate user evaluation and preference scoring. Finally, by open-sourcing Ragnarök, we aim to standardize RAG applications in preparation for TREC RAG, promoting wider adoption and fostering innovation in the RAG research community.

Chapter 9

Support Assessment in Retrieval-Augmented Generation

The shift from document retrieval to retrieval-augmented generation, presents a unique challenge in terms of evaluation. There is no standardized framework used for RAG evaluation and traditional retrieval metrics meant for document ranking are ill-suited for long-form response evaluation. As experimented briefly in Chapter 8, human evaluation remains the gold standard. However, the procedure is not feasible, as it is expensive and time-consuming to evaluate responses at scale.

Document-level citations for supporting facts in LLM-generated answers are integral to any deployed RAG system. Therefore, support evaluation assesses whether a RAG answer factually supports the information present in the cited documents, which is crucial for evaluating the quality of a RAG system.

In this chapter we focus exclusively on support evaluation, i.e., whether the information in an answer sentence is supported by the cited documents, which we consider as the "ground truth". We provide an automatic judgment process using an LLM judge, and compare it against human judgments, validated at scale in the TREC 2024 RAG Track [327], showing promise that LLMs can be effectively used to provide reliable support assessments. We ask the following research questions in our work:

RQ1 *How does LLM fare as judges, in contrast to human judges for leaderboard ranking?*
RQ2 *Do LLM judges exhibit certain biases towards a particular choice of labels?*
RQ3 *What types of errors are observed in support assessment by LLMs and human judges?*

Topic: how taylor swift's age affects her relationships

Answer: Taylor Swift's age has significantly influenced her relationships, often contributing to both the dynamics and outcomes of her romances [ ]. Swift's relationships have frequently involved notable age gaps, which have sometimes led to complications [ ]. For instance, her relationship with John Mayer, who was 11 years her senior, reportedly strained due to the age difference and Mayer's reputation as a playboy, leading to Swift feeling taken advantage of [0, 3, 4]. This relationship inspired her song "Dear John," reflecting her emotional turmoil [3, 4]. Similarly, Swift's relationship with Jake Gyllenhaal, who was nine years older, also highlighted the impact of age differences. Gyllenhaal reportedly felt the age gap, and Swift's inexperience made it difficult for her to navigate the complexities of the relationship, leading to heartbreak and inspiring several songs on her "Red" album [8, 1, 2]. [...] In summary, Taylor Swift's age has played a crucial role in shaping her relationships, influencing both the dynamics and outcomes [8, 1].

Passage [3]: A Timeline Of Taylor Swift's Age-Inappropriate Romances | Business Insider [...] Taylor Swift, 21, & John Mayer, 32. The inappropriateness of Swift's dating habits peaked when she started seeing 11-years her senior and notorious playboy, John Mayer. Neither party confirmed the romance, but they reportedly dated through February 2010 [...]

Passage [8]: What Really Went Wrong With Jake Gyllenhaal And Taylor Swift? [...] Taylor Swift and Jake Gyllenhaal dated from October to December 2010. [...] What went wrong with these two "*twin fire signs?*" It turns out, a lot. Maybe that's why it's such a touchy subject for him these days! The age gap, when they dated, Gyllenhaal was 29 to Swift's 20, and by the time they split, he was 30 to her 21. "*He said he could feel the age difference*," a source told Us Weekly [...]

Table 9.1: A sample RAG answer segmented into multiple sentences with citations to passages retrieved from the MS MARCO V2.1 segment collection for a Taylor Swift query. We highlight the cited passages, provided as a list (that can be empty, i.e., zero citations).

9.1 The TREC 2024 RAG Track

The Text Retrieval Conference (TREC) led the way in many aspects of evaluation in information retrieval (IR), natural language processing (NLP), and beyond, for accelerating research within the community (both researchers and practitioners). Each year, TREC organizes several tracks, focused on topics ranging from text or multimodal retrieval [76, 78, 382] to conversational QA [12].

The context for this work is the TREC 2024 RAG Track [327, 258], which is divided into three tasks: retrieval (R), augmented generation (AG), and retrieval-augmented generation (RAG). Here, we focus on the generation part, i.e., participant systems are presented with 301 non-factoid queries (called topics in TREC parlance) gathered using a fresh scrape of Bing Search logs. Queries are multifaceted and subjective, warranting RAG systems to provide long-form answers [258, 284], and retrieve candidate passages between 500-1000 characters from the MS MARCO V2.1 Segment collection, which we introduce in Ragnarök [258] (from Chapter 8).

These candidate passages are either generated by us (i.e., the track organizers) and shared with all the participants [258] (the AG task) or each participant can directly perform retrieval from the MS MARCO V2.1 segment collection (the end-to-end RAG task). The

candidate passages provide the context or grounding of RAG for synthesizing the final free-form answer. We require that answers be segmented into sentences (e.g., spaCy sentence segmentation [129]), and that each sentence is associated with citations to passages from the corpus, as shown in Table 9.1. We follow the output format defined in Ragnarök (Chapter 8), and therefore do not consider citation support at the variable unit of a span. For queries in TREC 2024 RAG Track, we empirically observe that an answer sentence majorly makes a single claim, and judging them at the sentence-level is sufficient for evaluation in this work.

9.2 Support Assessment

Consistent with prior support evaluations in RAG [199, 113], we use a three-level grading scheme with the following descriptions [1] for each support level:

FS **Full Support**: All of the information in the answer sentence is factually consistent with and supported by the cited passage.

PS **Partial Support**: Some of the information in the answer sentence is factually consistent with and supported by the cited passage, but other parts of the sentence are not supported.

NS **No Support**: The cited passage is completely irrelevant and does not support any part of the answer sentence.

Next, to evaluate the quality of LLM judges in comparison to human judges, we conduct our support assessment with human judges under two conditions: (1) manual from scratch and (2) manual with post-editing. We describe both conditions in detail below:

1. **Manual from scratch.** A human judge is provided with the answer sentence and the cited passage, and assigns one of the labels described above (FS, PS or NS).
2. **Manual with post-editing.** This condition is the same as above, except that the human judge is additionally provided with the label from the LLM judge.

For automatic labeling, we use GPT-4o as the automatic judge. We run inference using the Microsoft Azure API [247], providing a single passage at a time in the prompt[2] together with the answer sentence and the cited passage. The GPT-4o judge is presented with each

[1] An edge case is a sentence with zero citations: We automatically consider the support assessment to be "no support", as the sentence does not cite any retrieved passage.

[2] Separately, we experiment with providing multiple cited passages at once, but we anecdotally observe that providing one passage at a time performs better with the GPT-4o judge.

In this task, you will evaluate whether each statement is supported by its corresponding citations. Note that the system responses may appear very fluent and well-formed, but contain slight inaccuracies that are not easy to discern at first glance. Pay close attention to the text.

You will be provided with a statement and its corresponding passage which the statement cites. It may be helpful to ask yourself whether it is accurate to say "*according to the citation ...*" with the statement following this phrase. Be sure to check all of the information in the statement. You will be given three options:

• **Full Support**: All of the information in the statement is supported in the citation.
• **Partial Support**: Some parts of the information in the statement are supported, but other parts are missing in the citation.
• **No Support**: The citation does not support any part of the statement.

Please provide your response based on the information in the citation. If you are unsure, use your best judgment. Respond "Full Support", "Partial Support", or "No Support" with no additional information.
Statement: {`statement`}
Citation: {`citation`}

Figure 9.1: Pairwise prompt template used by the GPT-4o judge for support evaluation.

sentence and its cited passage and is asked to determine the support label without any explanation (full support, partial support, no support). The prompt is shown in Figure 9.1.

9.3 Computational Cost & Evaluation Tradeoffs

In TREC 2024 RAG Track, we allow participants to provide citations for up to 20 passages per answer sentence. To judge each sentence and its cited passage, our protocol requires a human judge to read the answer sentence and a relatively long text passage (typically, 500–1000 characters). Thus, conducting an exhaustive evaluation of all citations per answer sentence across multiple participants is not feasible due to budget constraints.

Therefore, we choose between sparse and dense annotations. Dense annotations provide fewer judged topics, but each answer sentence would be evaluated against k cited passages. On the other hand, sparse annotations provide higher diversity in judged topics, but at the cost of judging fewer cited passages for each answer sentence.

We opt for sparse annotations to achieve more judged topics. We fix both the human and GPT-4o judges to evaluate only the first cited passage of each answer sentence for

all participants. Using this setup, we were able to judge 36 topics in total with human assessments. As with all TREC evaluations, NIST provides the resources to perform human evaluations based on the guidance of the track organizers (i.e., us). NIST first trains each human judge to understand the task, and then each judge evaluates the topics sequentially.

9.4 Support Evaluation Metrics

Support is evaluated along two dimensions, following Liu et al. [199]: (1) *weighted precision*, which accounts for how many correct passage citations appear in the generated answer, and (2) *weighted recall*, which accounts for how many sentences in the answer are supported by passage citations. We define both metrics below:

Weighted Precision. This metric measures the weighted proportion of citations that support each answer sentence. We assign a weight to $s(a_i, d_j)$ of 1.0 to Full Support (FS), 0.5 to Partial Support (PS), and 0 to No Support (NS) for the answer sentence and cited passage. To illustrate the metric, assume a RAG answer with 3 sentences = $\{a_1, a_2, a_3\}$, and a corpus C with 2 passages: $\{p_1, p_2\}$. Suppose that passage p_1 partially supports a_1, passage p_2 fully supports a_2, and a_3 has zero citations. We compute weighted precision as follows:

$$\text{Weighted Precision} = \frac{s(a_1, p_1) + s(a_2, p_2)}{\text{count}(\{a_1, p_1\}, \{a_2, p_2\})} = \frac{0.5 + 1}{2} = 0.75$$

Weighted Recall. This metric measures the weighted proportion of answer sentences that are supported by their cited passages. We use the same weights as defined above for weighted precision. For the same example, we compute weighted recall as follows:

$$\text{Weighted recall} = \frac{s(a_1, p_1) + s(a_2, p_2)}{\text{count}(\{a_1, a_2, a_3\})} = \frac{0.5 + 1}{3} = 0.5$$

Precision penalizes *overcitation* in the answer text, which is often distracting and unnecessary for the user. On the other hand, recall penalizes answers with *undercitation*, i.e., sentences with zero citations. Therefore, if a sentence has zero citations, it reduces the recall score but leaves precision unchanged. As described above, we evaluate only the first cited passage per answer sentence for support in TREC 2024 RAG Track. Therefore, the weighted recall and precision scores are identical if all answer sentences have at least one citation. We leave for future work the question of how to best evaluate multiple passage citations for every sentence in the answer, as it would require a much larger annotation budget.

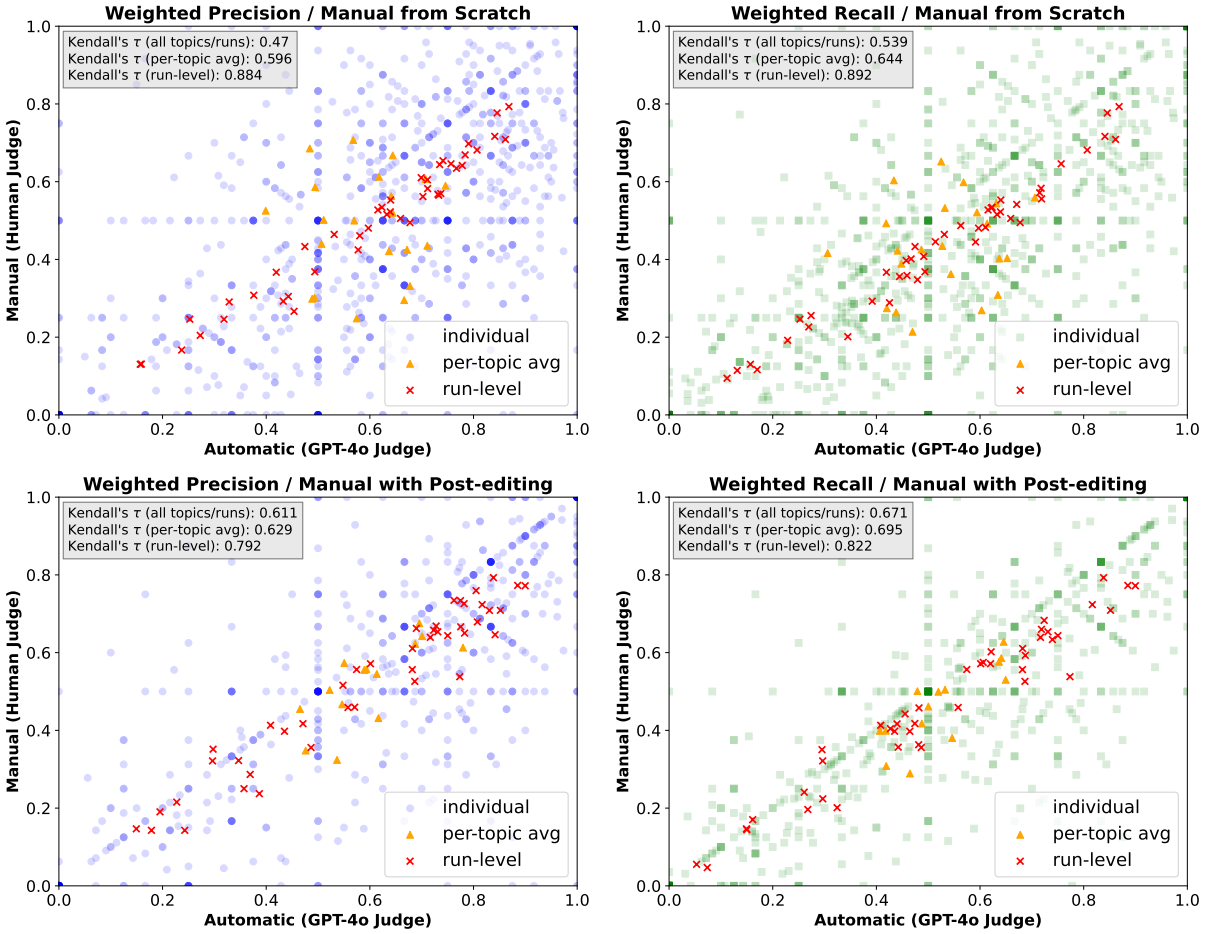


Figure 9.2: Correlations between scores from human and GPT-4o judges for the manual from-scratch condition (top) and the manual with post-editing condition (bottom), measuring weighted precision and recall. Scores observed exactly on the diagonal indicate a perfect overlap (Kendall's $\tau = 1$) between the human and GPT-4o judge score. Red markers show run-level scores, yellow triangles show per-topic averages, and blue dots or green boxes show all individual topic/run combinations. Each plot is annotated with rank correlations showing Kendall's τ.

	Condition	#Topics	#Annotations	Support level		
				FS	PS	NS
(1a)	Manual from scratch (Human)	22	6,742	2,752	1,652	2,338
(1b)	Automatic (GPT-4o)	22	6,742	3,110	2,421	1,211
(2a)	Manual with post-editing (Human)	14	4,165	1,812	1,076	1,277
(2b)	Automatic (GPT-4o)	14	4,165	2,045	1,330	790

Table 9.2: Descriptive statistics for support judgments for the (1) manual from-scratch and (2) manual with post-editing conditions for 45 participant submissions on 36 topics.

9.5 Experimental Results

For TREC 2024 RAG Track, NIST receives 93 runs from 20 groups for the end-to-end RAG task and 53 runs from 11 groups for the AG task. Given resource constraints, we evaluate only the two highest-priority submissions from each group across the RAG and AG tasks. As shown in Table 9.2, this yields 31 runs from 18 groups for RAG and 14 runs from 9 groups for AG, for a total of 45 participant submissions for each topic. The human judges were able to complete judgments for 36 topics, sparsely annotated: 6,742 annotations on 22 topics in the manual from-scratch condition and 4,165 annotations on 14 topics in the manual with post-editing condition.

9.5.1 Weighted Precision and Recall

Figure 9.2 shows scatter plots of weighted precision and recall scores obtained by all participant submissions. The scores directly on or nearby the diagonal indicate both human and GPT-4o judge provide a similar weighted precision or recall score. The scores are visibly more scattered in comparison to nuggets [260] or retrieval [340], as we judge support at the sentence-level, which introduces more variation in the evaluation scores. Run-level scores (denoted by $\times$) are strongly correlated (all above 0.79 Kendall's τ) between GPT-4o and human annotations. Per-topic averages (denoted by $\triangle$) vary on both axes, where certain topics achieve a higher weighted precision and recall score than humans over GPT-4o, and vice versa. Individual participant scores (denoted by $\bigcirc$ or $\square$) show a high variance in both weighted precision and recall scores. This is likely due to the mismatch of human annotators preferring "no support", whereas GPT-4o prefers "partial support". Overall, we observe the majority of scores in the bottom right triangle, indicating that humans take a more conservative approach and provide lower levels of support overall than GPT-4o, leading to lower weighted precision and recall scores.

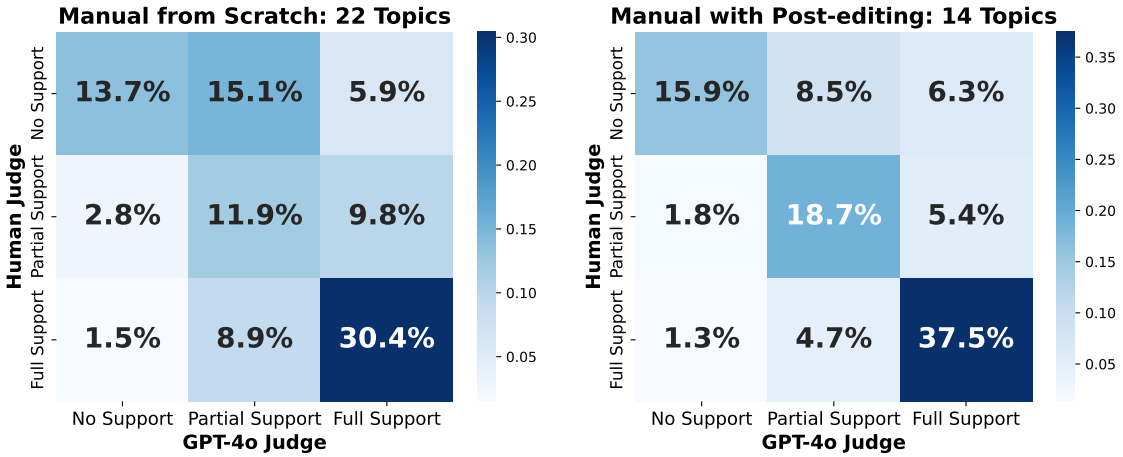


Figure 9.3: Confusion matrices comparing predictions from human and GPT-4o judges for the manual from-scratch (left) and the manual with post-editing (right) conditions.

Answer to RQ1: *We observe a strong run-level correlation of up to 0.892 Kendall's τ between GPT-4o and human annotations, indicating that GPT-4o can be used to overall assess participant answers reliably on all topics. Individual participant scores are scattered, as weighted precision and recall metrics are rather sensitive to sentence-level computations.*

9.5.2 Confusion Matrices

Next, to better understand how often the GPT-4o judge agrees with the human judges, we plot the confusion matrices in Figure 9.3. We compare predictions by human judges with GPT-4o on two conditions: manual from scratch and manual with post-editing.

Manual from-scratch Condition. For 56% (13.7% + 11.9% + 30.4%), GPT-4o and the human judge perfectly agree on their support judgment on 22 topics. Both "full support" and "no support" categories have higher percentages (30.4% and 13.7%), showing that humans and GPT-4o as judges agree more on both ends of the spectrum. For 15.1%, the GPT-4o judge assigns "partial support", which the human judge assigns "no support". Finally, GPT-4o is more likely to assign a higher support label than the human judge (the upper right triangle has a higher combined percentage over the lower left triangle).

Manual with post-editing Condition. Compared to the previous condition, the rate of perfect agreement increases to 72.1% (15.9% + 18.7% + 37.5%) on 14 topics annotated with post-editing GPT-4o labels. This shows that sentences and cited passages with "partial

Cohen's Kappa	From scratch		With post-editing	
	GPT-4o	Human	GPT-4o	Human
Author (independently)	**0.29**	−0.03	**0.27**	0.07
LLAMA-3.1-405B	**0.60**	−0.20	**0.46**	−0.06

Figure 9.4: Inter-annotator agreement score (Cohen's κ) for our unbiased study on disagreements between GPT-4o and human (NIST) annotators.

