Transformers for Search: Retrieval, Robustness, and Refusal

Thilina Chathuranga Rajapakse
Rajapakse Mudiyanselage

Transformers for Search: Retrieval, Robustness, and Refusal

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Agnietenkapel
op vrijdag 13 februari 2026, te 16:00 uur

door

Thilina Chathuranga Rajapakse Rajapakse Mudiyanselage

geboren te Peradeniya

Promotiecommissie

Promotor:	prof. dr. M. de Rijke	Universiteit van Amsterdam
Co-promotor:	dr. A.C. Yates	Johns Hopkins University
Overige leden:	dr. M. Alian Nejadi	Universiteit van Amsterdam
	prof. dr. C.L.A. Clarke	University of Waterloo
	dr. C. Hauff	Spotify
	prof. dr. E. Kanoulas	Universiteit van Amsterdam
	prof. dr. C. Monz	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research was carried out at the Information Retrieval Lab at the University of Amsterdam, with support from DreamsLab.

Printed by Proefschriftspecialist, Zaandam

ISBN: 978-94-93483-80-4

Acknowledgements

Growing up in a family of academics, pursuing a PhD felt like a natural next step. Not out of expectation or pressure, but out of familiarity. I knew the rhythms, the language, and the peculiar norms of academic life well enough, yet what I did not anticipate was how imperfectly I would fit the mould. My path through the PhD was rarely linear and often chaotic, shaped as much by distraction and disorder (qualities that are rarely listed among core academic competencies) as by curiosity and enthusiasm. Learning to function as a researcher without sanding myself down into something more conventional was not always easy. What made it possible was the patience, generosity, and trust of the people around me, who supported me not by correcting me into sameness, but by helping me find a way to do good research as myself. This thesis is as much a product of that support as it is of any individual effort.

Maarten, thank you for your supervision and patience throughout my PhD. I was not the easiest PhD student to supervise. I can be forgetful, outspoken, and disorganised, and my attention (or lack thereof) can wander. At various points, I focused far too much on details that ultimately did not matter, or struggled to focus at all. You had a remarkable ability to see the bigger picture and to remind me of it when I lost sight of it myself. When I was disheartened, you encouraged me. When I was scattered, you helped channel the chaos rather than suppress it. Your trust, calmness, and vision made it possible for me to grow as a researcher without having to become someone else in the process.

Andrew, thank you for always speaking your mind, in research and in life. Our relationship started in a slightly unconventional way, first as drinking buddies and only later as supervisor and student, which somehow ended up working remarkably well. You kept me on my toes and never let me get away with sloppy thinking, whether the topic was a half-baked research idea or a wildly impractical plan. I appreciated your honesty, your direct feedback, and your willingness to call things out when needed. Outside of research, you remain the only person I have met who could reliably go toe-to-toe with me on spicy food, a fact I respect (grudgingly). Your sharp insight, humour, and intellectual integrity made our discussions both challenging and enjoyable, and played an important role in shaping how I think about research.

I would also like to thank my internship managers, who made my time outside the lab both rewarding and formative. Anne and Claudia, thank you for your support, guidance, and trust during my internship at Spotify. I am especially grateful for the thoughtful feedback and for making me feel genuinely part of the team. Yang, thank you for the opportunity to work at DeepMind, and for your guidance and perspective during my time there. I learned a great deal from the experience, and your enthusiasm and curiosity made my time at DeepMind a genuine pleasure.

I am grateful as well to Mohammad Aliannejadi, Charlie Clarke, Claudia Hauff, Evangelos Kanoulas, and Christof Monz for agreeing to be on my defense committee.

Sami, you were the first person I got to know in Amsterdam, and you immediately made me feel at home. Over time, you became a friend, a confidant, and very much a chosen brother. From dinners, drinks, and board games to long conversations about philosophy, politics, and everything in between, you were a constant presence throughout my PhD. Thank you for your honesty, your loyalty, and for always being there, whether

I needed advice, distraction, or simply good company.

Jasmin, thank you for the countless coffees, and apologies for any role I may have played in your increased caffeine intake. You always listened with patience and understanding, and you somehow managed to say yes to most of my plans, even the poorly thought-out ones. I am grateful for your warmth, your support, and for the many conversations that made difficult days feel lighter. Thank you for showing up so consistently, every time I needed it.