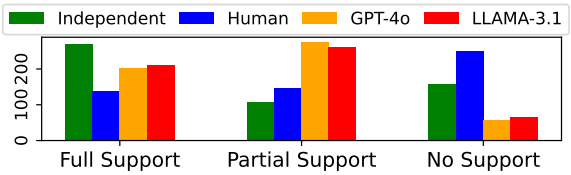


Figure 9.5: Support label prediction by different judges for each support category (FS, PS, NS) in the disagreement analysis on 537 sentence–passage pairs.

support" that led to disagreements in the manual from-scratch condition, decrease in this condition. Human judges are more likely to agree with the GPT-4o judge unless it is an obvious mistake, i.e., when the GPT-4o judge considers an annotation to be "full support" and the human judge considers it to be "no support" (increased now to 6.3% from 5.9% in the manual from-scratch condition).

Answer to RQ2: *We make two interesting observations. First, human judges are decisive (either full or no support) with only 22–26% of judgments labeled as partial support. Second, GPT-4o exhibits a bias towards positive (partial/full) support with only 18–19% of judgments labeled as no support.*

9.6 Annotator Disagreements

In the experiments reported in Section 9.5, we observe frequent disagreements between human judges and GPT-4o. To study this further, we perform unbiased annotations from scratch by carefully re-assessing a randomly sample of disagreements between the human and GPT-4o judge, using an independent human judge (the author of this thesis) and another LLM judge, LLAMA-3.1-405B [92] (with the same prompt as shown in Figure 9.1). We randomly sample 15 disagreement pairs per topic, re-evaluating 537 sentences and their first cited passages, including both assessment conditions: (1) manual from scratch and (2) manual with post-editing.

Disagreement Results. As shown in Figure 9.4, we interestingly find the author of this thesis as the assessor to be *better correlated* with GPT-4o than the human judge provided by NIST (Cohen's κ of 0.29 and 0.27 versus −0.03 and 0.07) in the manual from-scratch condition. The author of this thesis fully matches 31% of the time with the human judge and 51% of the time with the GPT-4o judge. Similarly, in the manual with post-editing

Answer: Her first Hollywood romance with Joe Jonas was age-appropriate, as both were 19 at the time.	**Answer:** Swift's youth and inexperience have been cited as factors in her relationships, with some partners reportedly taking advantage of her naivety.
Passage ID [3]: `doc_04_1081579649#2_2253244363` **Title:** Timeline of Taylor Swift's Relationships **Text:** [...] 2008: Taylor Swift, 19 & Joe Jonas, 19. Swift's first Hollywood romance was perfectly age-appropriate. While Swift was just emerging onto the pop scene, Jonas, along with his singing and dancing brothers, were at the top of their game. [...]	**Passage ID [8]:** `doc_48_481103263#4_840591332` **Title:** Taylor Swift Boyfriends: Taylor Swift's Relationships | New Idea Magazine **Text:** [...] John Mayer (December 2009-February 2010) The 11-year age gap between Taylor and John reportedly strained their relationship, with Taylor feeling that her naivety was taken advantage of. She references their ugly breakup in her song [...]
Human Judge: No Support × **GPT-4o Judge:** Full Support ✓	**Human Judge:** Full Support ✓ **GPT-4o Judge:** Partial Support ×

Table 9.3: Examples of annotation mistakes by either GPT-4o or the human judge for the topic "*how taylor swift's age affects her relationships*" taken from the disagreement analysis. The fragment of the passage that supports the answer sentence is highlighted.

condition, the author of this thesis fully matches 37% of the time with the NIST human judge and 52% of the time with the GPT-4o judge. LLAMA-3.1-405B shows a stronger correlation with another LLM (GPT-4o) than with human judges (Cohen's κ of 0.60 and 0.46 versus -0.20 and -0.06), demonstrating the likelihood of different LLMs producing similar prediction labels.

From the label distributions in the disagreement analysis shown in Figure 9.5, both LLMs (LLAMA-3.1-405B and GPT-4o) label about 49–51% of sentences and their cited passage as "partial support", whereas the human judge labels 47% of the sentences as "no support". The independent judge labels 50% of the sentences as "full support". We keep as future work explorations of consistency: why LLM judges assign only a few sentences as "no support", and similarly, why human judges assign a majority of sentences as "no support".

9.7 Qualitative Analysis

We further assess examples qualitatively to understand failure cases, for example, when a human or GPT-4o judge makes mistakes during support evaluation. In Table 9.3, we show a few examples of annotation mistakes on the Taylor Swift topic found in either the human or GPT-4o judgment. Overall, we summarize the following error types made by GPT-4o:

- GPT-4o can confuse words or phrases with similar meanings; for example, it is unable to distinguish between police and security specialists.

- GPT-4o can miss out on evaluating the whole sentence (especially information present at the end of the sentence), biasing towards the "full support" label.
- GPT-4o can label "partial support" if the theme in the answer sentence is similar, but the passage does not support any text present in the answer sentence, i.e., "no support".

In contrast, human judges make mistakes by not reading passages carefully. In some cases, answer sentences that are directly stated in the middle of or at the end of a passage, or mentioned in parts of the passage, are surprisingly overlooked by the human judge. This causes the human judge to label such cases as "no support" instead of "full support". Finally, we observe that human judges occasionally label an answer sentence as "full support" even though the passage doesn't provide any support information. We suspect that this could be due to an inherent bias relying on the human judge's memory or understanding of the topic, instead of strictly relying on the actual passage text.

Answer to RQ3: *LLM judges such as GPT-4o can confuse phrases, miss out evaluating the complete sentence, and can easily label partial support for a sentence on a similar theme. On the hand, human judges make errors by not reading the passage carefully, occasionally missing supporting information placed in the middle or bottom of the passage.*

9.8 Summary

In this Chapter, we evaluated support in RAG answers by analyzing 45 submissions across 36 topics from the TREC 2024 RAG Track in a large-scale comparative study involving both humans and LLMs as judges. We critiqued and evaluated strong LLM judges, like GPT-4o, against human annotators for support assessment. Our results show a high agreement between GPT-4o and human judgments. We observe that disagreements between humans and LLMs mainly occur for sentence–passage pairs indicating partial support, i.e., in the middle of the support evaluation spectrum.

To better understand these disagreements, we conducted an unbiased evaluation by carefully re-assessing judgments with an independent human judge and a different LLM. Interestingly, in cases of disagreements, both the independent human judge and the LLAMA-3.1 judge agreed more with the GPT-4o judge than with the human judge, providing evidence for widely divergent opinions and perhaps the veracity of using LLMs for support evaluation.

In the next chapter, we will expand RAG evaluations to the multilingual setting, by evaluating relevance assessment of multilingual-focused LLMs.

Chapter 10

Relevance Assessment in Multilingual Retrieval-Augmented Generation

In Chapter 9, we focused on support assessment in RAG, measuring the grounding of the answer with cited information. In order to generate a well-supported answer with information from cited information, the retrieved documents should be relevant in answering the user question. However, the retrieval system poses challenges in accurately retrieving relevant information when evaluated either on zero-shot domains [329] or low-resource languages [398]. The incorrect or non-relevant information contained in retrieved passages can frequently mislead the LLM to hallucinate [6, 60, 302, 387, 389]. Therefore, **relevance assessment** is a crucial metric in RAG evaluation to assess whether the LLM is able to judge the relevancy of retrieved documents, before generating the answer for the question.

In this chapter, we assess multilingual RAG and utilize the "invalid" queries, i.e., queries with no relevant passages discarded during human judgments in MIRACL [398] (as discussed earlier in Chapter 4). Using these invalid queries containing unknown answers, i.e., all top-k retrieved passages manually judged as non-relevant, we construct the NoMIRACL dataset.

Using the NoMIRACL dataset, we conduct relevance assessment of several existing powerful and multilingual-focused LLMs (*e.g.*, GPT-4, Mistral, LLAMA-3), by converting the listwise relevance judgment task into a binary classification task ("Yes, answer is present" or "I don't know" as labels). We ask the following research questions in this chapter:

RQ1 *Can we repurpose the relevance assessment task into a simpler evaluation task?*
RQ2 *How do frontier multilingual-focused LLMs perform on the relevance assessment task?*
RQ3 *Does prompt optimization help improve LLM robustness in relevance assessment?*

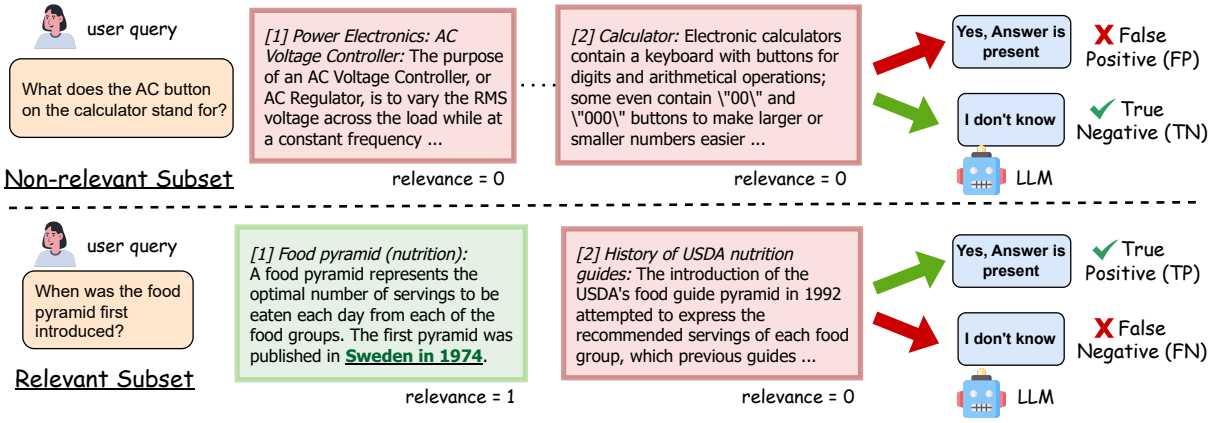


Figure 10.1: LLM robustness evaluation as a binary tree in NoMIRACL. When dealing with queries in the non-relevant subset, the LLM is expected to disregard all noisy passages and refrain from answering (TN). Conversely, for queries in the relevant subset, the LLM should recognize the relevant passage and provide a valid answer (TP).

10.1 Relevance Assessment Definition

We conduct our evaluation strategy as a contingency table to robustly evaluate the LLM behavior in both answerable and non-answerable scenarios using a binary classification task, by comparing LLM predictions against the ground truth provided by human annotators.

Definitions. NoMIRACL contains two subsets, we denote them as either non-relevant (N) or relevant (P). The non-relevant subset contains queries with no-known answers, i.e., all top-k passages are non-relevant, while the relevant subset contains queries with known answers, i.e., at least one of the top-k passages is relevant. Similarly, we denote the LLM prediction as either (P) indicating the model finds the passage relevant to answer the query and similarly negative (N) denotes the model does not find any passage relevant (i.e., containing the answer) for the query.

Confusion Matrix. In our evaluation (cf. Figure 10.1), for our non-relevant subset, True Negative (TN) denotes when the model correctly predicts queries with no-known answers using the non-relevant retrieved passages, whereas False Positive (FP) denotes when the model prediction is incorrect on the non-relevant subset. Similarly, True Positive (TP) denotes when LLM correctly predicts queries with known answers within the retrieved passages, whereas False Negative (FN) denotes when the model prediction is incorrect (i.e., the model finds no answer) on the relevant subset.

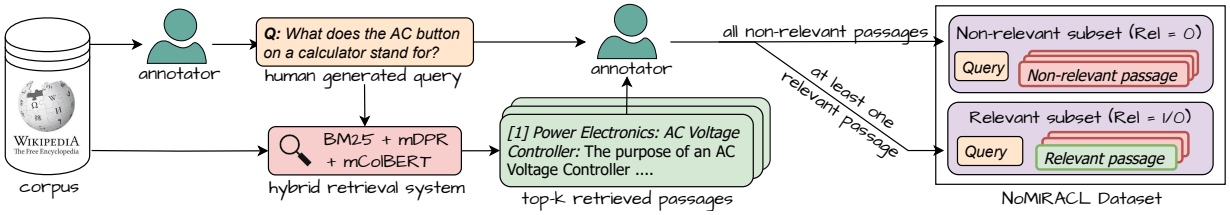


Figure 10.3: An overview of the data construction procedure involved in NoMIRACL.

Evaluation Metrics. Following prior works [6, 60], we assess LLM relevance assessment as a binary classification task using two metrics: (i) *hallucination rate* and (ii) *error rate*. First, we compute the hallucination rate (in %) = FP/(FP+TN) which measures the model's tendency to hallucinate an answer, when the answer is not available in any of the passages in the **non-relevant subset**. Next, we measure the error rate (in %) = FN/(FN+TP) which measures the model's ability to identify the relevant passage containing the answer within passages in the **relevant subset**.

Pred. / Subset	Non-Relevant	Relevant
LLM: Yes, answer is present	FP ✗	TP ✓
LLM: I don't know	TN ✓	FN ✗

Hallucination rate = FP / (FP + TN) | Error rate = FN / (FN + TP)

Figure 10.2: Confusion matrix for relevance assessment evaluation; (Subset) denotes the ground-truth; (Pred.) denotes the LLM output prediction.

Answer to RQ1: *Instead of the traditional pairwise relevance assessment, we convert the assessment into a binary classification task, emulating the augmented-generation setting. The setting is comparable to needle (here, the relevant passage) in a haystack problem (here, the set of passages) for the relevant subset and "no needle" for the non-relevant subset.*

10.2 NoMIRACL Dataset

As the goal of NoMIRACL is to understand to which extent LLMs tend to hallucinate across different languages, our dataset contains 18 diverse languages with a myriad of both correct or answerable queries (relevant) subset and hallucinated or unanswerable queries (non-relevant). We describe our dataset construction procedure in (§10.2.1) with an overview shown in Figure 10.3, and languages covered and dataset usage in (§10.2.2).

10.2.1 Data Construction Procedure

NoMIRACL is constructed using the same procedure utilized to develop MIRACL [398] (as explained earlier in Chapter 4). The data construction occurs in two stages, following [398, 71]. In the first stage, the annotator (a native language speaker) writes a well-formed query for each individual prompt text. Each prompt is a short text snippet containing the first 100 words from a language-specific Wikipedia corpus. Next, for each human-generated query, top-k passages are retrieved from the corpus using a hybrid multilingual retrieval system (more details in §10.3.1). In the second stage, annotators assess the binary relevance judgment of the top-k query–passage pairs, either relevant or non-relevant.

Non-relevant Subset. Annotators generate queries based on certain dataset guidelines, however, occasionally the human-generated queries cannot be answered with the external corpus, which leads to the scenario where none of the top-k passages are relevant, i.e., none contains the answer. These queries with unknown answers may occur due to the following reasons: (i) queries can be either generic or specific for information to be present in Wikipedia, for e.g. "*What does the AC button on a calculator stand for?*" retrieves the Wikipedia page on Calculator,[1] but it does not contain information about the AC button; (ii) spelling mistakes in query generation. We construct the non-relevant subset with queries with all top-k passages judged as non-relevant, i.e., with a relevance of 0.

Relevant Subset. Queries with known answers, i.e. at least one of top-k retrieved passages marked by the annotator as relevant to provide sufficient information to answer the query. We use the MIRACL development and test subsets [398] containing queries with at least one relevant passage, i.e., all query-passage pairs have been judged as either relevant with a relevance of 1 or non-relevant with relevance of 0.

10.2.2 Languages Covered and Dataset Usage

NoMIRACL covers 18 diverse typological languages [398].[2] The languages along with their ISO codes are: Arabic (`ar`), Bengali (`bn`), German (`de`), English (`en`), Spanish (`es`), Persian (`fa`), Finnish (`fi`), French (`fr`), Hindi (`hi`), Indonesian (`id`), Japanese (`ja`), Korean (`ko`), Russian (`ru`), Swahili (`sw`), Thai (`th`), Yoruba (`yo`), Chinese (`zh`). Overall, we observe an uneven amount of test queries present in each language. To avoid this non-uniformity and budget constraints (see subsection 10.3.3), in our experiments, we limit the evaluation to a maximum of **250 queries** for each language and subset (if available).

[1] https://en.wikipedia.org/wiki/Calculator

[2] NoMIRACL covers 10 families (from Niger-Congo to Indo-European) and 11 scripts (from Latin to Devanagari) covering diversity from the perspective of linguistic characteristics.

10.3 Experimental Setup

10.3.1 Evaluation Setup and Metrics

In NoMIRACL, we assess LLM relevance as either hallucination or error, using an input query, a prompting technique, and top-k (oracle) retrieved and relevance judged passages.

Retrieved Passages. For each query in NoMIRACL, a maximum of $k = 10$ passages are retrieved and judged by our annotators. We follow the hybrid retrieval setup in MIRACL [398], which includes three different multilingual retriever models: (i) BM25 [281], a lexical retriever, previously shown to be robust across domains and languages [329, 398]. We use the BM25 implementation available in Anserini [383] with default parameters ($k_1 = 0.9$ and $b = 0.4$) and the corresponding language-specific analyzer. (ii) mDPR [156], a dense retriever, using mBERT [86] as the backbone and fine-tuned on MS MARCO with the Tevatron toolkit [111]. (iii) mColBERT [159], a multi-vector retriever, fine-tuned following Khattab et al. [159] using mBERT as backbone and fine-tuned on MS MARCO. The top-k passages are ranked using an ensemble fusion by normalizing and averaging each model score within the range of $[0, 1]$.

Evaluation Objective. In our work, we evaluate LLM relevance as a response string y in a binary classification setup. Following prior evaluation strategies in [6, 389], we use the input query q_i, a vanilla prompt template Q, and a set of top-k annotated passages P_k. We prompt the LLM to evaluate if q_i can be answered using any passage in P_k. The LLM generates an answer output containing either $y =$ "`I don't know`" as negative (N) or "`Yes, answer is present`" as positive (P). The output is tagged "`Invalid`" if it does not fall in either one of the above. Recall from (§10.1), we calculate the *hallucination rate* (in %) = FP/(FP + TN), which measures the error in rejecting information from non-relevant passages and the *error rate* (in %) = FN/(FN + TP), which measures the error in identifying relevant passages amongst noisy ones.

10.3.2 Evaluation Models

We evaluate eleven state-of-the-art LLMs with a strong focus of multilingual instruction capabilities, including both open and closed-sourced.

(1) OpenAI: We include three closed-book LLM variants: GPT-3.5-turbo, GPT-4, and GPT-4o [246] using the Azure OpenAI service. **(2) Mistral:** We include two variants: (i) `Mistral-7B-Instruct-v0.3`, the latest 7B instruction-tuned parameter model [145] and (ii)

`Mixtral-8x7B-v0.1`, a sparse Mixture-of-Expert (MoE) model [146]. **(3) Orca-2:** In the Orca-2 series [226], we include both `Orca-2-7B` and `Orca-2-13B`. **(4) Aya**: `Aya-101` [342] is a recently introduced multilingual LLM containing 13B parameters and trained with 101 languages and `Aya-23-35B` finetuned across 23 languages [22]. **(5) LLAMA-2:** In the LLAMA-2 series [334], we include three chat variants: `Llama-2-7b-chat-hf`, `Llama-2-13b-chat-hf`, and `Llama-2-70b-chat-hf` instruction tuned chat models. **(6) LLAMA-3:** Following up on the LLAMA-2 series, we include both the instruction tuned models [92]: `Meta-Llama-3-8B-Instruct` and `Meta-Llama-3-70B-Instruct`.

10.3.3 Experimental Settings

We execute the generation of GPT-4o, GPT-4, and GPT-3.5-turbo, using the OpenAI service (API version `2023-05-15`) deployed on Microsoft Azure[3] and LLAMA-3 series using the AnyScale API service. We maintain a maximum input sequence length of 4096 tokens for a fair evaluation amongst all models. We set a low-temperature score = 0.1 for a deterministic output, and a top-p sampling ratio = 0.95. We output 50 maximum tokens.

Vanilla Prompting. The choice of prompt significantly influences the performance and LLMs have been shown brittle to prompting variations, training examples, or long context setups [200]. In our work, we evaluate all baselines using a *zero-shot monolingual listwise* prompting strategy. We construct a vanilla prompt template using all top-k (oracle) passages (available in NoMIRACL) as a list of contexts along with the input query, both in the same language. We evaluate all LLMs zero-shot, as we cannot fit few-shot exemplars due to insufficient context length (maximum of 4096 sequence length). Our template provides a short description in English describing the task [7]. Our vanilla prompt template used in our experiments is shown in black (cf. Figure 10.7).

Reducing Costs. Running GPT-4 is expensive and LLAMA-3-70B is rather slow at inference. For long contexts and low-resource languages, the costs can even multiply. To limit, we did not exceed our prompt above 4096 tokens. To effectively fit all $k = 10$ passages within the vanilla zero-shot prompt, we truncate each passage to use the first 375 tokens. Next, due to budget constraints, we keep our evaluation to a maximum of 250 randomly sampled queries for all languages in both NoMIRACL relevant and non-relevant split. We end up providing $\approx$ 20K API calls producing an expense of $1,474 (in USD) including miscellaneous costs.

[3]We compared Azure API with OpenAI API across four languages in NoMIRACL and observed no noticeable difference between different GPT-4 API version providers.

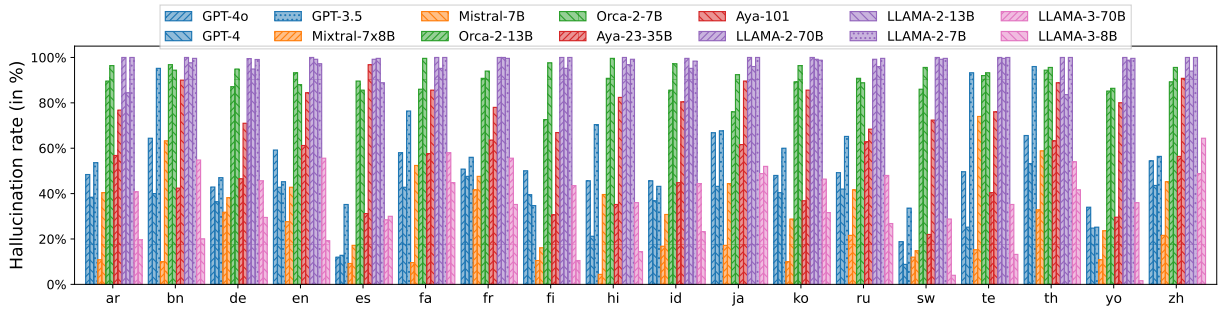


Figure 10.4: Hallucination rate (in %) = FP/(FP + TN) on the non-relevant subset (F) in NoMIRACL. Lower the hallucination rate is better. Models are sorted in terms of parameter size or recency.

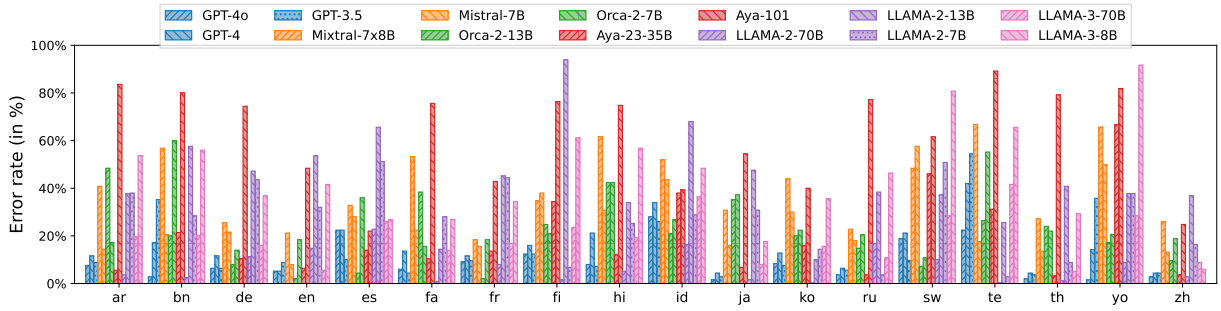


Figure 10.5: Error rate (in %) = FN/(FN + TP) on the relevant subset (T) in NoMIRACL. Lower the error rate is better. Models are sorted in terms of parameter size or recency.

10.4 Experimental Results

We discuss our LLM robustness evaluation results using the hallucination rate on the non-relevant subset in (§10.4.1) and using the error rate on the relevant subset in (§10.4.2), and compare overall both the relevant and non-relevant capacities in (§10.4.3).

10.4.1 NoMIRACL Non-relevant Subset

Figure 10.4 shows hallucination rates on the NoMIRACL non-relevant subset for a maximum of 250 queries evaluated (each language) on all 18 languages for best LLM in each category. Our findings indicate that all LLMs (except Mistral) hallucinate that an answer is present across all languages, thereby indicating their poor ability to abstain from answering. On

average, the lowest hallucination rate of 17.4% is observed by Mixtral-7x8B, followed by LLAMA-3-8B-Instruct with 26.8%. GPT-4 achieves a 35.5% hallucination rate, which highlights the challenge of identifying non-relevant passages. LLAMA-2, Orca-2, and Aya-101 perform much worse on average across all languages, by achieving a hallucination rate of more than 80%. We hypothesize that LLMs perform poorly to identify non-relevant passages as they are highly similar to the query, but do not contain the exact answer.

Overall, the lowest hallucination rates are observed in Swahili and Yoruba. We hypothesize that queries in low-resource languages (smaller Wikipedia corpus) contain retrieved information, likely to be non-relevant (easier negative), thereby making it easier for the LLM to judge as "I don't know". GPT-3.5 is observed with the highest deviation across languages with a hallucination rate as low as 25.2% on Swahili (`sw`) to 95.2% on Bengali (`bn`). Overall, all LLMs are found to perform poorly on NoMIRACL non-relevant subset, indicating our dataset is very challenging in robustness evaluation for LLMs.

10.4.2 NoMIRACL Relevant Subset

Figure 10.5 shows error rates on NoMIRACL relevant subset for a maximum of 250 queries (each language) on 18 languages for best LLM in each category Our findings indicate that all LLMs (except Mistral) identify the answer present within the relevant passage. On average, Aya-101 achieves the highest error rate of 62.5%. The lowest error rates are observed by LLAMA-2-70B and GPT-4o which are lower than 10%. Overall, Aya-101, Mixtral-7x8B, and LLAMA-3-8B perform worse on average by observing more than a 40% error rate. Overall, LLMs (except Mistral and Aya-101) perform well and do not suffer from errors in identifying answers in the NoMIRACL relevant subset.

10.4.3 NoMIRACL Overall Comparison

A robust LLM should be able to identify the answer captured within retrieved passages in the relevant subset and abstain from answering when none of the retrieved passages contain the answer in the non-relevant subset. To measure performance across both dimensions in NoMIRACL, we plot the average model accuracy across both the non-relevant (x-axis) and relevant subset (y-axis) for all tested models in Figure 10.6. Overall, LLMs positioned in the top-right corner provide an optimal performance on both subsets. A majority of LLMs (such as LLAMA-2, and Orca-2) in the top-left corner perform well on the relevant subset, however, hallucinate and struggle to perform well on the non-relevant subset, indicating their inability to accurately judge non-relevant passages.

On the other hand, Mistral and LLAMA-3 suffer less from hallucination on the non-relevant subset but observe a higher error rate (over 40%) on the relevant subset, indicating they are not confident in identifying passages containing the answer. Aya-101 is unable to perform well in either of the subsets. GPT-4 provides a good tradeoff by providing a low hallucination and error rate on NoMIRACL relevant and non-relevant subsets, however is expensive at scale for inference.

Answer to RQ2: *Proprietary LLMs such as GPT-4, perform optimally on both subsets in NoMIRACL. A majority of multilingual LLMs (Orca-2, LLAMA-2, Aya etc.) observe a high hallucination rate, and Mistral/Mixtral and LLAMA-3, on the other hand, observe a higher error rate.*

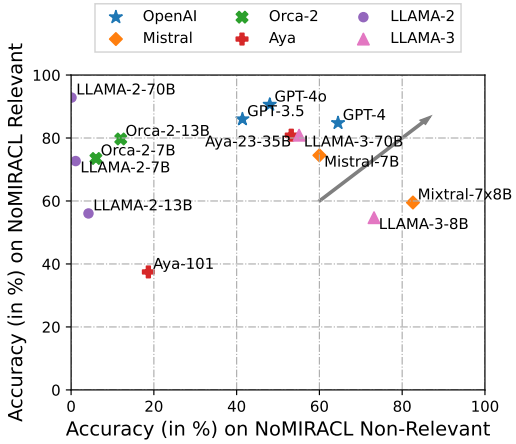


Figure 10.6: Average model accuracy on relevant (y-axis) and non-relevant subset (x-axis) in NoMIRACL. Models closer to the top-right corner are better.

10.5 Empirical Analysis

In this section, we conduct an empirical analysis of LLM outputs, with both the non-relevant subset (containing hallucinations) and the relevant subset (containing errors). We conduct our ablation study on the English (`en`) subset in NoMIRACL. We categorize each LLM output pattern as either positive or accurate (highlighted in **green**), unable to understand instruction (highlighted in **orange**), or either a hallucination or error (highlighted in **red**).

Non-relevant Subset. As shown in Table 10.1 (top segment), we observe a uniform distribution of the hallucination pattern of failed samples for a majority of LLMs by answering "Yes, answer is present" with or without additional explanation. LLMs such as LLAMA-2-7B and LLAMA-2-13B suffer from hallucinations by often repeating the question or instruction in their generation output. Mistral-7B interestingly always provides a rationale or explanation in their model response, whereas Aya-101, uses implicit memory heavily to directly provide an answer instead of grounding the answer from within the retrieved passages. Lastly, models such as Orca-2-7B tend to change the output generation style and often use synonyms such as "*No, answer is not present*" instead of "*I don't know*".

Relevant Subset. As shown in Table 10.1 (bottom segment), similar to the non-relevant subset, we observe a uniform distribution of accurate LLM responses. Interestingly, GPT-4

	OpenAI		Mistral		Orca-2		Aya	LLAMA-2		
	GPT-4	GPT-3.5	8x7B	7B	13B	7B	Aya-101	70B	13B	7B
***Empirical results on the non-relevant subset**: Queries with all human-judged non-relevant passages*										
(i) Perfectly answers "I don't know"	56.8%	54.8%	67.2%	1.6%	3.2%	9.2%	15.6%	0.0%	0.0%	0.8%
(ii) "I don't know" with explanation	0.0%	0.0%	3.6%	88.4%	0.0%	2.0%	0.0%	0.0%	0.0%	0.0%
(iii) Uses a synonym of "I don't know"	0.4%	0.0%	1.6%	1.2%	6.0%	7.6%	0.0%	0.4%	0.0%	2.4%
(iv) Refuses to answer	0.4%	0.8%	0.0%	0.8%	3.2%	0.0%	6.4%	0.0%	0.0%	0.0%
(v) Repeats question or instruction	0.0%	0.0%	0.0%	0.0%	0.0%	3.2%	0.0%	2.0%	64.8%	30.8%
(vi) Conversation	0.0%	0.0%	0.4%	0.0%	0.0%	0.4%	0.0%	0.0%	0.0%	0.4%
(vii) Answers "Yes" w. or w.o. explanation	42.4%	44.4%	27.2%	8.0%	87.6%	68.4%	67.2%	97.6%	34.8%	65.6%
(viii) Uses implicit memory to answer	0.0%	0.0%	0.0%	0.0%	0.0%	8.8%	10.8%	0.0%	0.0%	0.0%
***Empirical results on the relevant subset**: Queries with at least one human-judged relevant passage*										
(i) Perfectly answers "Yes, answer is present"	94.8%	45.2%	22.8%	5.2%	97.2%	71.2%	51.6%	1.2%	3.6%	7.2%
(ii) "Yes, answer is present" with explanation	0.0%	46.0%	56.0%	31.6%	0.8%	11.2%	0.0%	98.8%	46.8%	63.2%
(iii) Refuses to answer	0.4%	0.8%	0.0%	0.0%	0.0%	0.0%	1.2%	0.0%	0.0%	0.0%
(iv) Repeats question or instruction	0.0%	0.0%	0.4%	0.0%	0.0%	0.4%	0.0%	0.0%	48.8%	24.8%
(v) Conversation	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.6%
(vi) Answers "No" w. or w.o. explanation	4.8%	8.0%	20.8%	60.8%	0.8%	2.8%	4.0%	0.0%	0.0%	1.2%
(vii) Uses implicit memory to answer	0.0%	0.0%	0.0%	2.4%	1.2%	14.4%	42.0%	0.0%	0.0%	0.4%

Table 10.1: Empirical results on the complete NoMIRACL English (`en`) non-relevant (above) and relevant (below) subsets. The analysis is bracketed into three categories, where **green** category denotes an accurate response, **orange** denotes limitations in understanding the instruction and **red** denotes model hallucination or error respectively.

and Orca-2-13B overall only provide a single output classification token, whereas models such as GPT-3.5 or Mixtral-8x7B provide an additional rationale or explanation. Similar to the non-relevant subset, LLAMA-2-7B and 13B models repeat the instruction in their output and Aya-101 sometimes uses implicit memory.

10.6 Further Studies

Prompt Optimization. Prompting is crucial in handling the robustness evaluation of multilingual-focused LLMs. Techniques such as Chain-of-Thought (CoT) [367] or algorithmically optimizing prompts using DSPy [158] highlight the necessity of prompt optimization. Although optimizing for the prompt is certainly challenging and expensive to evaluate all LLMs across 18 languages relevant and non-relevant subsets, we experiment with three listwise variations techniques inspired by Thomas et al. [331]. The prompt ablations are shown in Figure 10.7: (i) **role**, we highlight the role of LLM as an evaluator within the prompt at the beginning, (ii) **repeat**, we repeat the task instructions at the end of the prompt to remind the LLM, and (iii) **explanation**, we ask the LLM model to provide a

```
Read the query and the contexts carefully and provide a step-by-step explanation for your answer.
If any of the contexts answers the question, respond as either "Yes, answer is present" or "I don't
know". You must strictly follow the output format with ## Reasoning: ... ## Answer: "Yes, answer is
present" OR "I don't know".
-----------------------------------------------------------------------------------------------------
You are an evaluator checking whether the question contains the answer within the provided contexts or
not. I will give you a question and several contexts containing information about the question. Read
the contexts carefully. If any of the contexts answers the question, respond as either "Yes, answer is
present" or "I don't know":
-----------------------------------------------------------------------------------------------------
QUESTION: {query}

CONTEXTS:
[1] {Passage title}: {Passage text}
[2] {Passage title}: {Passage text}
...
[10] {Passage title}: {Passage text}

Please remember to read all the contexts carefully. If any of the contexts answers the question:
{query}, respond as either "Yes, answer is present" or "I don't know".

OUTPUT:
```

Figure 10.7: The prompt and its ablations used for LLM robustness evaluation for 18 languages in NoMIRACL. The instruction is provided in English, similar to Ahuja et al. [7]. Vanilla zero-shot prompt template is written in black. Three ablations and their modifications are highlighted by color: (i) **role**, (ii) **repeat**, and (iii) **explanation** that replaces the preamble of the vanilla template.

step-by-step explanation and then answer and require 400 output tokens to fit both the LLM reasoning and the answer.

Prompt Optimization Results. We evaluate Mistral-7B with three prompt variations independently on NoMIRACL. The complete results are listed in Table 10.2. On average, both role and repeat techniques help reduce the error rate in the NoMIRACL relevant subset by 6.3% and 15.2% but overall increase the hallucination rate by 8.7% and 15.9% respectively. On the other hand, prompting with explanation decreases the hallucination rate by 9.7% but increases the error rate by 8.3%.

Answer to RQ3: *Prompt optimization techniques leads to mixed results with Mistral-7B: role and repeat templates, decreases the error rate but increases the hallucination rate. Whereas, generating explanations (or reasoning) before the output, decreases the hallucination rate but increases the error rate.*

	ar	bn	de	en	es	fa	fr	fi	hi	id	ja	ko	ru	sw	te	th	yo	zh	Avg.
Hallucination Rates (in %) on NoMIRACL test split (non-relevant subset)																			
Original	40.0	63.2	38.2	42.8	17.2	52.4	47.6	16.1	39.6	30.8	44.4	28.8	41.6	14.8	74.0	58.8	23.6	45.2	40.0
(+ Role)	35.6	60.0	53.5	62.4	40.4	38.0	71.2	29.8	38.4	49.2	59.2	41.2	55.6	21.2	72.0	54.4	32.4	62.8	48.7
(+ Repeat)	47.2	72.4	56.7	50.8	35.2	69.2	70.0	50.0	49.6	48.8	65.2	48.4	56.0	35.2	78.4	72.0	43.6	58.4	55.9
(+ Explanation)	28.0	33.6	30.4	34.8	16.8	39.6	42.4	31.5	18.0	27.2	32.8	26.8	31.6	22.8	36.4	34.0	27.2	31.6	**30.3**
Error Rates (in %) on NoMIRACL test split (relevant subset)																			
Original	14.4	20.4	21.6	8.0	28.0	22.4	15.6	38.0	30.8	43.6	16.0	30.0	18.0	57.6	17.6	13.6	50.0	13.2	25.5
(+ Role)	15.2	23.6	10.0	3.2	14.4	23.2	7.2	29.2	25.2	32.8	9.6	21.6	10.8	47.2	17.6	17.2	33.8	4.4	19.2
(+ Repeat)	8.4	12.4	9.2	5.2	12.4	4.8	5.6	6.0	16.4	16.8	1.6	12.0	6.8	22.8	14.0	6.0	21.6	3.6	**10.3**
(+ Explanation)	28.8	39.6	26.0	16.8	35.6	35.6	20.0	32.0	46.8	46.8	24.4	33.6	24.8	50.4	48.8	31.2	48.0	18.8	33.8

Table 10.2: Hallucination and error rates on the NoMIRACL test split (non-relevant and relevant subsets) with three types of prompting techniques on Mistral-7B (v0.3).

10.7 Summary

We introduced NoMIRACL, a multilingual human-labeled dataset for relevance assessment of LLM robustness as a binary relevance identification task in 18 languages. Our multilingual dataset was human-annotated and constructed with 31 native speakers. We provided two subsets in NoMIRACL, the non-relevant subset, where queries contained all judged non-relevant passages, and the relevant subset, where queries contained at least one relevant judged passage to measure hallucination on the non-relevant and error on the relevant subset, by converting the relevance assessment into a binary classification task.

Our experimental results indicated that existing multilingual-focused LLMs were not robust, as we observed challenges in LLM robustness in either the hallucination or error metric. GPT-4 achieved the best model and performance tradeoff across both subsets. NoMIRACL can facilitate research in understanding to which extent LLMs tend to hallucinate, ultimately paving the way for building more effective and robust multilingual-focused LLMs in the future.

In the next chapter, we will learn about a technique to combine heuristic features and train a surrogate judge to automatically rank LLMs for multilingual RAG evaluation.

Chapter 11

Automatic Benchmarking of Multilingual Retrieval-Augmented Generation

RAG benchmarks can be broadly classified as either (i) *heuristic-based*, where benchmarks handcraft evaluation metrics (e.g., faithfulness or fluency) to evaluate systems on multiple dimensions [113, 59, 384] or (ii) *arena-based*: where systems compete each other in a tournament and an LLM-based teacher is used as the judge for evaluation [265, 258]. Heuristic-based benchmarks are computationally cheaper to evaluate but require human preferences as gold truth for reference. They also face challenges in aggregating different features into a ranking order for models. On the other hand, arena-based benchmarks require a high-performance or reasoning-intensive LLM as a teacher for accuracy [403], which makes exhaustive pairwise comparisons expensive for a large set of models.

In this chapter, we get the best of both worlds with a surrogate judge, a learning to rank model, e.g. random forest [126], using heuristic features to estimate an arena-based leaderboard obtained with a Bradley-Terry model [135] from pairwise evaluations using an LLM as a judge [403]. We use bootstrapping to obtain confidence bounds for better statistical estimates. After training, the learning to rank model can be used to estimate the performance of newer released models reliably in the future without the expensive LLM as a judge. It also provides better interpretability and is easily retrainable with a different or newer set of heuristic features.