And thank you both for agreeing to be my paranymphs, fully aware that this time would be more demanding than usual.

There are many others who made this PhD lighter, easier, and more enjoyable in ways that are hard to categorise, but impossible to forget. Ruben, you are the most helpful person I know, always offering help with anything I asked for, and often with things I had not yet realised I needed. Sameera, thank you for visiting me in Amsterdam so often, and for letting me stay with you in Liège when I no longer had an apartment in Amsterdam, turning what could have been a very stressful period into something I now look back on fondly. Mohammad, thank you for the pizzas, and for getting me into baking; both have had a lasting impact. Sam, thank you for being Sam the wise, and for your patience and calm perspective. Clara, thank you for always greeting me with a big smile and an even bigger hug, and for your ability to light up a room. Syrenna, thank you for the physics discussions and for always keeping me honest about them. Thong, thank you for the encouragement, the research discussions, your kindness, and the generosity with which you share all of that. Clem, thank you for your zero-bullshit policy and for consistently providing clarity. Gabriel, thank you for always making things feel a little lighter. Philipp, thank you for your humour and for being a supportive friend. Flo, thank you for your words of wisdom and your sense of perspective. Helen, thank you for always being there with a sympathetic ear, and for your patience in dealing with all the nerds. Tatvan, we became friends in the most random of ways, and now I would be lost without your support. Roxana, thank you for the countless hours we spent talking about everything; it was always a pleasure. Shashank, thank you for making some meetings more tolerable. Shao, thank you for all the hotpots, and for the oysters that you do not remember. Ming, thank you for all the dinners in London. Tom, thank you for all the pecan cakes.

A special place in this journey belongs to the Wednesday dinners, a weekly ritual that provided structure, good food, and even better company throughout much of my PhD. What started as a simple idea quickly became something I looked forward to every week, even when it meant cooking chaos and, on more than one occasion, fitting close to twenty people into a very small apartment. Those dinners were a constant during periods that were otherwise anything but stable. Antonis, Dan, Ivona, Jin, Jingfen, Nathalie, Romain, Vera, and Weijia, thank you for showing up so regularly, for the conversations, and for helping turn my home into a place of warmth and laughter. Many others, including friends mentioned earlier, joined often as well, and helped make those evenings what they were. I am also grateful to Cosimo and Alessio, who, despite being in Amsterdam only briefly, quickly became regulars and made the dinners feel complete during their time here.

I am also grateful to the wider IRLab community, past and present, for the many coffee breaks, Friday drinks, lunches, and moments of shared frustration and laughter. I

would like to thank Ali, Ana, Arezoo, Chuan, David, Dylan, Georgios, Gabrielle, Ivana, Jingwei, Julien, Kidist, Kiki, Maarten M, Maartje, Maria, Mariya, Maurits, Maxime, Ming, Mohanna, Mozhdeh, Panagiotis, Petra, Pooya, Ruqing, Siddharth, Sid, Simon, Svitlana, Teng, Vaishali, Xinyi, Yibin, Yixing, Yongkang, Yougang, Yuanna, Yubao, Yuyue, and Zahra for helping make IRLab such a supportive, engaging, and genuinely enjoyable place to work.

Even from afar, my friends from Sri Lanka remained an important constant throughout my PhD. Jani, thank you for being my best friend and for always being just a call away, even from the other side of the world. Maithri, your unwavering support and our regular conversations consistently lifted my mood, no matter how difficult things felt at the time. To Banda, Batta, Gayya, and Sajana, thank you for the friendship, the shared history, and for reminding me that time and distance do not weaken the bonds that matter. Knowing that you were there, even when we spoke less often, meant more than you probably realise.

Finally, I want to thank my family. Amma, thank you for your unwavering love and support, and apologies for contributing more than my fair share of grey hairs over the years, especially compared to my brothers. Thaththa, you were the first scientist in my life, and you sparked my curiosity about the world and how things work long before I ever thought about research as a career. Lokkayya, thank you for the shared love of fantasy and science fiction, and for all the conversations that came with it. Chutiayya, thank you for always believing in me, even when I doubted myself. I would not be here without all of you.