We develop Mirage-Bench, a RAG benchmark across 18 languages for multilingual generation evaluation on Wikipedia. Mirage-Bench data is sourced from MIRACL [398], a multilingual retrieval dataset containing human-generated queries and human-labeled relevance judgments for Wikipedia passages. We benchmark 19 frontier multilingual LLMs

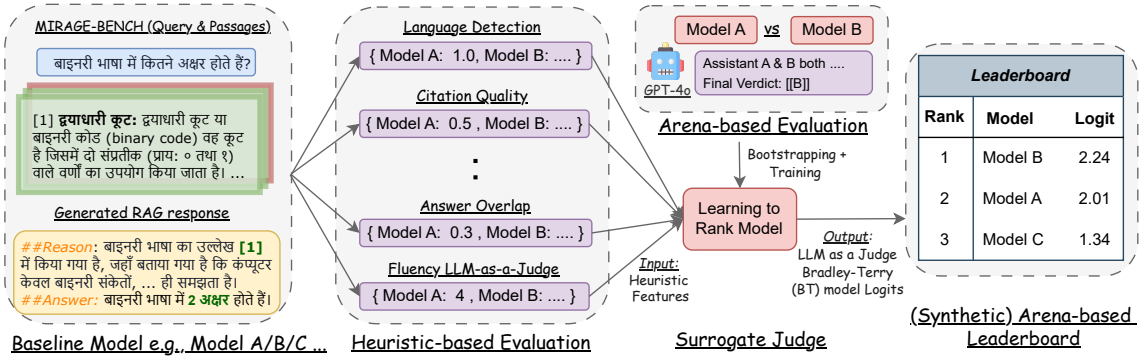


Figure 11.1: The MIRAGE-BENCH evaluation flowchart consists of three steps: (i) heuristic-based features evaluating the baseline model response across several dimensions; (ii) exhaustive pairwise comparisons with GPT-4o as a judge on a small subset of queries to train our surrogate judge. (iii) After training, we utilize our surrogate judge to output the model ranking on the whole subset of queries, to construct the RAG arena-based leaderboard.

in our experiments. Our evaluation flowchart adopted in MIRAGE-BENCH is shown in Figure 11.1. More specifically, we ask the following research questions in our work:

RQ1 *Can we avoid the LLM as a judge cost and combine heuristic-based evaluation?*
RQ2 *How do frontier multilingual-focused LLMs perform in multilingual answer generation?*
RQ3 *Does fine-tuning on MIRAGE-BENCH training dataset improve LLM performance?*

11.1 An Overview of MIRAGE-BENCH

We select 18 languages in MIRAGE-BENCH as the starting point, representing an appropriate cross-section of the diversity of the languages spoken worldwide at this point. MIRAGE-BENCH is a comprehensive multilingual RAG benchmark focusing on the generation task evaluation. As shown in Table 11.1, the benchmark includes 11,195 evaluation pairs and 39,763 training pairs across 18 languages.

11.1.1 Distinction and Extension from MIRACL

MIRACL [398] (as discussed earlier in Chapter 4) is a monolingual retrieval dataset, which evaluates the retrieval task, i.e., given a user query and a passage corpus, retrieve the

	ar	bn	de	en	es	fa	fi	fr	hi	id	ja	ko	ru	sw	te	th	yo	zh
MIRAGE-BENCH Evaluation Dataset																		
# Queries	1501	411	304	787	617	632	1183	343	350	939	797	213	1247	481	150	730	119	391
# Avg. Query Tokens	10.2	17.6	10.4	9.1	11.4	14.1	12.2	10.4	17.3	10.1	14.6	14.2	14.3	12.0	16.1	19.6	13.0	8.0
Avg. Rel. Passages / Q	2.0	2.1	2.6	2.8	4.3	2.1	2.0	2.1	2.1	3.1	2.2	2.6	2.8	1.9	1.2	1.8	1.2	2.5
Avg. Non Rel. Passages / Q	8.1	8.0	7.7	7.7	5.6	8.3	8.1	7.9	7.8	7.0	8.2	11.8	7.6	8.7	4.9	8.5	8.8	7.5
MIRAGE-BENCH Training Dataset																		
# Queries	3468	1624	—	2857	2159	2104	2878	1137	1165	4054	3466	859	4567	1866	3283	2965	—	1311
# Avg. Query Tokens	10.2	17.4	—	8.7	11.2	14.2	11.8	10.4	17.1	9.9	14.5	14.9	14.3	12.0	16.1	19.3	—	7.9
Avg. Rel. Passages / Q	1.8	2.3	—	2.5	3.6	2.0	1.7	2.0	2.0	2.9	1.9	2.1	2.0	1.4	1.2	1.6	—	2.3
Avg. Non Rel. Passages / Q	5.5	7.9	—	7.5	5.3	8.3	5.3	8.0	7.9	7.1	7.9	12.4	5.2	3.5	4.3	5.6	—	7.6
# Avg. GPT-4o Context Tokens	105.5	144.5	—	137.9	203.4	133.8	129.4	180.1	149.4	100.8	140.2	136.4	90.0	106.4	54.6	86.0	—	137.9
# Avg. GPT-4o Answer Tokens	35.7	22.9	—	20.6	55.3	51.6	30.3	56.0	26.6	22.6	24.1	23.9	16.1	18.4	11.4	20.4	—	40.6

Table 11.1: Dataset statistics for 18 languages in MIRAGE-BENCH; All tokens are calculated using the GPT-4o tokenizer [246]; (Rel.) denotes relevancy; (# Avg GPT-4o) counts the tokens in the GPT-4o generated context and answer; Training data is not available for two languages (denoted by —): German (de) and Yoruba (yo).

ranked list of passages as output. In contrast, MIRAGE-BENCH evaluates the generation task in RAG, requiring LLMs to generate a summarized answer given the query and context available from oracle-judged passages.

In our work, we reuse the queries and relevance-judgments from MIRACL, but solely evaluate the multilingual generation task in RAG measuring answers using both heuristics and LLM as a judge. Since MIRAGE-BENCH evaluates the generation task containing the oracle context, independent of the retrieval task, the chances of contamination in MIRAGE-BENCH evaluation from fine-tuning MIRACL is less overall.

11.1.2 MIRAGE-BENCH Evaluation Dataset

MIRACL queries are high-quality and generated by native speakers [398]. The annotation procedure in MIRACL is identical to TyDI-QA [71]. The passage collection is constructed from language-specific Wikipedia corpora and parsed using WikiExtractor. The MIRAGE-BENCH evaluation dataset is constructed re-using the queries and oracle-judged passages available in the MIRACL development split.[1]

We incorporate two changes: (i) In Arabic (ar), we randomly sample a smaller subset of 1,501 out of 2,896 queries for uniformity in the number of queries available across other languages. (ii) We filter out a small subset of queries with zero non-relevant passages, i.e.,

[1]We did not utilize the test split in MIRACL as the relevance judgments are not publicly available.

we always include queries with hard passages from MIRACL to make the MIRAGE-BENCH evaluation challenging.[2]

11.1.3 MIRAGE-BENCH Training Dataset

The MIRAGE-BENCH training dataset is developed from the MIRACL training dataset [398]. In MIRAGE-BENCH, we reuse all the MIRACL training pairs available in 16 languages (except German (de) and Yoruba (yo)) and convert it using a simple recipe into a multilingual instruction-tuned RAG dataset [396, 238]. In our recipe, we first keep only the relevant passages as context along with the input query to generate a zero-shot RAG output by prompting strong teachers such as GPT-4o [246], LLAMA-3 (70B) [92] and Mixtral (8×22B) [146]. After generation, we include non-relevant passages within our prompt as "distracting and noisy context" to help improve the quality of the training dataset. Since we convert a retrieval dataset into a RAG benchmark, we do *not* have human-annotated answers in MIRAGE-BENCH.

11.2 Multilingual RAG Evaluation

11.2.1 Heuristic-based Evaluation

Answer generation in RAG requires evaluation on various dimensions. For example, whether a system's response provides the correct final answer or cites the relevant documents, a single metric alone is *not sufficient* to capture the comprehensive evaluation required for RAG systems. Inspired by other recent works [160, 297, 113], we introduce five *deterministic* features and two *LLM-measured* features for evaluation in our work. We rely on features that are explainable, cheap, and fast to compute.

Language Detection. In a multilingual RAG system, the output response should ideally be in the same language in which the user asked their query. To capture this feature, we attempt to identify which natural language the output response is in. We use `langid` [207], an off-the-shelf language detecting Python library for detecting the language of the long-form RAG answer. We use the probability of the target language detected as the score for language identification, i.e., $\hat{p} = \text{langid}(a, t)$, where t denotes the target language and a denotes the long-form answer.

[2] An exception is Telugu (te), where 78 queries have at least one non-relevant passage. Therefore, we randomly sample 72 additional queries with only relevant judged passages.

Citation Quality. A multilingual RAG system must cite information from the relevant passages within their answers, to improve faithfulness and reduce hallucinations. We capture whether the passages (using relevance judgments provided in MIRACL [398]) are cited in the LLM response. For scoring, we compute Recall@k that is 1.0 for a generated response a, if and only if a cites all available relevant passages, and MAP@k score measures the percentage of relevant passages within the top-k cited passages in response a.

Support. RAG systems have been shown to hallucinate across retrieval-augmented generation tasks, especially when provided with non-relevant contexts [324]. Grounding is necessary to avoid hallucinations. We compute support evaluation [113] of every sentence s_j in generated answer A along with the cited context c_j using the multilingual NLI model [124],[3] which computes the similarity score as a probability of either *entailment*, *neutral* or *contradiction*. We segment the answer into sentences using Stanza [261] or SentenceX tokenizer[4] (if language not available in Stanza). We compute the probability of the *entailment* and *neutral* score, macro-averaged across the sentence-citation pairs.

Reranker Score. The reranker score measures the semantic similarity between the user query and the cited passages within the system's response. If the cited passages are relevant in answering the query, the reranker model would output a higher similarity score. We compute the reranker score using a multilingual reranker model [59],[5] macro-averaged across the query-passage pairs.

Answer Overlap. Having the correct answer is crucial in the RAG system's response. Since MIRAGE-BENCH does not include a human-labeled answer, we use the generated answer from GPT-4 [246] as the gold truth. We compute two traditional metrics: SacreBLEU [252] and ROUGE-L [188] measuring the lexical word overlap between the gold answer (GPT-4's answer is used for reference) and the system's response.

Answer Overlap (LLM-measured). In addition, we evaluate using LLAMA-3 (8B) [92], an open-source LLM as a judge evaluator in a pointwise setup, providing a semantic word overlap integer score in the range $[1, 5]$.

Fluency (LLM-measured). Fluency measures for grammatical correctness and idiomatic word choices in the system's response. In multilingual long-form generation, evaluation is not straightforward. While existing techniques are available for English such as MAUVE [255], only a few models evaluate multilingual texts [70]. Inspired by G-EVAL [201], we use LLAMA-3 (8B) [92] in a pointwise setup, providing an integer score in $[1, 5]$.

[3] https://huggingface.co/MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7
[4] https://github.com/wikimedia/sentencex
[5] https://huggingface.co/BAAI/bge-reranker-v2-m3

11.2.2 Arena-based Evaluation

Heuristic-based evaluation metrics often rely on a gold standard for evaluation. Tasks such as text retrieval [29, 329] require human-labeled relevance judgments, and similarly, NLP tasks such as machine translation [313], require human-annotated translations. As human preferences are seldom available in numerous applications, using LLM as a judge [403, 56, 64] is becoming a de facto approach for arena-based evaluation of LLMs.

We evaluate pointwise, listwise, and pairwise LLM as judge evaluations. We anecdotally observe the pointwise judge, which is efficient $O(n)$, but is not good for ranking LLMs as it provides similar scores for a wide range of models (e.g., 4/5 score for 16 out of 18 LLMs evaluated). Similarly, the listwise judge finds it difficult to rank all 19 models in the correct order. Therefore, although suboptimal in complexity, $O(n^2)$, where n is the number of models evaluated, we choose a pairwise evaluation in our work.

Pairwise LLM as a Judge. Following prior works on arena-based evaluation in RAG [265, 258], we evaluate two system's responses in a head-on comparison by computing pairwise judgments with LLM as a judge. We reuse the RAGElo prompt template [265] with minor additions. The prompt is not provided in this thesis, but can be found in the original publication. LLM as a judge evaluator includes three types of biases [403]: (i) verbosity bias [372] (ii) self-enhancement bias [378, 251] and (iii) position bias [364]. We avoid the verbosity bias, as RAG evaluation has fixed evaluation criteria requiring sentence-level citations and answers [258] and the position bias by randomly swapping the position of two models.

11.2.3 Learning to Approximate Rankings

There is no predefined way to aggregate the heuristic features to provide a leaderboard ranking in MIRAGE-BENCH. Averaging the scores is too simplistic as features measure different aspects of RAG evaluation. On the other hand, arena-based evaluations provide ranked leaderboards, but are computationally expensive with a strong teacher model. To avoid computational costs, smaller models as teachers have been proposed [325, 237].

Motivated by similar observations, we train a surrogate judge to effectively emulate an arena-based leaderboard without incurring the expensive LLM as a judge pairwise cost. We find the random forest model to serve as a scalable and cost-effective judge that can be trained within minutes on small training datasets without expensive computation. Therefore, we train a random forest (learning to rank model) as a surrogate judge to approximate the Bradley-Terry model coefficients [135] learned from an arena-based evaluation that uses GPT-4o as a judge for pairwise judgments.

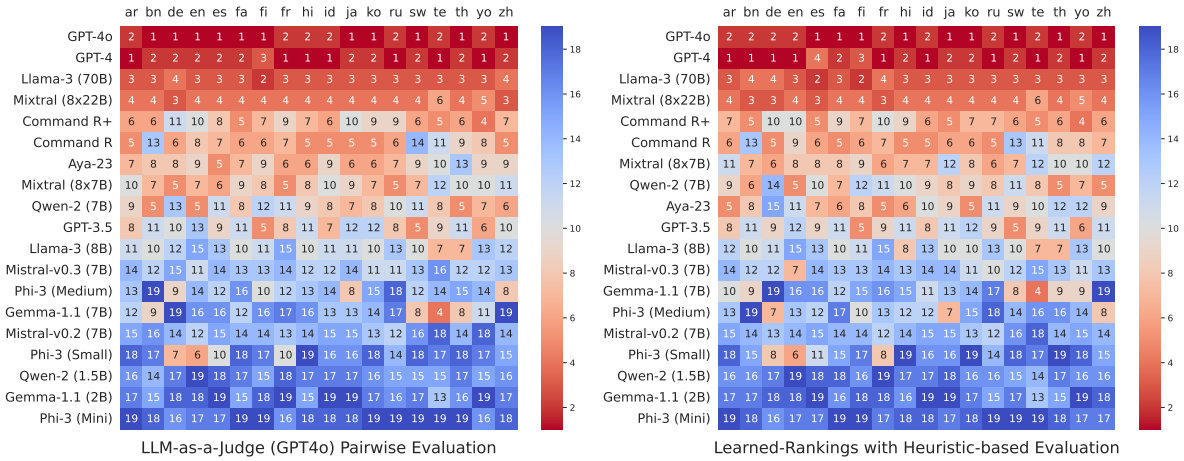


Figure 11.2: MIRAGE-BENCH arena-based leaderboards: (left heatmap) Bradley-Terry model coefficients with GPT-4o as a pairwise judge for a subset of 100 sampled queries; (right heatmap) Synthetic rankings using heuristic-based features and a random forest model as a surrogate judge on all queries. Each highlighted cell denotes the rank of the LLM (lower the better). LLMs are sorted by lowest to highest average rank.

Learning to Rank Model. While the heuristic features introduced in Section 11.2.1 can be computed efficiently and without the reliance on proprietary LLMs, inducing a ranking from pairwise comparisons via a Bradley-Terry model is computationally expensive and requires access to a high-performance LLM. Furthermore, as we demonstrate in Section 11.4.3, the ranking accuracy, measured by the Kendall-Tau (τ) coefficient, degrades rapidly when subsampling tournament matches.

The procedure, detailed in Algorithm 1, simulates N_t tournaments, each involving a total of N_l models and N_q queries. For each query, judgments are obtained for all $\binom{N_l}{2}$ pairings of models. We employ bootstrapping on the query selection process to estimate the variance in the R^2 metric in the learning to rank models' approximations of the Bradley-Terry coefficients over a randomly-sampled holdout set, $\text{LLM}_{predict}$.

We randomly select two models, Gemma 1.1 (2B) and LLAMA-3 (70B) as holdout models, i.e., we do not train on the features for holdout models. For English, we observe an average $\bar{R}^2 = 0.971$ with a 95% confidence interval of $[0.905, 0.999]$. On the other hand, for Bengali, we observe $\bar{R}^2 = 0.937$ with a 95% confidence interval of $[0.766, 0.998]$. $\bar{R}^2$ scores for all 18 languages with 95% confidence intervals. Taken together, these results indicate that the training procedure is fairly robust with $N_q = 100$.

Answer to RQ1: *Training a surrogate model as a judge to approximate the pairwise*

Algorithm 1 Simulate Tournaments and Fit Models

for $i \in [N_t]$ **do**
  $M_{BT}^i \leftarrow \textsc{Tournament}(N_q)$
  $X_t, Y_t \leftarrow \textsc{Dataset}(\text{LLM}_{\text{train}}, M_{BT}^i)$
  $X_p, Y_p \leftarrow \textsc{Dataset}(\text{LLM}_{\text{predict}}, M_{BT}^i)$
  $M_{reg}^i \leftarrow \textsc{Fit}(X_t, Y_t)$
  $R_2^i \leftarrow M_{reg}^i(X_p, Y_p)$
end for
$M_{BT} \leftarrow [M_{BT}^1; M_{BT}^2; \ldots; M_{BT}^{N_t}]$
$M_{reg} \leftarrow [M_{reg}^1; M_{reg}^2; \ldots; M_{reg}^{N_t}]$
$R_2 \leftarrow [R_2^1; R_2^2; \ldots; R_2^{N_t}]$

Note: The Tournament function runs a battle arena, sampling q queries, and returning the learned Bradley-Terry model. The Dataset function accepts a set of LLMs and a learned Bradley-Terry model. It returns X, the heuristic RAG feature values, and Y, the Bradley-Terry coefficients for each LLM model.

comparisons using an LLM judge with a Bradley-Terry model, by learning the input heuristic feature scores helps in generating a synthetic RAG arena effectively.

11.3 Experimental Settings

11.3.1 Multilingual Baselines

Existing frontier LLMs are either English-only or support a limited set of languages, predominantly due to the *curse of multilinguality* for large models [74]. It is unclear how well existing LLMs perform on RAG on a wide variety of languages, due to the scarce availability of multilingual instruction tuning datasets. We experiment with LLMs from seven different families, containing proprietary and open-source LLMs. Wherever possible, we evaluate the *instruction-tuned* version if available.

- **OpenAI:** GPT-3.5-turbo, GPT-4, and GPT-4o [246] using the Azure OpenAI service.
- **Mistral:** Mistral-Instruct-v0.2 and v0.3 (7B) [145] and Mixtral-Instruct-v0.1 (8×7B) and (8×22B) versions [146].
- **Cohere:** Command-R (35B), Command-R+ (104B) and Aya-23 (35B) [22].
- **Gemma:** Gemma 1.1 instruct (2B) and (7B) models [223].

Lang.	τ	Lang.	τ	Lang.	τ	Lang.	τ	Lang.	τ	Lang.	τ	Lang.	τ	Lang.	τ	Lang.	τ
ar	0.951	bn	0.874	de	0.825	en	0.835	es	0.876	fa	0.924	fi	0.949	fr	0.914	hi	0.946
id	0.896	ja	0.892	ko	0.950	ru	0.849	sw	0.958	te	0.938	th	0.946	yo	0.906	zh	0.941
Avg. Kendall Tau (τ) across 18 languages = 0.909																	

Table 11.2: Kendall τ rank correlation between Bradley-Terry model (using GPT-4o as judge) and the synthetic arena-based leaderboard (using our surrogate judge).

- **LLAMA-3:** LLAMA-3 instruct (8B) and (70B) models [92].
- **Phi-3:** Phi 3 instruct series: Medium (14B), Small (7B), and Mini (3.8B) [2].
- **Qwen-2:** Qwen-2-instruct series: 1.5B and 7B [380].

Prompt template. We internally optimize[6] the ChatQA prompt template [200], to include in-text citations of the context passages following the IEEE format [152]. In MIRAGE-BENCH, we have about 10 passages annotated in the oracle setting. Therefore, we trim each passage available and take the first 800 tokens to fit all passages within a fixed context length of 8192 tokens. following prior work in Shi et al. [302], the prompt requires the LLM to explain the multilingual generation task starting with "`##Reason`" and the answer itself starting with "`##Answer`". Utilizing this output format has its advantages in easily parsing the generated answer and the rationale behind the answer.