Thilina Rajapakse
Eindhoven

Contents

1 Introduction

Access to information has never been easier, quicker, nor more fragile. The same technologies that can instantly locate a single drop of information in the vast ocean of human knowledge can also invent an answer, where none exists, just as quickly [43, 46, 62]. As language models have grown to dominate both search and question answering, the line between retrieving and generating information has quietly blurred [45, 57, 59, 93, 110, 130, 147]. What began as a way to find relevant information now extends to producing entire explanations, stitched together from memory and retrieved evidence [16, 45, 57]. This new generation of retrieval-augmented models makes it quick and easy to answer even complex questions [45, 85, 130], yet that convenience often hides a fragility [67, 87–89, 111]. They work best when the world looks like their training data, and often falter when it does not.

Modern information retrieval systems increasingly follow a pipeline architecture, where a first-stage retriever finds a small candidate set of relevant documents and a reader or generator model produces an answer conditioned on the retrieved candidate set [31, 50, 51]. The now ubiquitous transformer models dominate both stages of this pipeline, with transformer encoders supporting semantic matching and scalable indexing for the first stage, and decoder-based language models providing language understanding and answer generation for the second stage [26, 44, 48, 50, 51, 86, 113, 147]. This shift has practical consequences. What is retrieved and what the generator is tasked to do with it are tightly coupled [42, 126]. Two primary requirements follow if such systems are to perform reliably at scale. First,

generalizability: a retrieval model trained on one distribution of data must continue to work on new distributions of data such as new datasets, domains, and languages [20, 111, 144].

And second,

grounded answers: a generator model should base answers on the retrieved evidence, and crucially, abstain from answering when evidence is missing [19, 79, 121, 133].

When retrieval and generation lose alignment, the boundary between fact and fabrication starts to thin [67]. A retriever that overlooks key evidence leaves the generator to fill in the gaps; a generator that overreaches beyond the retrieved evidence can produce answers that seem plausible but are inaccurate. In low-stakes settings, this may pass unnoticed, but in higher stakes settings (scientific, legal, medical, etc.) such errors can

have severe consequences [6, 88]. The challenge, then, is to build systems that remain reliable when the data shifts, when the context is noisy, and when the correct response is to say nothing at all.

This thesis studies those requirements together and focuses on practical interventions that can be adopted. On the retrieval side, it studies how training data augmentation and negative sampling strategies shape the behavior of dense retrievers under distribution shift, and proposes methods that make them more stable across domains and languages [87, 89]. On the generation side, it investigates how small, open-source language models can be trained to answer queries based solely on retrieved evidence, and refuse to answer when that evidence is insufficient [88, 133, 141]. Finally, it also emphasizes the accessibility of language technology and introduces an accompanying open-source library, Simple Transformers, that lowers the barrier to building, evaluating, and reproducing transformer-based retrieval and question answering systems, enabling these methods to be shared, tested, and extended by a wider community [28, 61, 90, 128].

1.1 Research Outline and Questions

Building on the motivations above, this section outlines the research directions and questions that structure the thesis. Part I focuses on dense retrieval and the challenge of maintaining performance under distribution shift. Part II turns to generation, studying how small language models can be trained to ground answers in retrieved evidence and refuse when evidence is insufficient. Part III complements these empirical studies with a practical contribution: an open-source library that lowers the barrier to building, evaluating, and reproducing transformer-based retrieval and question answering systems.

1.1.1 Robust retrieval under distribution shifts

In terms of effectinveness, dense retrievers based on the transformer architecture far surpass traditional lexical retrieval techniques [50, 51, 131] when tested under in-distribution settings, but its performance can degrade under domain, style, or language shifts [111, 138, 144, 145]. Part I of the thesis focuses on two factors that influence this behavior and determine how well dense retrievers generalize: (i) the choice of negative examples during training, and (ii) the composition of the training data itself. Together, these studies investigate how training-time design choices shape the robustness of dense retrievers across domains and languages.

RQ 1 How do negative sampling strategies affect the generalization of dense retrievers under distribution shift across domains and languages?

Chapter 2 investigates multiple negative sampling strategies in a multilingual setting. The study compares lexical, iterative, and clustering-based methods across in-distribution, out-of-distribution, and zero-shot conditions, and introduces *iterative clustered training* (ICT), a method that periodically refreshes hard negatives from semantically similar clusters. ICT achieves the most robust performance when test distributions differ from training data and when applied to unseen languages.

RQ 2 Does training a dense passage retriever (DPR) model on data containing multiple queries per passage improve the generalizability of the model?

Chapter 3 explores how training data composition influences dense retriever generalization. Typical retrieval datasets pair each passage with a single query, which can narrow what the model learns to represent. By generating datasets with multiple queries per passage, this study shows that retrieval models learn richer representations and achieve more stable performance under distribution and domain shifts.

1.1.2 Robust refusal and evidence-based answering

Better retrieval methods alone do not prevent ungrounded or overconfident answers. In retrieval-augmented generation (RAG) systems, generators can produce plausible but unsupported responses, or attempt to answer questions even when no relevant evidence is available [43, 67, 79, 92]. This part of the thesis focuses on small (below 10B-parameter) instruction-tuned models, which are attractive for deployment due to their efficiency and accessibility [113, 141]. A controlled evaluation across multiple multi-hop question answering datasets reveals three recurring weaknesses at these scales: (i) limited accuracy even with gold evidence, (ii) sharp accuracy degradation in the presence of distractor documents, and (iii) unreliable refusal, with models frequently answering despite explicit prompts to abstain. These observations motivate the need for training objectives that explicitly teach models when to answer and when to refuse.

RQ 3 How can relevance-based rewards be used in reward shaping to train small language models to answer based on retrieved evidence and to refuse when evidence is insufficient?

Chapter 4 introduces *reward shaping for robust refusal* (RSRR), a reinforcement learning framework that augments the standard proximal policy optimization (PPO) fine-tuning setup with relevance-based rewards [99, 148]. These rewards explicitly encourage models to answer based on retrieved evidence and to refuse when evidence is insufficient. Across datasets containing distractor and unanswerable questions, RSRR yields substantial improvements in correct refusals and robustness to irrelevant context while maintaining accuracy when evidence is present.

Part II of the thesis investigates how relevance-based reward shaping can be used to train small language models for robust refusal and reliable, evidence-based answering in retrieval-augmented settings.

1.1.3 Accessibility and shared practice

Methods only matter if they can be used. Reproducibility and accessibility are essential for sustained progress in transformer-based information retrieval and question answering. Training and evaluating transformer models often requires substantial engineering effort, inconsistent interfaces, and ad-hoc evaluation pipelines, which limit the broader adoption and reuse of research outputs [18, 100]. This part of the thesis addresses these challenges by introducing a unified, open-source framework, called Simple Transformers, that standardizes transformer training, evaluation, and analysis across tasks.

RQ 4 Can an open-source framework like Simple Transformers lower the technical barriers to training and reproducing transformer-based retrieval and QA models?

Chapter 5 presents *Simple Transformers*, an open-source library that provides a consistent interface for training and evaluating transformer models across tasks such as classification, retrieval, and question answering. The framework encapsulates standard components, such as model configuration, data preprocessing, evaluation metrics, and logging, into accessible abstractions that reduce boilerplate code and lower the engineering overhead required to experiment with transformer architectures. By unifying these components, the library promotes reproducibility, accelerates experimentation, and enables researchers and practitioners to adopt and extend transformer-based methods with minimal setup.

Part III of the thesis contributes practical infrastructure that underpins the experiments described in Parts I and II, supporting reproducible research and lowering the barriers to entry for transformer-based information retrieval and generation.

Together, the three parts of this thesis address complementary aspects of reliability and accessibility in retrieval-augmented systems. Part I examines how dense retrievers can be trained to remain robust under distribution shifts through better negative sampling and data composition. Part II introduces relevance-based reward shaping to improve the reliability of small language models, enabling robust refusal and evidence-based answering. Part III complements these contributions with an open-source framework that standardizes training and evaluation practices, ensuring that the methods developed in this work can be reproduced, extended, and applied more broadly.

1.2 Main Contributions

This thesis makes contributions across training methods for dense retrieval, training objectives for small language models in retrieval-augmented generation, and open-source tooling for reproducible information retrieval.

Methods and algorithms

Iterative clustered training (ICT) for negative sampling (Chapter 2). This method periodically clusters training passages or queries and refreshes hard negatives from semantically similar clusters. ICT achieves stronger robustness under domain, style, and language shifts than lexical or static iterative methods, while avoiding the heavy indexing cost of full dense negative mining. Empirical results across multilingual retrieval benchmarks show that BM25 negatives yield the best in-distribution performance, whereas ICT provides the strongest generalization under out-of-distribution and zero-shot settings.