11.4 Experimental Results

11.4.1 Heuristic-based Results

Figure 11.3 shows lollipop plots indicating the average heuristic-feature value (y-axis) distribution for each language (x-axis). In English detection (higher the worse), smaller LLMs such as Gemma-1.1 (2B) do not generate output in the required target language but rather generate reasoning and answers in English. For citation quality, support, and reranker score features, LLMs from OpenAI and LLAMA-3 family achieve high Recall@10 and entailment scores (except LLAMA-3 (70B) for a few languages), indicating generated answers include grounded citations from relevant passages. In contrast, LLMs from the Qwen-2 or Gemma-1.1 family, tend to under-cite passages in their answers.

[6] A majority of the prompt optimization was based on eye-balling responses across LLMs.

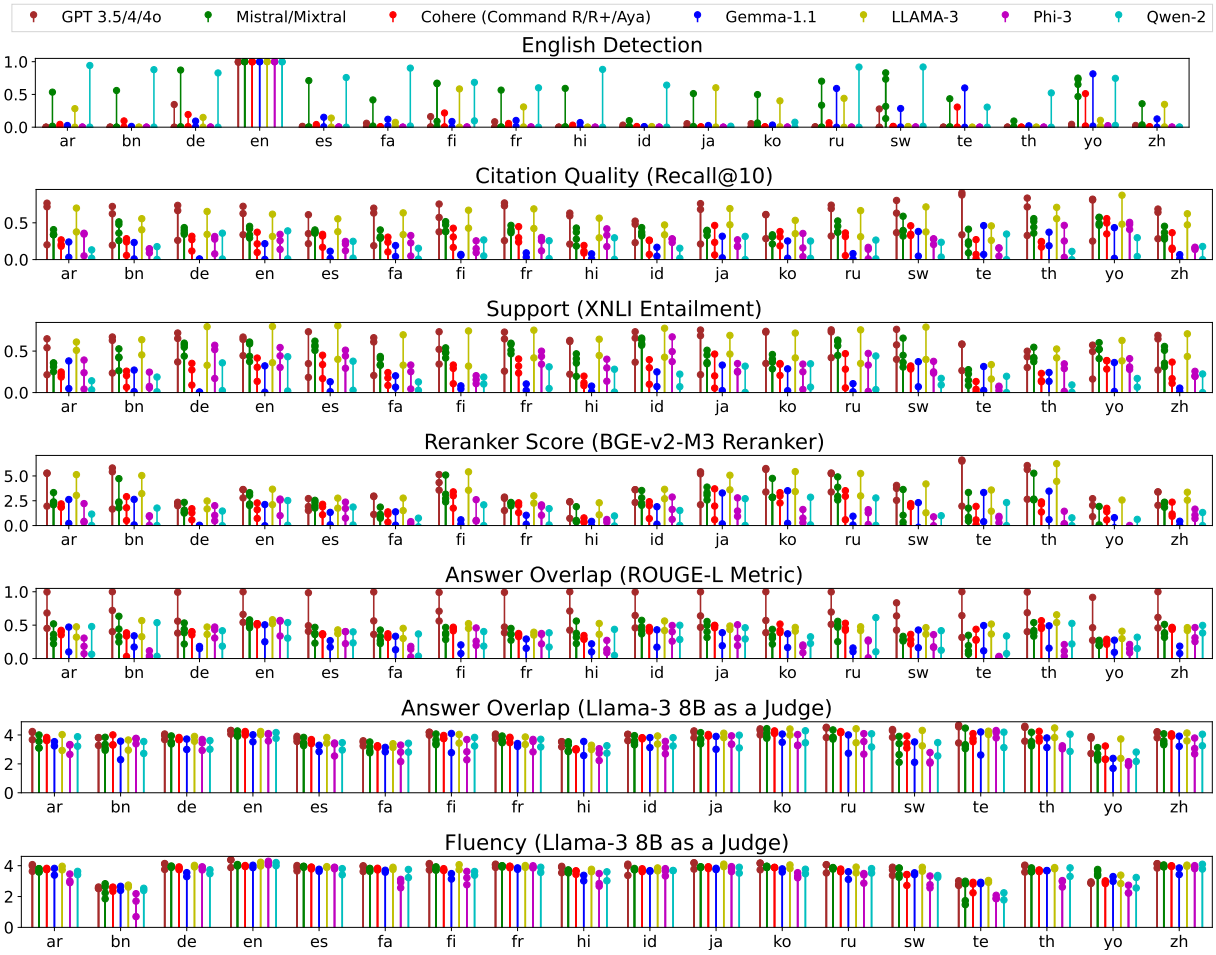


Figure 11.3: Lollipop plots denoting the average heuristic-based feature scores achieved by LLM baselines for each language in MIRAGE-BENCH. x-axis denotes the 18 languages; whereas y-axis plots every heuristic feature score. Models in the same LLM family are represented in the same color in a lollipop (as multiple circles).

Furthermore, we observe LLMs from the OpenAI family achieve the highest word overlap in the ROUGE-L metric (GPT-4 used as ground truth, which could be a potential bias) and LLAMA-3 (8B) as a judge, while we observe less variance across other LLMs. In Fluency, we observe a majority of the LLMs are rather fluent in generation, except Bengali (bn), Telugu (te), and Yoruba (yo).

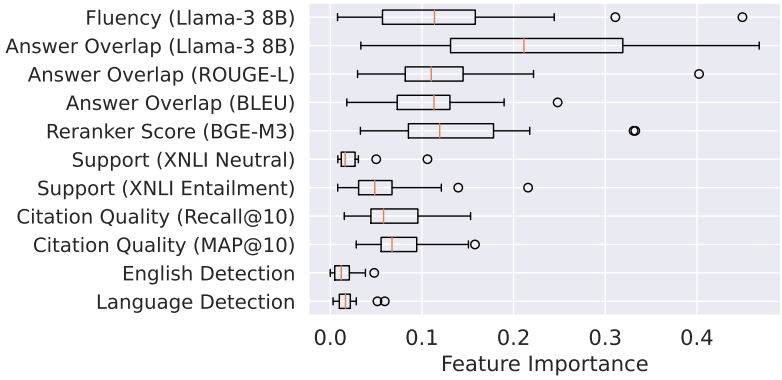


Figure 11.4: Boxplot with the feature importance value (averaged across 18 languages in MIRAGE-BENCH) observed by the learning to rank (random forest) model.

11.4.2 Arena-based Results

Figure 11.2 (left heatmap) is the arena-based leaderboard with the Bradley Terry model conducting 200 tournaments and bootstrapping 100 matches per tournament on a subset of 100 queries using GPT-4o as a pairwise judge. We observe that proprietary LLMs such as GPT-4o and GPT-4, and larger open-source LLMs such as LLAMA-3 (70B) and Mixtral (8×22B) perform better than other LLMs. LLM rankings across languages are usually stable; with a few notable exceptions such as Gemma-1.1 (7B) which achieves a rank of 4 in Telugu. Command R (35B) performs poorly in low-resource languages such as Bengali (rank 13) or Swahili (rank 14).

Synthetic Rankings using Random Forest. Figure 11.2 (right heatmap) is the learned synthetic leaderboard rankings on all queries using heuristic-based features trained with a random forest model. The synthetic leaderboard generated using a surrogate judge, highly correlates to the GPT-4o as a pairwise judge, achieving an average Kendall-Tau (τ) rank correlation = **0.909**, by training on 17 LLMs during training and keeping 2 LLMs as a holdout for every language. Individual language-specific Kendall-Tau rank correlation scores are listed in Table 11.2. This provides evidence of the efficacy of training a random forest model as a surrogate judge.

Heuristic Feature Importance. In Figure 11.4, we plot the average feature importance achieved by our random forest model as a surrogate judge. Using LLAMA-3 (8B) as a judge, for fluency and answer overlap are the important heuristic features. Similarly, deterministic answer overlap and reranking-based metrics are equally important. Some heuristic features such as language detection, and neutral score in support evaluation

Model (↓) /	Train R^2 (15 models)							Holdout R^2 (4 held-out models)						
Language (→)	ar	bn	fi	ja	ko	ru	te	ar	bn	fi	ja	ko	ru	te
Random Forest	0.97	0.96	0.97	0.96	0.97	0.95	0.97	**0.50**	0.41	0.49	**-0.03**	**0.45**	0.07	**0.83**
Linear Regression	0.98	0.98	0.98	**1.00**	0.97	0.99	**1.00**	-1.92	-5.45	-19.38	-0.06	-26.53	-2.13	0.31
MLP Regressor	0.97	0.98	0.97	0.96	0.96	0.98	0.99	0.33	0.37	0.45	-0.76	-0.04	-0.48	0.78
XGB Regressor	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-0.02	0.44	0.22	-1.33	-0.09	-0.80	0.59
SVR	0.75	0.77	0.81	0.59	0.67	0.73	0.39	-0.03	**0.48**	**0.64**	-0.53	-0.09	**0.15**	-0.10

Table 11.3: Train and Holdout R^2 scores using different learning to rank model choices. Each experiment has been repeated 50 times with four held-out models.

obtain the least importance. We observe all answer-related heuristic features achieve a high importance indicating that the generated "answer" portion in an LLM's response is crucial and required to learn rankings from GPT-4o as a pairwise judge.

Answer to RQ2: *Proprietary models such as GPT-4o and GPT-4 and larger open-source models such as LLAMA-3 (70B) and Mixtral (8×22B) outperforms other frontier multilingual LLMs in multilingual answer generation in* MIRAGE-BENCH.

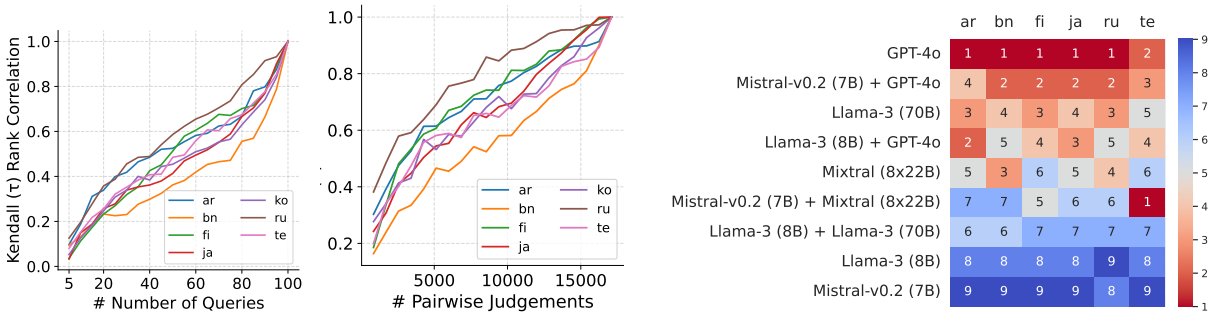
11.4.3 Ablations & Discussion

To understand the gaps observed during training of the random forest model as a surrogate judge, we conduct four ablations on a subset of seven languages Arabic (ar), Bengali (bn), Finnish (fi), Japanese (ja), Korean (ko), Russian (ru) and Telugu (te):

Learning to rank model choice. We compare different learning to rank models as choices for learning the Bradley-Terry model coefficients. We conduct our experiments on the train set, where models contain pairwise judgments, and on a randomly sampled holdout set, a realistic scenario, with no available training data. We evaluate the following choices: Random Forest, Linear Regression, MLP Regressor, XGB Regressor, and SVR. All models are implemented via `scikit-learn`.[7] The results are shown in Table 11.3. Random forest achieves the best R^2 metric on the holdout subset for 4 out of 7 languages. SVR also achieves a similar R^2 metric on the holdout subset, however, underperforms random forest on the training subset. Other baselines, such as XGB Regressor and MLP Regressor show signs of significant overfitting on the training subset, thereby underperforming random forest on the holdout subset.

Non-exhaustive pairwise comparisons lead to performance degradation. Exhaustive pairwise comparisons across a subset of 19 models in MIRAGE-BENCH using GPT-4o for

[7] https://scikit-learn.org/stable/supervised_learning.html



(a) Sampling experiments: reducing queries (left) and reducing pairwise judgments (right).

(b) Approximate rankings using heuristic features for LLAMA-3 (8B) and Mistral-v0.2 (7B) after fine-tuning on MIRAGE-BENCH.

Figure 11.5: Ablation experiments: (a) Sampling experiments and (b) approximate ranking performance after fine-tuning.

all queries are quite expensive. To avoid this, we investigate whether all pairwise exhaustive comparisons are necessary during training. We utilize two sampling techniques: (i) full pairwise judgments on a subsample of 100 queries, e.g., 20 or 50 queries; (ii) partially judge a non-exhaustive random sample of the pairwise judgments across 100 queries, e.g., only 50% of all the exhaustive pairwise combinations. Both results are shown in Figure 11.5a. we observe that Kendall-Tau (τ) correlations increase linearly with queries and pairwise judgments. In summary, an exhaustive pairwise comparison and a sufficient number of queries, such as 100, are necessary without impacting the leaderboard rankings.

All heuristic features are not necessary. We experiment with features used for random forest model training as a surrogate judge. We evaluate four training configurations: (i) all features (ii) without LLM-measured features (iii) without language detection and support, i.e., the low-correlation features observed in Figure 11.4, and (iv) including only LLM-measured features. From Table 11.4, we observe that removing low-correlated heuristic RAG features helps train the random forest model better leading to a conclusion that not necessarily all heuristic features are important. Removing the LLM-measured features completely or only using them for training the model decreases the Kendall-Tau (τ) correlation score.

Fine-tuning on MIRAGE-BENCH training data. We evaluate three variants of the MIRAGE-BENCH training dataset using two smaller open-source LLMs: Mistral-v0.2 (7B) and LLAMA-3 (8B). We fine-tune the MIRAGE-BENCH training datasets using (i) both on GPT-4o, (ii) LLAMA-3 (8B) on LLAMA-3 (70B), and (iii) Mistral-v0.2 (7B) on Mixtral

Features / Language	#F	ar	bn	fi	ja	ko	ru	te
(All) Features	11	0.938	**0.867**	0.921	0.881	0.921	0.826	0.918
(W/o) LLM as a Judge	9	0.912	0.866	0.898	0.853	0.891	0.811	0.904
(W/o) Low. Correlation	7	**0.951**	**0.867**	**0.923**	**0.885**	**0.929**	0.829	**0.940**
(Only) LLM as a Judge	2	0.948	0.728	0.907	0.884	0.916	**0.851**	0.872

Table 11.4: Kendall Tau (τ) scores using different features for training the surrogate judge.

(8×22B). From Figure 11.5b, we observe that GPT-4o is a strong teacher, Mistral-v0.2 (7B) fine-tuned on GPT-4o distilled training data achieves rank 2 outperforming LLAMA-3 (70B). This shows that MIRAGE-BENCH training data is useful for improving the RAG answer generation task quality.

Answer to RQ3: *Instruction tuning helps improve smaller open-source LLMs on multilingual generation, e.g. instruction-tuned Mistral-v0.2 (7B) on GPT-4o distilled training data can outperform the naive LLAMA-3 (70B) baseline on MIRAGE-BENCH.*

11.5 Summary

In this chapter, we presented MIRAGE-BENCH, a multilingual RAG benchmark for 18 languages aimed at evaluating the multilingual generation component within RAG, and aggregated traditional heuristic-based features to train a lightweight learning-to-rank model as a surrogate judge that learned a Bradley–Terry model from GPT-4o pairwise judgments.

We demonstrated the effectiveness of our efficient, inexpensive, and easy-to-retrain random forest surrogate judge, which was trained using only computationally cheap heuristic features for arena-based leaderboard ranking, by achieving a strong correlation of 0.909 Kendall τ with GPT-4o as a pairwise judge.

On MIRAGE-BENCH, we observed that most proprietary and open-source larger LLMs dominated, whereas smaller open-source LLMs continued to struggle. Finally, we found that instruction tuning on MIRAGE-BENCH training data improved smaller open-source LLMs on multilingual generation; for example, instruction-tuned Mistral-v0.2 (7B) on GPT-4o-distilled training data outperformed LLAMA-3 (70B) on MIRAGE-BENCH.

Chapter 12

Conclusion and Future Work

In this thesis, we presented techniques for improving robustness in retrieval and retrieval-augmented generation across heterogeneous domains and languages. Throughout the thesis, we explored retrieval benchmarks focused on specialized domains and languages, generated synthetic data and curated supervised retrieval training datasets, and evaluated retrieval-augmented generation, collectively pushing the boundaries of information retrieval and retrieval-augmented generation.

Our work on benchmark construction demonstrates substantial advances in benchmarking and evaluation of first-stage retrievers on real-world challenges. Subsequently, our work on data curation and synthetic data generation improves data quality and thereby improving retrieval robustness. In addition, our work on evaluation of retrieval-augmented generation systems, proposes a benchmark for a reusable and standardized RAG system module, and provides deep insights into relevance assessment and source attribution in generation. Specifically, our contributions are summarized as follows:

- **Retrieval Benchmarks to Enable Challenging Evaluation** (Chapters 3, 4 and 5): We curate the Touché 2020 dataset by removing short non-argumentative documents and adding post-hoc relevance judgments. On the multilingual side, we develop MIRACL, a large human-supervised dataset that accelerates retrieval accuracy across diverse languages. Finally, we introduce FreshStack, a holistic framework for constructing realistic datasets in niche and challenging technical domains, designed to avoid data contamination due to perpetual recency.

- **Curating and Generating Training Datasets for Robust Dense Retrieval** (Chapters 6 and 7): We construct SWIM-IR, a 28M synthetic multilingual training

dataset across 33 languages by prompting a frontier multilingual LLM, and use it to train effective multilingual dense retrievers on par with supervised counterparts. Next, in RLHN, we prune datasets and relabel false negatives at the sample level in supervised training datasets from the BGE collection using cascading LLM judges. The curated training datasets directly improve model generalization and robustness on queries from out-of-domain (OOD) retrieval datasets.

- **Standardizing Retrieval-Augmented Generation Evaluation** (Chapters 8, 9, 10 and 11): We build Ragnarök, an end-to-end RAG benchmarking framework that includes a curated MS MARCO V2.1 collection, diverse query distributions, and modular pipelines, which is critical for providing baselines at the TREC 2024 Retrieval-Augmented Generation (RAG) Track. In addition, we evaluate support automatically using LLMs as judges, and validate it against human annotations in the TREC 2024 RAG Track. Subsequently, we evaluate relevance assessment by frontier multilingual LLMs as a binary classification task in NoMIRACL, and develop MIRAGE-BENCH, a RAG benchmark with synthetic arena leaderboards learned using Bradley-Terry models by training a surrogate judge with diverse heuristic features for RAG evaluation.

We hope that, with these contributions, the community continues to improve retrieval and RAG systems in terms of access to and reliability of information across diverse domains and languages. The advancements in benchmarks address the growing demand for efficient and robust RAG systems across digital platforms. At the end of the thesis, we conclude by outlining several promising directions for future work—such as search agents—focused on evaluation, datasets and benchmarking:

- **Benchmarking Deep Research or Search Agents**: Deep research systems, such as OpenAI Deep Research [248] and Gemini Deep Research [115], integrate LLMs with search tools and demonstrate success in handling complex queries that require iterative search planning and reasoning over search results [147, 373]. For example, BrowseComp-Plus introduced by Chen et al. [62] enables reproducible evaluation of search agents using local collections and offline retrieval systems. In the TREC 2025 RAG track, we include topics focused on "deep research" or long, descripted queries to stress test our LLM-based automated evaluations in realistic settings.

- **Avoiding Common Pitfalls in Benchmark Contamination**: We will continue our efforts (as described in Chapter 5) to build new retrieval benchmarks that avoid Goodhart's law of benchmark saturation. Benchmarks are quickly getting saturated (e.g., BEIR [329]) due to excessive leaderboard overfitting. Looking ahead, we explore

industry-grade or private benchmarks, as well as open benchmarks over private data, to help mitigate staleness. Recently introduced benchmarks (such as FreshStack [326]) also require regularly refreshes to avoid staleness and data contamination.

- **Generating Reasoning-Intensive Training Datasets**: On the data side, we explore techniques to generate datasets with reasoning-intensive queries for training search agents [147, 373]. Such queries are longer and more convulated than traditional web search queries, requiring multiple reasoning hops to retrieve relevant documents, and can be easily verified using automated metrics containing short verifiable answers. Reinforcement Learning (RL) is used in post-training search agents, such as in Search-R1 [147]. A promising direction is to incorporate nugget or rubric-based (factual) and support (grounding) metrics as reward models to generate realiable long-form generations in search agents.