Training data composition for generalizable dense retrievers (Chapter 3). A synthetic data-generation pipeline creates multiple queries per passage, encouraging models to encode a more complete view of each passage's semantics. Dense retrievers trained on these multi-query datasets generalize far better to unseen domains and datasets,

outperforming models trained on single-query corpora even when trained with fewer total examples.

Reward shaping for robust refusal (RSRR) (Chapter 4). A reinforcement-learning framework based on PPO that uses relevance-based rewards to train small language models for retrieval-augmented question answering. The approach explicitly rewards models for answering based on retrieved evidence and for refusing when evidence is insufficient, substantially improving calibrated refusals and robustness to distractor documents while maintaining answer accuracy when evidence is present.

Artifacts and software

Simple Transformers library (Chapter 5). An open-source framework that standardizes transformer training and evaluation across tasks such as classification, retrieval, and question answering. The library lowers the engineering overhead for research on dense retrievers and retrieval-augmented generation, provides consistent metrics and per-query analysis, and underpins the experimental work presented in this thesis.

Released datasets and evaluation settings. This includes *generated multi-query datasets* for training generalizable dense retrievers and *augmented QA datasets* with withheld and distractor documents for training and evaluating robust refusal.

Empirical findings and guidance

Negative sampling guidance. BM25 negatives perform best for in-distribution training, while ICT—especially passage-level clustering—provides the most reliable generalization across domains and languages.

Data composition guidance. Training on multiple queries per passage yields better passage representations and more stable out-of-distribution performance than single-query training.

Small-LM RAG guidance. Instruction-tuned models below 10B parameters are prone to answering without sufficient evidence; relevance-based reward shaping significantly improves refusal calibration and robustness to noisy retrieval.

1.3 Thesis Overview

The thesis is organized into three parts. Part I focuses on dense retrieval and examines how training-time choices such as negative sampling and data composition influence the generalization of transformer-based retrievers under distribution shifts. Part II investigates retrieval-augmented generation and introduces a reward shaping framework for robust refusal in small language models. Part III complements these empirical studies with an open-source contribution, the *Simple Transformers* library, which enables accessible and reproducible transformer training and evaluation. Each part can be read independently.

Part I – Robust Retrieval under Distribution Shifts. This part consists of two chapters. Chapter 2 studies the effect of negative sampling strategies on the generalization of dense retrievers across domains and languages. It introduces *iterative clustered training* (ICT), a method that periodically clusters passages or queries and refreshes hard negatives from semantically similar clusters, achieving strong robustness under distribution shift while remaining computationally efficient. Chapter 3 investigates the role of training data composition, demonstrating that training on datasets with multiple queries per passage produces more generalizable retrievers that maintain performance on unseen datasets and domains. Together, these chapters provide practical guidance on training dense retrieval models that remain stable under domain, style, and language variation.

Part II – Robust Refusal and Evidence-Based Answering. Chapter 4 introduces *reward shaping for robust refusal* (RSRR), a reinforcement learning framework that teaches small instruction-tuned language models to answer based on retrieved evidence and to refuse when evidence is insufficient. The framework augments standard PPO fine-tuning with relevance-based rewards that balance correctness, refusal, and formatting objectives. Empirical results across multiple multi-hop QA datasets show consistent improvements in calibrated refusal and robustness to distractor documents without loss of accuracy when sufficient evidence is provided.

Part III – Accessibility and Shared Practice. Chapter 5 presents the *Simple Transformers* library, an open-source framework designed to make transformer-based retrieval and question answering more accessible and reproducible. The library provides unified interfaces for training and evaluation across tasks, integrated metrics and per-query analysis, and implementations of the retrieval and RAG methods explored in this thesis. By lowering the engineering barriers to experimentation, it enables the broader research community to replicate, extend, and apply the methods developed in Parts I and II.

1.4 Origins

Chapter 2 is based on the following paper:

- T. C. Rajapakse, A. Yates, and M. de Rijke. Negative sampling techniques for dense passage retrieval in a multilingual setting. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 575–584, 2024.

  TCR: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Writing – Original Draft Preparation, Writing – Review & Editing. AY and MdR: Conceptualization, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing.

Chapter 3 is based on the following paper:

- T. C. Rajapakse and M. de Rijke. Improving the generalizability of the dense passage retriever using generated datasets. In *European Conference on Information Retrieval*, pages 94–109. Springer, 2023.