- **Expanding RAG-based Evaluation in TREC RAG 2026**: Historically, we evaluate relevance, nuggets and support as RAG metrics in both the TREC 2024 and 2025 RAG tracks. In the upcoming iterations, we plan to incorporate additional aspects that are central to evaluation, including: (1) safety [16] when grounding information in sensitive documents, (2) counterfactual robustness [60] to identify and disregard incorrect information, and (3) negative rejection, i.e., refraining from responding when available information is insufficient.

References

[1] Zahra Abbasiantaeb, Chuan Meng, Leif Azzopardi, and Mohammad Aliannejadi. Can we use large language models to fill relevance judgment holes?, 2024.

[2] Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219, 2024.

[3] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Y. Zou. Gradio: Hassle-free sharing and testing of ML models in the wild. *CoRR*, abs/1906.02569, 2019.

[4] Arkadeep Acharya, Rudra Murthy, Vishwajeet Kumar, and Jaydeep Sen. Benchmarking and building zero-shot Hindi retrieval model with Hindi-BEIR and NLLB-e5. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference*

of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4328–4348, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[5] Mofetoluwa Adeyemi, Akintunde Oladipo, Xinyu Zhang, David Alfonso-Hermelo, Mehdi Rezagholizadeh, Boxing Chen, Abdul-Hakeem Omotayo, Idris Abdulmumin, Naome A. Etori, Toyib Babatunde Musa, Samuel Fanijo, Oluwabusayo Olufunke Awoyomi, Saheed Abdullahi Salahudeen, Labaran Adamu Mohammed, Daud Olamide Abolade, Falalu Ibrahim Lawan, Maryam Sabo Abubakar, Ruqayya Nasir Iro, Amina Imam Abubakar, Shafie Abdi Mohamed, Hanad Mohamud Mohamed, Tunde Oluwaseyi Ajayi, and Jimmy Lin. CIRAL: A Test Collection for CLIR Evaluations in African Languages. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 293–302, New York, NY, USA, 2024. Association for Computing Machinery.

[6] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Evaluating correctness and faithfulness of instruction-following models for question answering. *Trans. Assoc. Comput. Linguistics*, 12:681–699, 2024.

[7] Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Uttama Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. MEGA: multilingual evaluation of generative AI. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4232–4267. Association for Computational Linguistics, 2023.

[8] Sionic AI. Nano-BEIR: A multilingual information retrieval benchmark with quality-enhanced queries, 2025.

[9] Voyage AI. rerank-2 & rerank-2-lite: the next generation of voyage multilingual rerankers, 2024.

[10] Yamen Ajjour, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. Data acquisition for argument search: The args.me corpus. In *Proceedings of the 42nd German Conference on AI, KI 2019*, pages 48–59. Springer, 2019.

[11] Marwah Alaofi, Negar Arabzadeh, Charles LA Clarke, and Mark Sanderson. Generative information retrieval evaluation. In *Information Access in the Era of Generative AI*, pages 135–159. Springer, 2024.

[12] Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffrey Dalton, and Leif Azzopardi. TREC iKAT 2023: A Test Collection for Evaluating Conversational and Interactive Knowledge Assistants. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 819–829, New York, NY, USA, 2024. Association for Computing Machinery.

[13] Loubna Ben allal, Anton Lozhkov, Elie Bakouch, Gabriel Martin Blazquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Agustín Piqueres Lajarín, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan Son Nguyen, Ben Burtenshaw, Clémentine Fourrier, Haojun Zhao, Hugo Larcher, Mathieu Morlon, Cyril Zakka, Colin Raffel, Leandro Von Werra, and Thomas Wolf. SmolLM2: When Smol goes big — data-centric training of a fully open small language model. In *Second Conference on Language Modeling*, 2025.

[14] Giambattista Amati. Frequentist and bayesian approach to information retrieval. In *Proceedings of the 28th European Conference on IR Research, ECIR 2006*, volume 3936 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2006.

[15] Samuel Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. QAMPARI: A benchmark for open-domain questions with many answers. In Sebastian Gehrmann, Alex Wang, João Sedoc, Elizabeth Clark, Kaustubh Dhole, Khyathi Raghavi Chandu, Enrico Santus, and Hooman Sedghamiz, editors, *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 97–110, Singapore, December 2023. Association for Computational Linguistics.

[16] Bang An, Shiyue Zhang, and Mark Dredze. RAG LLMs are not safer: A safety analysis of retrieval-augmented generation for large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5444–5474, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[17] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha

Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. PaLM 2 Technical Report. *CoRR*, abs/2305.10403, 2023.

[18] Negar Arabzadeh and Charles L. A. Clarke. A comparison of methods for evaluating generative IR. *CoRR*, abs/2404.04044, 2024.

[19] Negar Arabzadeh and Charles L. A. Clarke. A human-AI comparative analysis of prompt sensitivity in LLM-based relevance judgment. In Nicola Ferro, Maria Maistro, Gabriella Pasi, Omar Alonso, Andrew Trotman, and Suzan Verberne, editors, *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 2784–2788. ACM, 2025.

[20] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles L. A. Clarke. Shallow pooling for sparse labels. *Inf. Retr. J.*, 25(4):365–385, 2022.

[21] 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.

[22] Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan N. Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. Aya 23: Open weight releases to further multilingual progress. *CoRR*, abs/2405.15032, 2024.

[23] 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.

[24] Akari Asai, Timo Schick, Patrick S. H. Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. Task-aware retrieval with instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3650–3675. ACL, 2023.

[25] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

[26] Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. One question answering model for many languages with cross-lingual dense passage retrieval. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 7547–7560, 2021.

[27] Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. Faithfulness tests for natural language explanations. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 283–294. Association for Computational Linguistics, 2023.

[28] Ehud Alexander Avner, Noam Ordan, and Shuly Wintner. Identifying translationese at the word and sub-word level. *Digital Scholarship in the Humanities*, 31(1):30–54, 2016.

[29] 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. *CoRR*, abs/1611.09268, 2016.

[30] Mehmet Selman Baysan and Tunga Gungor. TR-MTEB: A comprehensive benchmark and embedding model suite for Turkish sentence representations. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8867–8887, Suzhou, China, November 2025. Association for Computational Linguistics.

[31] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. In *First Conference on Language Modeling*, 2024.

[32] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 610–623. ACM, 2021.

[33] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly, 2009.

[34] Valeria Bolotova, Vladislav Blinov, Falk Scholer, W. Bruce Croft, and Mark Sanderson. A non-factoid question-answering taxonomy. In Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1196–1207. ACM, 2022.

[35] Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. Overview of Touché 2020: Argument retrieval. In *Proceedings of the 11th International Conference of the CLEF Association, CLEF 2020*, pages 384–395. Springer, 2020.

[36] Alexander Bondarenko, Maik Fröbe, Johannes Kiesel, Ferdinand Schlatt, Valentin Barriere, Brian Ravenet, Léo Hemamou, Simon Luck, Jan Heinrich Reimer, Benno Stein, Martin Potthast, and Matthias Hagen. Overview of Touché 2023: Argument and causal retrieval. In *Proceedings of the 14th International Conference of the CLEF Association, CLEF 2023*, pages 507–530. Springer, 2023.

[37] Alexander Bondarenko, Maik Fröbe, Johannes Kiesel, Shahbaz Syed, Timon Gurcke, Meriem Beloucif, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. Overview of Touché 2022: Argument retrieval. In Alberto Barrón-Cedeño, Giovanni Da San Martino, Mirko Degli Esposti, Fabrizio Sebastiani, Craig Macdonald, Gabriella Pasi, Allan Hanbury, Martin Potthast, Guglielmo Faggioli, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 311–336, Cham, 2022. Springer International Publishing.

[38] Alexander Bondarenko, Maik Fröbe, Jan Heinrich Reimer, Benno Stein, Michael Völske, and Matthias Hagen. Axiomatic retrieval experimentation with ir_axioms.

In Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3131–3140. ACM, 2022.

[39] Alexander Bondarenko, Lukas Gienapp, Maik Fröbe, Meriem Beloucif, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. Overview of Touché 2021: Argument retrieval. In K. Selçuk Candan, Bogdan Ionescu, Lorraine Goeuriot, Birger Larsen, Henning Müller, Alexis Joly, Maria Maistro, Florina Piroi, Guglielmo Faggioli, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 450–467, Cham, 2021. Springer International Publishing.

[40] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. InPars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2387–2392, New York, NY, USA, 2022. Association for Computing Machinery.

[41] Luiz Henrique Bonifacio, Israel Campiotti, Roberto de Alencar Lotufo, and Rodrigo Frassetto Nogueira. mMARCO: A multilingual version of MS MARCO passage ranking dataset. *CoRR*, abs/2108.13897, 2021.

[42] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 2022.

[43] Leonid Boytsov, Preksha Patel, Vivek Sourabh, Riddhi Nisar, Sayani Kundu, Ramya Ramanathan, and Eric Nyberg. InPars-Light: Cost-effective unsupervised training of efficient rankers. *Transactions on Machine Learning Research*, 2024. Reproducibility Certification.

[44] Andrei Z. Broder. On the resemblance and containment of documents. In Bruno Carpentieri, Alfredo De Santis, Ugo Vaccaro, and James A. Storer, editors, *Compression and Complexity of SEQUENCES 1997, Positano, Amalfitan Coast, Salerno, Italy, June 11-13, 1997, Proceedings*, pages 21–29. IEEE, 1997.

[45] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[46] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. Bias and the limits of pooling for large collections. *Inf. Retr.*, 10(6):491–508, December 2007.

[47] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 25–32, New York, NY, USA, 2004. Association for Computing Machinery.

[48] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. *SIGIR Forum*, 51(2):235–242, August 2017.

[49] Yinqiong Cai, Jiafeng Guo, Yixing Fan, Qingyao Ai, Ruqing Zhang, and Xueqi Cheng. Hard negatives or false negatives: Correcting pooling bias in training neural ranking models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 118–127, New York, NY, USA, 2022. Association for Computing Machinery.

[50] Arthur Câmara and Claudia Hauff. Diagnosing BERT with retrieval heuristics. In Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, volume 12035 of *Lecture Notes in Computer Science*, pages 605–618. Springer, 2020.

[51] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA, 1998. Association for Computing Machinery.

[52] Ramraj Chandradevan, Kaustubh Dhole, and Eugene Agichtein. DUQGen: Effective unsupervised domain adaptation of neural rankers by diversifying synthetic query generation. 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 1: Long Papers)*, pages 7437–7451, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[53] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pretraining tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*, 2020.

[54] Harrison Chase. LangChain, 2022.

[55] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In Regina Barzilay and Min-Yen Kan, editors, *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.

[56] Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[57] Haonan Chen, Liang Wang, Nan Yang, Yutao Zhu, Ziliang Zhao, Furu Wei, and Zhicheng Dou. Little giants: Synthesizing high-quality embedding data at scale. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1392–1411, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[58] Jianlyu Chen, Nan Wang, Chaofan Li, Bo Wang, Shitao Xiao, Han Xiao, Hao Liao, Defu Lian, and Zheng Liu. AIR-Bench: Automated heterogeneous information retrieval benchmark. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19991–20022, Vienna, Austria, July 2025. Association for Computational Linguistics.

[59] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[60] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[61] Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

[62] Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. BrowseComp-Plus: A More Fair and Transparent Evaluation Benchmark of Deep-Research Agent. *CoRR*, abs/2508.06600, 2025.

[63] Artem N. Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. TARGER: Neural argument mining at your fingertips. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 195–200. ACL, 2019.

[64] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. *CoRR*, abs/2403.04132, 2024.

[65] Nadezhda Chirkova, David Rau, Hervé Déjean, Thibault Formal, Stéphane Clinchant, and Vassilina Nikoulina. Retrieval-augmented generation in multilingual settings. In Sha Li, Manling Li, Michael JQ Zhang, Eunsol Choi, Mor Geva, Peter Hase, and Heng Ji, editors, *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 177–188, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[66] Pranjal A Chitale, Bishal Santra, Yashoteja Prabhu, and Amit Sharma. Evaluating the effectiveness and scalability of LLM-based data augmentation for retrieval. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17581–17617, Suzhou, China, November 2025. Association for Computational Linguistics.

[67] Chanyeol Choi, Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, and Jy-yong Sohn. Linq-Embed-Mistral technical report. *CoRR*, abs/2412.03223, 2024.

[68] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. QuAC: Question answering in context. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium, oct-nov 2018. Association for Computational Linguistics.

[69] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[70] Elizabeth Clark, Shruti Rijhwani, Sebastian Gehrmann, Joshua Maynez, Roee Aharoni, Vitaly Nikolaev, Thibault Sellam, Aditya Siddhant, Dipanjan Das, and Ankur P. Parikh. SEAHORSE: A multilingual, multifaceted dataset for summarization evaluation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9397–9413. Association for Computational Linguistics, 2023.

[71] 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.

[72] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 659–666, New York, NY, USA, 2008. Association for Computing Machinery.

[73] Cohere. Introducing Command R+: A scalable LLM built for business, 2024.

[74] 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 Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics, 2020.

[75] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. Overview of the TREC 2021 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC, May 2022.

[76] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. Overview of the TREC 2022 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC, March 2023.

[77] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Ellen M. Voorhees, and Ian Soboroff. TREC deep learning track: Reusable test collections in the large data regime. In *Proceedings of the 44th International ACM SIGIR Conference on*

Research and Development in Information Retrieval, SIGIR '21, page 2369–2375, New York, NY, USA, 2021. Association for Computing Machinery.

[78] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. Overview of the TREC 2023 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC, February 2024.

[79] María Andrea Cruz Blandón, Jayasimha Talur, Bruno Charron, Dong Liu, Saab Mansour, and Marcello Federico. MEMERAG: A multilingual end-to-end meta-evaluation benchmark for retrieval augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22577–22595, Vienna, Austria, July 2025. Association for Computational Linguistics.

[80] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for RAG systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 719–729, New York, NY, USA, 2024. Association for Computing Machinery.

[81] Sunhao Dai, Weihao Liu, Yuqi Zhou, Liang Pang, Rongju Ruan, Gang Wang, Zhenhua Dong, Jun Xu, and Ji-Rong Wen. Cocktail: A comprehensive information retrieval benchmark with LLM-generated documents integration. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7052–7074, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[82] Zhuyun Dai and Jamie Callan. Context-aware sentence/passage term importance estimation for first stage retrieval. *CoRR*, abs/1910.10687, 2019.

[83] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*, 2023.

[84] Tadele T. Damessie, Falk Scholer, and J. Shane Culpepper. The influence of topic difficulty, relevance level, and document ordering on relevance judging. In *Proceedings*

of the 21st Australasian Document Computing Symposium, ADCS '16, page 41–48, New York, NY, USA, 2016. Association for Computing Machinery.

[85] Hoa Trang Dang and Jimmy Lin. Different structures for evaluating answers to complex questions: pyramids won't topple, and neither will human assessors. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 768–775, Prague, Czech Republic, 2007.

[86] 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.

[87] Laxman Dhulipala, Majid Hadian, Rajesh Jayaram, Jason Lee, and Vahab Mirrokni. MUVERA: Multi-vector retrieval via fixed dimensional encoding. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 101042–101073. Curran Associates, Inc., 2024.

[88] Laura Dietz. A workbench for autograding retrieve/generate systems. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 1963–1972. ACM, 2024.

[89] Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. Climate-FEVER: A dataset for verification of real-world climate claims. In *Tackling Climate Change with Machine Learning workshop at NeurIPS 2020*, 2020.

[90] Xinya Du and Heng Ji. Retrieval-augmented generative question answering for event argument extraction. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4649–4666. Association for Computational Linguistics, 2022.

[91] Dheeru Dua, Emma Strubell, Sameer Singh, and Pat Verga. To adapt or to annotate: Challenges and interventions for domain adaptation in open-domain question

answering. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14429–14446. Association for Computational Linguistics, 2023.

[92] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, and 520 others. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

[93] Sauleh Eetemadi and Kristina Toutanova. Asymmetric features of human generated translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 159–164, Doha, Qatar, October 2014.

[94] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Veysel Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Suppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal A Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Mariya Hendriksen, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri K, Maksimova Anna, Silvan Wehrli, Maria Tikhonova, Henil Shalin Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Validad Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. MMTEB: Massive multilingual text embedding benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025.

[95] Kenneth Enevoldsen, Márton Kardos, Niklas Muennighoff, and Kristoffer Laigaard Nielbo. The scandinavian embedding benchmarks: Comprehensive assessment of multilingual and monolingual text embedding. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural*

Information Processing Systems, volume 37, pages 40336–40358. Curran Associates, Inc., 2024.

[96] Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In Nikolaos Aletras and Orphee De Clercq, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta, March 2024. Association for Computational Linguistics.

[97] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '23, page 39–50, New York, NY, USA, 2023. Association for Computing Machinery.

[98] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: long form question answering. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics, 2019.

[99] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA, 2024. Association for Computing Machinery.

[100] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In Mark Sanderson, Kalervo Järvelin, James Allan, and Peter Bruza, editors, *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pages 49–56. ACM, 2004.

[101] Hui Fang, Tao Tao, and ChengXiang Zhai. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.*, 29(2):7:1–7:42, 2011.

[102] Hui Fang and ChengXiang Zhai. Semantic term matching in axiomatic approaches to information retrieval. In Efthimis N. Efthimiadis, Susan T. Dumais, David

Hawking, and Kalervo Järvelin, editors, *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 115–122. ACM, 2006.

[103] Naghmeh Farzi and Laura Dietz. Pencils down! automatic rubric-based evaluation of retrieve/generate systems. In Harrie Oosterhuis, Hannah Bast, and Chenyan Xiong, editors, *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2024, Washington, DC, USA, 13 July 2024*, pages 175–184. ACM, 2024.

[104] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics, 2022.

[105] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382, 1971.

[106] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE v2: Sparse lexical and expansion model for information retrieval. *CoRR*, abs/2109.10086, 2021.

[107] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2288–2292, New York, NY, USA, 2021.

[108] Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pages 2843–2853. ACL, 2022.

[109] Luyu Gao, Zhuyun Dai, and Jamie Callan. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,NAACL-HLT 2021*, pages 3030–3042. ACL, 2021.

[110] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. 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 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics.

[111] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Tevatron: An efficient and flexible toolkit for neural retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 3120–3124, New York, NY, USA, 2023. Association for Computing Machinery.

[112] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[113] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6465–6488. Association for Computational Linguistics, 2023.

[114] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3356–3369. Association for Computational Linguistics, 2020.

[115] Gemini. Gemini deep research. 2025.

[116] Lukas Gienapp, Benno Stein, Matthias Hagen, and Martin Potthast. Efficient pairwise annotation of argument quality. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5772–5781. Association for Computational Linguistics, 2020.

[117] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. Doc2query-: When less is more. In Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo, editors, *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR*

2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part II, volume 13981 of *Lecture Notes in Computer Science*, pages 414–422. Springer, 2023.

[118] Michael Günther, Saba Sturua, Mohammad Kalim Akram, Isabelle Mohr, Andrei Ungureanu, Bo Wang, Sedigheh Eslami, Scott Martens, Maximilian Werk, Nan Wang, and Han Xiao. jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval. In David Ifeoluwa Adelani, Catherine Arnett, Duygu Ataman, Tyler A. Chang, Hila Gonen, Rahul Raja, Fabian Schmidt, David Stap, and Jiayi Wang, editors, *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pages 531–550, Suzhuo, China, November 2025. Association for Computational Linguistics.

[119] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 2020.

[120] David Haines and W. Bruce Croft. Relevance feedback and inference networks. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, page 2–11, New York, NY, USA, 1993. Association for Computing Machinery.

[121] Rujun Han, Yuhao Zhang, Peng Qi, Yumo Xu, Jenyuan Wang, Lan Liu, William Yang Wang, Bonan Min, and Vittorio Castelli. RAG-QA arena: Evaluating domain robustness for long-form retrieval augmented question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4354–4374, Miami, Florida, November 2024.

[122] Donna Harman. *Information Retrieval Evaluation*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2011.

[123] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. DBpedia-Entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1265–1268, New York, NY, USA, 2017. Association for Computing Machinery.

[124] Hangfeng He, Hongming Zhang, and Dan Roth. Rethinking with retrieval: Faithful large language model inference. *CoRR*, abs/2301.00303, 2023.

[125] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[126] Tin Kam Ho. Random decision forests. In *Third International Conference on Document Analysis and Recognition, ICDAR 1995, August 14 - 15, 1995, Montreal, Canada. Volume I*, pages 278–282. IEEE Computer Society, 1995.

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

[128] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. *Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling*, page 113–122. Association for Computing Machinery, New York, NY, USA, 2021.

[129] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[130] Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. CQADupStack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, New York, NY, USA, 2015. Association for Computing Machinery.

[131] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[132] 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.

[133] Xinshuo Hu, Zifei Shan, Xinping Zhao, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, Haofen Wang, Jun Yu, and Min Zhang. KaLM-Embedding: Superior training data brings a stronger embedding model. *CoRR*, abs/2501.01028, 2025.

[134] Xinyu Hua, Zhe Hu, and Lu Wang. Argument generation with retrieval, planning, and realization. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2661–2672. Association for Computational Linguistics, 2019.

[135] David R. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384 – 406, 2004.

[136] Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. CodeSearchNet challenge: Evaluating the state of semantic code search. *CoRR*, abs/1909.09436, 2019.

[137] Makoto Iwayama, Atsushi Fujii, Noriko Kando, and Akihiko Takano. Overview of patent retrieval task at NTCIR-3. In *Proceedings of the ACL-2003 Workshop on Patent Corpus Processing*, pages 24–32, „ July 2003. Association for Computational Linguistics.

[138] 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.

[139] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics, 2021.

[140] Mathew Jacob, Erik Lindgren, Matei Zaharia, Michael Carbin, Omar Khattab, and Andrew Drozdov. Drowning in Documents: Consequences of Scaling Reranker Inference. *CoRR*, abs/2411.11767, 2024.

[141] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

[142] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct 2002.

[143] Vitor Jeronymo, Luiz Henrique Bonifacio, Hugo Queiroz Abonizio, Marzieh Fadaee, Roberto de Alencar Lotufo, Jakub Zavrel, and Rodrigo Frassetto Nogueira. InPars-v2: Large language models as efficient dataset generators for information retrieval. *CoRR*, abs/2301.01820, 2023.

[144] Qi Jia, Siyu Ren, Yizhu Liu, and Kenny Q. Zhu. Zero-shot faithfulness evaluation for text summarization with foundation language model. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 11017–11031. Association for Computational Linguistics, 2023.

[145] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023.

[146] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024.

[147] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025.

[148] Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. FlashRAG: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 737–740, New York, NY, USA, 2025. Association for Computing Machinery.

[149] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502, 2004.

[150] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[151] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online, July 2020.

[152] Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. HAGRID: A Human-LLM Collaborative Dataset for Generative Information-Seeking with Attribution. *CoRR*, abs/2307.16883, 2023.

[153] Ehsan Kamalloo, Xinyu Zhang, Odunayo Ogundepo, Nandan Thakur, David Alfonso-Hermelo, Mehdi Rezagholizadeh, and Jimmy Lin. Evaluating embedding APIs for information retrieval. In Sunayana Sitaram, Beata Beigman Klebanov, and Jason D. Williams, editors, *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 518–526. Association for Computational Linguistics, 2023.

[154] Gregory Kamradt. Needle In A Haystack - pressure testing LLMs, 2023.

[155] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[156] 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, November 2020.

[157] Diane Kelly and Jimmy Lin. Overview of the TREC 2006 ciQA task. *SIGIR Forum*, 41(1):107–116, June 2007.

[158] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*, 2024.

[159] Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA, 2020. Association for Computing Machinery.

[160] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online, June 2021. Association for Computational Linguistics.

[161] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 05 2018.

[162] Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[163] Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. AQuaMuSe: Automatically generating datasets for query-based multi-document summarization. *CoRR*, abs/2010.12694, 2020.

[164] Oren Kurland and J. Shane Culpepper. Fusion in information retrieval: SIGIR 2018 half-day tutorial. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1383–1386, New York, NY, USA, 2018. Association for Computing Machinery.

[165] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019.

[166] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[167] Philippe Laban, Wojciech Kryscinski, Divyansh Agarwal, Alexander R. Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. SummEdits: Measuring LLM ability at factual reasoning through the lens of summarization. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9662–9676. Association for Computational Linguistics, 2023.

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

[169] Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. SPLADE-v3: New baselines for SPLADE. *CoRR*, abs/2403.06789, 2024.

[170] Carlos Lassance, Ronak Pradeep, and Jimmy Lin. naverloo @ TREC Deep Learning and NeuCLIR 2023: As easy as zero, one, two, three — Cascading dual encoders, mono, duo, and listo for ad-hoc retrieval. In *Text REtrieval Conference (TREC)*. NIST, February 2024.

[171] Dawn Lawrie, Efsun Kayi, Eugene Yang, James Mayfield, Douglas W. Oard, and Scott Miller. Generate-distill: Training cross-language ir models with synthetically-generated data. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 2926–2930, New York, NY, USA, 2025. Association for Computing Machinery.

[172] Dawn J. Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldaini, and Eugene Yang. Overview of the TREC 2022 NeuCLIR track. In Ian Soboroff and Angela Ellis, editors, *Proceedings of the Thirty-First Text REtrieval*

Conference, TREC 2022, online, November 15-19, 2022, volume 500-338 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022.

[173] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[174] Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, Feng Han, Andreas Doumanoglou, Nithi Gupta, Fedor Moiseev, Cathy Yip, Aashi Jain, Simon Baumgartner, Shahrokh Shahi, Frank Palma Gomez, Sandeep Mariserla, Min Choi, Parashar Shah, Sonam Goenka, Ke Chen, Ye Xia, Koert Chen, Sai Meher Karthik Duddu, Yichang Chen, Trevor Walker, Wenlei Zhou, Rakesh Ghiya, Zach Gleicher, Karan Gill, Zhe Dong, Mojtaba Seyedhosseini, Yun-Hsuan Sung, Raphael Hoffmann, and Tom Duerig. Gemini embedding: Generalizable embeddings from Gemini. *CoRR*, abs/2503.07891, 2025.

[175] Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Zhao. Rethinking the role of token retrieval in multi-vector retrieval. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[176] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernández Ábrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile text embeddings distilled from large language models. *CoRR*, abs/2403.20327, 2024.

[177] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics.

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

[179] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating cross-lingual extractive question answering. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online, July 2020. Association for Computational Linguistics.

[180] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[181] Chaofan Li, Minghao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Defu Lian, Yingxia Shao, and Zheng Liu. Making text embedders few-shot learners. In *The Thirteenth International Conference on Learning Representations*, 2025.

[182] Hongyu Li, Seohyun Kim, and Satish Chandra. Neural code search evaluation dataset. *CoRR*, abs/1908.09804, 2019.

[183] 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, ACL 2023*, pages 11891–11907. ACL, 2023.

[184] Xiangyang Li, Kuicai Dong, Yi Quan Lee, Wei Xia, Hao Zhang, Xinyi Dai, Yasheng Wang, and Ruiming Tang. CoIR: A comprehensive benchmark for code information retrieval models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22074–22091, Vienna, Austria, July 2025. Association for Computational Linguistics.

[185] Yulong Li, Martin Franz, Md Arafat Sultan, Bhavani Iyer, Young-Suk Lee, and Avirup Sil. Learning cross-lingual IR from an English retriever. In *Proceedings of the 2022*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4428–4436, Seattle, United States, July 2022. Association for Computational Linguistics.

[186] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *CoRR*, abs/2308.03281, 2023.

[187] Davis Liang, Peng Xu, Siamak Shakeri, Cícero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. Embedding-based zero-shot retrieval through query generation. *CoRR*, abs/2009.10270, 2020.

[188] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[189] Jimmy Lin and Dina Demner-Fushman. Automatically evaluating answers to definition questions. In Raymond Mooney, Chris Brew, Lee-Feng Chien, and Katrin Kirchhoff, editors, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 931–938, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[190] Jimmy Lin and Dina Demner-Fushman. Methods for automatically evaluating answers to complex questions. *Inf. Retr.*, 9(5):565–587, November 2006.

[191] Jimmy Lin and Dina Demner-Fushman. Will pyramids built of nuggets topple over? In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 383–390, New York, New York, 2006.

[192] Jimmy Lin and Xueguang Ma. A few brief notes on DeepImpact, COIL, and a Conceptual Framework for information retrieval techniques. *CoRR*, abs/2106.14807, 2021.

[193] 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.

[194] Jimmy Lin and Pengyi Zhang. Deconstructing nuggets: the stability and reliability of complex question answering evaluation. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, pages 327–334, Amsterdam, the Netherlands, 2007.

[195] Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. How to Train Your Dragon: Diverse augmentation towards generalizable dense retrieval. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6385–6400. Association for Computational Linguistics, 2023.

[196] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics, 2022.

[197] Aldo Lipani. On biases in information retrieval models and evaluation. *SIGIR Forum*, 52(2):172–173, 2018.

[198] Jerry Liu. LlamaIndex, 2022.

[199] Nelson Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore, December 2023. Association for Computational Linguistics.

[200] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[201] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics.

[202] Yi Liu, Matei Zaharia, and Ritendra Datta. Enhancing LLM-as-a-Judge with grading notes, 2024.

[203] Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. ChatQA: surpassing GPT-4 on conversational QA and RAG. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.

[204] 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.

[205] Ehsan Lotfi, Nikolay Banar, and Walter Daelemans. BEIR-NL: Zero-shot information retrieval benchmark for the Dutch language. In Serge Sharoff, Ayla Rigouts Terryn, Pierre Zweigenbaum, and Reinhard Rapp, editors, *Proceedings of the 18th Workshop on Building and Using Comparable Corpora (BUCC)*, pages 36–45, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[206] Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, MING GONG, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie LIU. CodeXGLUE: A machine learning benchmark dataset for code understanding and generation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[207] Marco Lui and Timothy Baldwin. langid.py: An off-the-shelf language identification tool. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 25–30. The Association for Computer Linguistics, 2012.

[208] Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. CRUD-RAG: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Trans. Inf. Syst.*, 43(2), January 2025.

[209] Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1075–1088, Online, April 2021. Association for Computational Linguistics.

[210] Xueguang Ma, Luyu Gao, Shengyao Zhuang, Jiaqi Samantha Zhan, Jamie Callan, and Jimmy Lin. Tevatron 2.0: Unified document retrieval toolkit across scale, language, and modality. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 4061–4065, New York, NY, USA, 2025. Association for Computing Machinery.

[211] 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.

[212] Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. Zero-shot listwise document reranking with a large language model. *CoRR*, abs/2305.02156, 2023.

[213] Sean MacAvaney and Luca Soldaini. One-shot labeling for automatic relevance estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, July 2023.

[214] Joel Mackenzie, Matthias Petri, and Alistair Moffat. A sensitivity analysis of the MSMARCO passage collection. *CoRR*, abs/2112.03396, 2021.

[215] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. WWW'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1941–1942, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[216] Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. ExpertQA: Expert-curated questions and attributed answers. 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 1: Long Papers)*, pages 3025–3045, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

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

[218] James Mayfield, Eugene Yang, Dawn Lawrie, Sean MacAvaney, Paul McNamee, Douglas W. Oard, Luca Soldaini, Ian Soboroff, Orion Weller, Efsun Kayi, Kate

Sanders, Marc Mason, and Noah Hibbler. On the evaluation of machine-generated reports. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

[219] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. On faithfulness and factuality in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1906–1919. Association for Computational Linguistics, 2020.

[220] Hansa Meghwani, Amit Agarwal, Priyaranjan Pattnayak, Hitesh Laxmichand Patel, and Srikant Panda. Hard negative mining for domain-specific retrieval in enterprise systems. In Georg Rehm and Yunyao Li, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1013–1026, Vienna, Austria, July 2025. Association for Computational Linguistics.

[221] Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. Query performance prediction using relevance judgments generated by large language models. *ACM Trans. Inf. Syst.*, 43(4), July 2025.

[222] Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. Arctic-Embed: Scalable, efficient, and accurate text embedding models. *CoRR*, abs/2405.05374, 2024.

[223] Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Cristian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. Gemma: Open models based on Gemini research and technology. *CoRR*, abs/2403.08295, 2024.

[224] Microsoft. Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web, 2023.

[225] Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. FaithEval: Can Your Language Model Stay Faithful

to Context, Even If "The Moon is Made of Marshmallows". In *The Thirteenth International Conference on Learning Representations*, 2025.

[226] Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andrés Codas, Clarisse Simões, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. Orca 2: Teaching small language models how to reason. *CoRR*, abs/2311.11045, 2023.

[227] Fedor Moiseev, Gustavo Hernandez Abrego, Peter Dornbach, Imed Zitouni, Enrique Alfonseca, and Zhe Dong. SamToNe: Improving contrastive loss for dual encoder retrieval models with same tower negatives. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12028–12037, Toronto, Canada, July 2023. Association for Computational Linguistics.

[228] Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. Improving text embedding models with positive-aware hard-negative mining. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 2169–2178, New York, NY, USA, 2025. Association for Computing Machinery.

[229] John X Morris and Alexander M Rush. Contextual document embeddings. In *The Thirteenth International Conference on Learning Representations*, 2025.

[230] Niklas Muennighoff. SGPT: GPT sentence embeddings for semantic search, 2022.

[231] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[232] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[233] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen

Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training. *CoRR*, abs/2201.10005, 2022.

[234] Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA, may 2 - may 7 2004. Association for Computational Linguistics.

[235] Xuan-Phi Nguyen, Shrey Pandit, Senthil Purushwalkam, Austin Xu, Hailin Chen, Yifei Ming, Zixuan Ke, Silvio Savarese, Caiming Xong, and Shafiq Joty. SFR-RAG: towards contextually faithful llms. *CoRR*, abs/2409.09916, 2024.

[236] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. 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, EMNLP 2022*, pages 9844–9855. ACL, 2022.

[237] Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. MixEval: Deriving wisdom of the crowd from LLM benchmark mixtures. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[238] Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[239] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *CoRR*, abs/1901.04085, 2019.

[240] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, November 2020.

[241] Rodrigo Nogueira and Jimmy Lin. From doc2query to doctttttquery. *Online preprint*, 6:2, 2019.

[242] Zach Nussbaum, John Xavier Morris, Andriy Mulyar, and Brandon Duderstadt. Nomic Embed: Training a reproducible long context text embedder. *Transactions on Machine Learning Research*, 2025. Reproducibility Certification.

[243] Odunayo Ogundepo, Tajuddeen R. Gwadabe, Clara E. Rivera, Jonathan H. Clark, Sebastian Ruder, David Ifeoluwa Adelani, Bonaventure F. P. Dossou, Abdou Aziz Diop, Claytone Sikasote, Gilles Hacheme, Happy Buzaaba, Ignatius Ezeani, Rooweither Mabuya, Salomey Osei, Chris Emezue, Albert Njoroge Kahira, Shamsuddeen Hassan Muhammad, Akintunde Oladipo, Abraham Toluwase Owodunni, Atnafu Lambebo Tonja, Iyanuoluwa Shode, Akari Asai, Tunde Oluwaseyi Ajayi, Clemencia Siro, Steven Arthur, Mofetoluwa Adeyemi, Orevaoghene Ahia, Anuoluwapo Aremu, Oyinkansola Awosan, Chiamaka Chukwuneke, Bernard Opoku, Awokoya Ayodele, Verrah Otiende, Christine Mwase, Boyd Sinkala, Andre Niyongabo Rubungo, Daniel A. Ajisafe, Emeka Felix Onwuegbuzia, Habib Mbow, Emile Niyomutabazi, Eunice Mukonde, Falalu Ibrahim Lawan, Ibrahim Said Ahmad, Jesujoba O. Alabi, Martin Namukombo, Mbonu Chinedu, Mofya Phiri, Neo Putini, Ndumiso Mngoma, Priscilla A. Amouk, Ruqayya Nasir Iro, and Sonia Adhiambo. AfriQA: Cross-lingual open-retrieval question answering for African languages. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14957–14972, Singapore, December 2023. Association for Computational Linguistics.

[244] Barlas Oguz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Scott Yih, Sonal Gupta, and Yashar Mehdad. Domain-matched pre-training tasks for dense retrieval. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1524–1534, Seattle, United States, July 2022. Association for Computational Linguistics.

[245] Hanseok Oh, Hyunji Lee, Seonghyeon Ye, Haebin Shin, Hansol Jang, Changwook Jun, and Minjoon Seo. INSTRUCTIR: A benchmark for instruction following of information retrieval models. *CoRR*, abs/2402.14334, 2024.

[246] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[247] OpenAI. Hello GPT-4o, 2024.

[248] OpenAI. Introducing deep research. 2025.

[249] Arnold Overwijk, Chenyan Xiong, and Jamie Callan. ClueWeb22: 10 billion web documents with rich information. In Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3360–3362. ACM, 2022.

[250] Paul Owoicho, Jeff Dalton, Mohammad Aliannejadi, Leif Azzopardi, Johanne R. Trippas, and Svitlana Vakulenko. TREC CAsT 2022: Going beyond user ask and system retrieve with initiative and response generation. In Ian Soboroff and Angela Ellis, editors, *Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022, online, November 15-19, 2022*, volume 500-338 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022.

[251] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[252] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[253] Virgil Pavlu, Shahzad Rajput, Peter B. Golbus, and Javed A. Aslam. IR system evaluation using nugget-based test collections. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, page 393–402, New York, NY, USA, 2012. Association for Computing Machinery.

[254] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick S. H. Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2523–2544. Association for Computational Linguistics, 2021.

[255] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. MAUVE: measuring the gap between neural text and

human text using divergence frontiers. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4816–4828, 2021.

[256] Martin Potthast, Lukas Gienapp, Florian Euchner, Nick Heilenkötter, Nico Weidmann, Henning Wachsmuth, Benno Stein, and Matthias Hagen. Argument search: Assessing argument relevance. In *Proceedings of the 42nd International Conference on Research and Development in Information Retrieval, SIGIR 2019*, pages 1117–1120. ACM, 2019.

[257] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. RankZephyr: Effective and robust zero-shot listwise reranking is a breeze! *CoRR*, abs/2312.02724, 2023.

[258] Ronak Pradeep, Nandan Thakur, Sahel Sharifymoghaddam, Eric Zhang, Ryan Nguyen, Daniel Campos, Nick Craswell, and Jimmy Lin. Ragnarök: A reusable RAG framework and baselines for TREC 2024 Retrieval-Augmented Generation Track. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part I*, page 132–148, Berlin, Heidelberg, 2025. Springer-Verlag.

[259] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. Initial nugget evaluation results for the TREC 2024 RAG track with the AutoNuggetizer framework. *CoRR*, abs/2411.09607, 2024.

[260] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. The great nugget recall: Automating fact extraction and RAG evaluation with large language models. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 180–190, New York, NY, USA, 2025. Association for Computing Machinery.

[261] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[262] Jiazuo Qiu and Deyi Xiong. Generating highly relevant questions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 5983–5987, Hong Kong, China, November 2019. Association for Computational Linguistics.

[263] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online, June 2021. Association for Computational Linguistics.

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

[265] Zackary Rackauckas, Arthur Câmara, and Jakub Zavrel. Evaluating RAG-fusion with RAGElo: an automated Elo-based framework. In Clemencia Siro, Mohammad Aliannejadi, Hossein A. Rahmani, Nick Craswell, Charles L. A. Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz, editors, *Proceedings of The First Workshop on Large Language Models for Evaluation in Information Retrieval (LLM4Eval 2024) co-located with 10th International Conference on Online Publishing (SIGIR 2024), Washington D.C., USA, July 18, 2024*, volume 3752 of *CEUR Workshop Proceedings*, pages 92–112. CEUR-WS.org, 2024.

[266] 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.

[267] Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles L. A. Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. LLMJudge: LLMs for relevance judgments. In Clemencia Siro, Mohammad Aliannejadi, Hossein A. Rahmani, Nick Craswell, Charles L. A. Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz, editors, *Proceedings of The First Workshop on Large Language Models for Evaluation in Information Retrieval (LLM4Eval 2024) co-located with 10th International Conference on Online Publishing (SIGIR 2024), Washington D.C., USA, July 18, 2024*, volume 3752 of *CEUR Workshop Proceedings*, pages 1–3. CEUR-WS.org, 2024.

[268] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao, editors,

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[269] 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.

[270] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

[271] Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. Learning to retrieve passages without supervision. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2687–2700, Seattle, United States, July 2022. Association for Computational Linguistics.

[272] David Rau, Hervé Déjean, Nadezhda Chirkova, Thibault Formal, Shuai Wang, Stéphane Clinchant, and Vassilina Nikoulina. BERGEN: A benchmarking library for retrieval-augmented generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7640–7663, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[273] Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online, June 2021. Association for Computational Linguistics.

[274] Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 05 2019.

[275] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on*

Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[276] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4512–4525. Association for Computational Linguistics, 2020.

[277] Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. Classification and clustering of arguments with contextualized word embeddings. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 567–578. Association for Computational Linguistics, 2019.

[278] Houxing Ren, Linjun Shou, Ning Wu, Ming Gong, and Daxin Jiang. Empowering dual-encoder with query generator for cross-lingual dense retrieval. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3107–3121, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[279] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 2825–2835. ACL, 2021.

[280] Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin,

Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024.

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

[282] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994*, pages 109–126. NIST, 1994.

[283] Sara Rosenthal, Avirup Sil, Radu Florian, and Salim Roukos. CLAPnq: Cohesive long-form answers from passages in natural questions for RAG systems. *Transactions of the Association for Computational Linguistics*, 13:53–72, 2025.

[284] Corbin Rosset, Ho-Lam Chung, Guanghui Qin, Ethan Chau, Zhuo Feng, Ahmed Awadallah, Jennifer Neville, and Nikhil Rao. Researchy questions: A dataset of multi-perspective, decompositional questions for deep research. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 3712–3722, New York, NY, USA, 2025. Association for Computing Machinery.

[285] Corby Rosset, Bhaskar Mitra, Chenyan Xiong, Nick Craswell, Xia Song, and Saurabh Tiwary. An axiomatic approach to regularizing neural ranking models. In Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer, editors, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 981–984. ACM, 2019.

[286] Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. LAReQA: Language-agnostic answer retrieval from a multilingual pool. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5919–5930, Online, November 2020. Association for Computational Linguistics.

[287] Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. RAGChecker: a fine-grained framework for diagnosing retrieval-augmented generation. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.

[288] Sebastian Ruder, Jonathan H. Clark, Alexander Gutkin, Mihir Kale, Min Ma, Massimo Nicosia, Shruti Rijhwani, Parker Riley, Jean-Michel A Sarr, Xinyi Wang, John Wieting, Nitish Gupta, Anna Katanova, Christo Kirov, Dana L. Dickinson, Brian Roark, Bidisha Samanta, Connie Tao, David I. Adelani, Vera Axelrod, Isaac Caswell, Colin Cherry, Dan Garrette, Reeve Ingle, Melvin Johnson, Dmitry Panteleev, and Partha Talukdar. XTREME-UP: A user-centric scarce-data benchmark for under-represented languages. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1856–1884, Singapore, December 2023. Association for Computational Linguistics.

[289] Jon Saad-Falcon, Daniel Y. Fu, Simran Arora, Neel Guha, and Christopher Ré. Benchmarking and building long-context retrieval models with LoCo and M2-BERT. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[290] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. ARES: An automated evaluation framework for retrieval-augmented generation systems. 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 1: Long Papers)*, pages 338–354, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[291] Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Sultan, and Christopher Potts. UDAPDR: Unsupervised domain adaptation via LLM prompting and distillation of rerankers. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11265–11279, Singapore, December 2023. Association for Computational Linguistics.

[292] Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. Improving passage retrieval with zero-shot question generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang,

editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3781–3797. Association for Computational Linguistics, 2022.

[293] Alireza Salemi and Hamed Zamani. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2395–2400, New York, NY, USA, 2024. Association for Computing Machinery.

[294] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2020.

[295] Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. PLAID: An efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 1747–1756, New York, NY, USA, 2022. Association for Computing Machinery.

[296] 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.

[297] Keshav Santhanam, Jon Saad-Falcon, Martin Franz, Omar Khattab, Avi Sil, Radu Florian, Md Arafat Sultan, Salim Roukos, Matei Zaharia, and Christopher Potts. Moving beyond downstream task accuracy for information retrieval benchmarking. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11613–11628, Toronto, Canada, July 2023. Association for Computational Linguistics.

[298] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Search result diversification. *Foundations and Trends® in Information Retrieval*, 9(1):1–90, 2015.

[299] Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online, April 2021. Association for Computational Linguistics.

[300] Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. CCMatrix: Mining billions of high-quality parallel sentences on the

web. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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 6490–6500, Online, August 2021. Association for Computational Linguistics.

[301] Mahsa S. Shahshahani and Jaap Kamps. Argument retrieval from web. In *Proceedings of the11th International Conference of the CLEF Association, CLEF 2020*, pages 75–81. Springer, 2020.

[302] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR, 2023.

[303] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. 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 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[304] Noah Y. Siegel, Oana-Maria Camburu, Nicolas Heess, and María Pérez-Ortiz. The probabilities also matter: A more faithful metric for faithfulness of free-text explanations in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pages 530–546. Association for Computational Linguistics, 2024.

[305] Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. End-to-end training of multi-document reader and retriever for open-domain question answering. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25968–25981. Curran Associates, Inc., 2021.

[306] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In Hans-Peter Frei, Donna Harman, Peter Schäuble, and Ross Wilkinson,

editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96, August 18-22, 1996, Zurich, Switzerland (Special Issue of the SIGIR Forum)*, pages 21–29. ACM, 1996.

[307] Shangeetha Sivasothy, Scott Barnett, Stefanus Kurniawan, Zafaryab Rasool, and Rajesh Vasa. RAGProbe: An automated approach for evaluating RAG applications. *CoRR*, abs/2409.19019, 2024.

[308] Artem Snegirev, Maria Tikhonova, Maksimova Anna, Alena Fenogenova, and Aleksandr Abramov. The Russian-focused embedders' exploration: ruMTEB benchmark and Russian embedding model design. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 236–254, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[309] Tingyu Song, Guo Gan, Mingsheng Shang, and Yilun Zhao. IFIR: A comprehensive benchmark for evaluating instruction-following in expert-domain information retrieval. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10186–10204, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[310] Raphaël Sourty. NanoBEIR-Multilingual: Multilingual version of NanoBEIR for quick evaluation., 2025.

[311] Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. ArgumenText: Searching for arguments in heterogeneous sources. In Yang Liu, Tim Paek, and Manasi Patwardhan, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21–25, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[312] Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. Cross-topic argument mining from heterogeneous sources. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3664–3674. Association for Computational Linguistics, 2018.

[313] Felix Stahlberg. Neural machine translation: A review. *J. Artif. Intell. Res.*, 69:343–418, 2020.

[314] Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: factoid questions meet long-form answers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8273–8288. Association for Computational Linguistics, 2022.

[315] Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. Jina embeddings v3: Multilingual text encoder with low-rank adaptations. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part V*, page 123–129, Berlin, Heidelberg, 2025. Springer-Verlag.

[316] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada, July 2023. Association for Computational Linguistics.

[317] Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025.

[318] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[319] 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.

[320] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large

language models as re-ranking agents. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore, December 2023. Association for Computational Linguistics.

[321] Xichen Sun, Wenhan Chao, and Zhunchen Luo. Syntax and coherence - the effect on automatic argument quality assessment. In Lu Wang, Yansong Feng, Yu Hong, and Ruifang He, editors, *Natural Language Processing and Chinese Computing - 10th CCF International Conference, NLPCC 2021, Qingdao, China, October 13-17, 2021, Proceedings, Part II*, volume 13029 of *Lecture Notes in Computer Science*, pages 3–12. Springer, 2021.

[322] Manveer Singh Tamber, Ronak Pradeep, and Jimmy Lin. Scaling Down, LiTting Up: Efficient Zero-Shot Listwise Reranking with Seq2seq Encoder-Decoder Models. *CoRR*, abs/2312.16098, 2023.

[323] Gemma Team et al. Gemma 2: Improving open language models at a practical size, 2024.

[324] Nandan Thakur, Luiz Bonifacio, Crystina Zhang, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, and Jimmy Lin. "Knowing When You Don't Know": A multilingual relevance assessment dataset for robust retrieval-augmented generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12508–12526, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[325] Nandan Thakur, Suleman Kazi, Ge Luo, Jimmy Lin, and Amin Ahmad. MIRAGE-Bench: Automatic multilingual benchmark arena for retrieval-augmented generation systems. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 274–298, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[326] Nandan Thakur, Jimmy Lin, Sam Havens, Michael Carbin, Omar Khattab, and Andrew Drozdov. FreshStack: Building realistic benchmarks for evaluating retrieval on technical documents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[327] Nandan Thakur, Ronak Pradeep, Shivani Upadhyay, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. Assessing support for the TREC 2024 RAG Track: A large-scale comparative study of LLM and human evaluations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 2759–2763, New York, NY, USA, 2025. Association for Computing Machinery.

[328] Nandan Thakur, Nils Reimers, and Jimmy Lin. Injecting domain adaptation with learning-to-hash for effective and efficient zero-shot dense retrieval. *CoRR*, abs/2205.11498, 2022.

[329] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, 2021.

[330] Nandan Thakur, Kexin Wang, Iryna Gurevych, and Jimmy Lin. SPRINT: A unified toolkit for evaluating and demystifying zero-shot neural sparse retrieval. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete, editors, *Proceedings of the 46th International Conference on Research and Development in Information Retrieval, SIGIR 2023*, pages 2964–2974. ACM, 2023.

[331] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. Large language models can accurately predict searcher preferences. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 1930–1940. ACM, 2024.

[332] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[333] Assaf Toledo, Shai Gretz, Edo Cohen-Karlik, Roni Friedman, Elad Venezian, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. Automatic argument quality assessment - new datasets and methods. In Kentaro Inui, Jing Jiang, Vincent

Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5624–5634. Association for Computational Linguistics, 2019.

[334] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

[335] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics, 2023.

[336] TruLens. TruLens: Evaluate and Trace AI Agents, 2025.

[337] Shangqing Tu, Yuanchun Wang, Jifan Yu, Yuyang Xie, Yaran Shi, Xiaozhi Wang, Jing Zhang, Lei Hou, and Juanzi Li. R-Eval: A unified toolkit for evaluating domain knowledge of retrieval augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5813–5824, New York, NY, USA, 2024. Association for Computing Machinery.

[338] Saad Obaid Ul Islam, Anne Lauscher, and Goran Glavaš. How much do LLMs hallucinate across languages? on realistic multilingual estimation of LLM hallucination. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural*

Language Processing, pages 29077–29098, Suzhou, China, November 2025. Association for Computational Linguistics.

[339] Shivani Upadhyay, Ehsan Kamalloo, and Jimmy Lin. LLMs can patch up missing relevance judgments in evaluation. *CoRR*, abs/2405.04727, 2024.

[340] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Daniel Campos, Nick Craswell, Ian Soboroff, and Jimmy Lin. A large-scale study of relevance assessments with large language models using UMBRELA. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, ICTIR '25, page 358–368, New York, NY, USA, 2025. Association for Computing Machinery.

[341] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. UMBRELA: UMbrela is the (Open-Source Reproduction of the) Bing RELevance Assessor. *CoRR*, abs/2406.06519, 2024.

[342] Ahmet Üstün, Viraat Aryabumi, Zheng Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. Aya model: An instruction finetuned open-access multilingual language model. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15894–15939. Association for Computational Linguistics, 2024.

[343] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[344] Vectara. Introducing Open RAG Eval: The open-source framework for comparing RAG solutions, 2025.

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

[346] Michael Völske, Alexander Bondarenko, Maik Fröbe, Benno Stein, Jaspreet Singh, Matthias Hagen, and Avishek Anand. Towards axiomatic explanations for neural ranking models. In Faegheh Hasibi, Yi Fang, and Akiko Aizawa, editors, *ICTIR '21: The 2021 ACM SIGIR International Conference on the Theory of Information Retrieval, Virtual Event, Canada, July 11, 2021*, pages 13–22. ACM, 2021.

[347] Ellen Voorhees. I come not to bury cranfield, but to praise it. HCIR 2009: Bridging Human Computer Interaction and Information Retrieval, Washington DC, -1, 2009-10-26 2009.

[348] 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.

[349] Ellen M. Voorhees. Evaluating answers to definition questions. In *Companion Volume of the Proceedings of HLT-NAACL 2003 — Short Papers*, pages 109–111, Edmonton, Canada, 2003.

[350] Ellen M. Voorhees. Overview of the TREC 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, Gaithersburg, Maryland, 2003.

[351] Ellen M. Voorhees and Dawn M. Tice. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA, 2000. Association for Computing Machinery.

[352] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13697–13720, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[353] Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. Building an Argument Search Engine for the Web. In Kevin Ashley, Claire Cardie, Nancy Green, Iryna Gurevych, Ivan Habernal, Diane Litman, Georgios Petasis, Chris Reed, Noam Slonim, and Vern Walker, editors, *4th Workshop on Argument Mining (ArgMining 2017) at EMNLP*, pages 49–59. Association for Computational Linguistics, September 2017.

[354] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. Retrieval of the best counterargument without prior topic knowledge. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 241–251, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[355] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online, November 2020. Association for Computational Linguistics.

[356] Bingning Wang, Ting Yao, Weipeng Chen, Jingfang Xu, and Xiaochuan Wang. Multilingual question generation with language agnostic language model. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2262–2272. Association for Computational Linguistics, 2021.

[357] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360, Seattle, United States, July 2022. Association for Computational Linguistics.

[358] 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. *CoRR*, abs/2212.03533, 2022.

[359] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11897–11916. Association for Computational Linguistics, 2024.

[360] Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore, December 2023. Association for Computational Linguistics.

[361] Shiqi Wang, Yeqin Zhang, and Cam-Tu Nguyen. Mitigating the impact of false negatives in dense retrieval with contrastive confidence regularization. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[362] Xiaoyue Wang, Jianyou Wang, Weili Cao, Kaicheng Wang, Ramamohan Paturi, and Leon Bergen. BIRCO: A benchmark of information retrieval tasks with complex objectives. *CoRR*, abs/2402.14151, 2024.

[363] Yang Wang, Alberto Garcia Hernandez, Roman Kyslyi, and Nicholas Kersting. Evaluating quality of answers for retrieval-augmented generation: A strong LLM is all you need. *CoRR*, abs/2406.18064, 2024.

[364] Yue Wang, Hung Le, Akhilesh Gotmare, Nghi D. Q. Bui, Junnan Li, and Steven C. H. Hoi. CodeT5+: Open code large language models for code understanding and generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1069–1088. Association for Computational Linguistics, 2023.

[365] Zora Zhiruo Wang, Akari Asai, Xinyan Velocity Yu, Frank F. Xu, Yiqing Xie, Graham Neubig, and Daniel Fried. CodeRAG-bench: Can retrieval augment code generation? In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3199–3214, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[366] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria, July 2025. Association for Computational Linguistics.

[367] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on*

Neural Information Processing Systems, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[368] 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.

[369] John Wieting, Jonathan Clark, William Cohen, Graham Neubig, and Taylor Berg-Kirkpatrick. Beyond contrastive learning: A variational generative model for multilingual retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12044–12066, Toronto, Canada, July 2023. Association for Computational Linguistics.

[370] Konrad Wojtasik, Kacper Wołowiec, Vadim Shishkin, Arkadiusz Janz, and Maciej Piasecki. BEIR-PL: Zero shot information retrieval benchmark for the Polish language. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2149–2160, Torino, Italia, May 2024. ELRA and ICCL.

[371] Haolun Wu, Yansen Zhang, Chen Ma, Fuyuan Lyu, Bowei He, Bhaskar Mitra, and Xue Liu. Result diversification in search and recommendation: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 36(10):5354–5373, April 2024.

[372] Xing Wu, Guangyuan Ma, Meng Lin, Zijia Lin, Zhongyuan Wang, and Songlin Hu. Contextual masked auto-encoder for dense passage retrieval. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023*, pages 4738–4746. AAAI Press, 2023.

[373] Ziyi Xia, Kun Luo, Hongjin Qian, Siqi Bao, and Zheng Liu. Open data synthesis for deep research. In *The Fourteenth International Conference on Learning Representations*, 2026.

[374] Chenghao Xiao, G. Thomas Hudson, and Noura Al Moubayed. RAR-b: Reasoning as retrieval benchmark. *CoRR*, abs/2404.06347, 2024.

[375] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. RetroMAE: Pre-training retrieval-oriented language models via masked auto-encoder. In *Proceedings of the*

2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, pages 538–548. ACL, 2022.

[376] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA, 2024. Association for Computing Machinery.

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

[378] Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. Pride and prejudice: LLM amplifies self-bias in self-refinement. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15474–15492. Association for Computational Linguistics, 2024.

[379] 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 Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics, 2021.

[380] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.

[381] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024.

[382] Jheng-Hong Yang, Carlos Lassance, Rafael Sampaio De Rezende, Krishna Srinivasan, Miriam Redi, Stéphane Clinchant, and Jimmy Lin. AToMiC: An image/text retrieval test collection to support multimedia content creation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2975–2984, New York, NY, USA, 2023. Association for Computing Machinery.

[383] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1253–1256, New York, NY, USA, 2017. Association for Computing Machinery.

[384] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. CRAG - comprehensive RAG benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[385] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, oct-nov 2018. Association for Computational Linguistics.

[386] Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. Learning discriminative projections for text similarity measures. In Sharon Goldwater and Christopher D. Manning, editors, *Proceedings of the Fifteenth Conference on Computational Natural Language Learning, CoNLL 2011, Portland, Oregon, USA, June 23-24, 2011*, pages 247–256. ACL, 2011.

[387] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[388] Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel F Campos. Arctic-Embed 2.0: Multilingual retrieval without compromise. In *Second Conference on Language Modeling*, 2025.

[389] Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. Chain-of-Note: Enhancing robustness in retrieval-augmented language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14672–14685, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[390] Xiaodong Yu, Hao Cheng, Xiaodong Liu, Dan Roth, and Jianfeng Gao. ReEval: Automatic hallucination evaluation for retrieval-augmented large language models via transferable adversarial attacks. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1333–1351, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[391] Matei Zaharia, Omar Khattab, Lingjiao Chen, Jared Quincy Davis, Heather Miller, Chris Potts, James Zou, Michael Carbin, Jonathan Frankle, Naveen Rao, and Ali Ghodsi. The shift from models to compound AI systems, 2024.

[392] Hamed Zamani, Michael Bendersky, Donald Metzler, Honglei Zhuang, and Xuanhui Wang. Stochastic retrieval-conditioned reranking. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '22, page 81–91, New York, NY, USA, 2022. Association for Computing Machinery.

[393] George Zerveas, Navid Rekabsaz, and Carsten Eickhoff. Enhancing the ranking context of dense retrieval through reciprocal nearest neighbors. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10779–10803, Singapore, December 2023. Association for Computational Linguistics.

[394] Chengxiang Zhai and John D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. *SIGIR Forum*, 51(2):268–276, 2017.

[395] Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. Jasper and Stella: distillation of SOTA embedding models. *CoRR*, abs/2412.19048, 2024.

[396] Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. RAFT: Adapting language model to domain specific RAG. In *First Conference on Language Modeling*, 2024.

[397] 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.

[398] 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, 09 2023.

[399] Yanzhao Zhang, Mingxin Li an 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. *CoRR*, abs/2506.05176, 2025.

[400] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the AI ocean: A survey on hallucination in large language models. *Computational Linguistics*, 51(4):1373–1418, 12 2025.

[401] Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. SPARTA: Efficient open-domain question answering via sparse transformer matching retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 565–575. ACL, 2021.

[402] Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Zetian Sun, Zhenyu Liu, Dongfang Li, Xinyuan Wei, Qian Chen, Youcheng Pan, Yang Xiang, Meishan Zhang, Haofen Wang, Jun Yu, Baotian Hu, and Min Zhang. KaLM-Embedding-V2: Superior training techniques and data inspire A versatile embedding model. *CoRR*, abs/2506.20923, 2025.

[403] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez,

and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[404] Shuyan Zhou, Uri Alon, Frank F. Xu, Zhengbao Jiang, and Graham Neubig. DocPrompting: Generating code by retrieving the docs. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[405] Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. LongEmbed: Extending embedding models for long context retrieval. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 802–816, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[406] Shengyao Zhuang, Linjun Shou, and Guido Zuccon. Augmenting passage representations with query generation for enhanced cross-lingual dense retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 1827–1832, New York, NY, USA, 2023. Association for Computing Machinery.

[407] Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 307–314, New York, NY, USA, 1998. Association for Computing Machinery.