  TCR: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Writing – Original Draft Preparation, Writing – Review & Editing. MdR: Conceptualization, Supervision, Writing – Review & Editing.

Chapter 4 is based on the following paper:

- T. C. Rajapakse and M. de Rijke. Reward shaping for robust refusal in small language models for retrieval-augmented question answering. In *Under Review*, 2026.

  TCR: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Writing – Original Draft Preparation, Writing – Review & Editing. MdR: Conceptualization, Supervision, Writing – Review & Editing.

Chapter 5 is based on the following paper:

- T. C. Rajapakse, A. Yates, and M. de Rijke. Simple Transformers: Open-source for all. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 209–215, 2024.

  TCR: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Writing – Original Draft Preparation, Writing – Review & Editing. AY and MdR: Conceptualization, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing.

The writing of the thesis also benefited from work on the following publications:

- N. Vermeer, V. Provatorova, D. Graus, T. Rajapakse, and S. Mesbah. Using Robbert and extreme multi-label classification to extract implicit and explicit skills from Dutch job descriptions. *Compjobs' 22: Computational Jobs Marketplace*, 1 (1):2–6, 2022.

- W. Zhang, S. Vakulenko, T. Rajapakse, Y. Xu, and E. Kanoulas. Tackling query-focused summarization as a knowledge-intensive task: A pilot study. *arXiv preprint arXiv:2112.07536*, 2023.

- W. Zhang, J.-H. Huang, S. Vakulenko, Y. Xu, T. Rajapakse, and E. Kanoulas. Beyond relevant documents: A knowledge-intensive approach for query-focused summarization using large language models. In A. Antonacopoulos, S. Chaudhuri, R. Chellappa, C.-L. Liu, S. Bhattacharya, and U. Pal, editors, *Pattern Recognition*, pages 89–104, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-78495-8. doi: 10.1007/978-3-031-78495-8_6.

- M. de Rijke, B. van den Hurk, F. Salim, A. A. Khourdajie, N. Bai, R. Calzone, D. Curran, G. Demil, L. Frew, N. Gießing, M. K. Gupta, M. Heuss, S. Hobeichi, D. Huard, J. Kang, A. Lucic, T. Mallick, S. Nath, A. Okem, B. Pernici, T. Rajapakse, H. Saleem, H. Scells, N. Schneider, D. Spina, Y. Tian, E. Totin, A. Trotman, R. Valavandan, D. Workneh, and Y. Xie. Report on the 1st workshop on information retrieval for climate impact (MANILA24) at SIGIR 2024. *SIGIR Forum*, 59(1):1–23, Oct. 2025. ISSN 0163-5840. doi: 10.1145/3769733.3769737.

Part I

ROBUST RETRIEVAL UNDER DISTRIBUTION SHIFTS

2

Negative Sampling Techniques for Dense Passage Retrieval in a Multilingual Setting

2.1 Introduction

Dense retrieval architectures consisting of two transformer models (bi-encoders) have become the state-of-the-art architecture for passage retrieval [39, 50, 51, 131]. The dense passage retriever (DPR) model consists of two transformer models that encode the queries and passages separately. Bi-encoder architectures for dense retrieval are typically employed to pre-compute passage representations at indexing time, on top of which computationally more costly re-rankers such as the cross-encoder architecture [26] can be run.

Hard negative mining or negative sampling techniques have been used in prior work to improve the effectiveness of bi-encoder models [39, 123, 131]. However, recent work has demonstrated that reported results can vary significantly based on multiple factors that can be easily overlooked. For example, Lassance and Clinchant [55] show that some previous work uses the titles from the MS MARCO dataset leading to unfair comparisons with methods that do not use the titles. In light of this, we implement all the methods compared in this work from scratch, using the same libraries and library versions, and evaluate the methods using the same evaluation framework to provide a fair comparison. All code is publicly available on GitHub.

Effectiveness of negative sampling strategies on MS MARCO (English). The choice of negative sampling strategy has a significant effect on the effectiveness of the final retrieval model. Based on the work of Xiong et al. [131], Hofstätter et al. [39], and Wang and Zuccon [123], we find that clustering-based negative sampling methods and iterative negative sampling methods offer comparable performance while outperforming lexical negative sampling methods (details on the different methods can be found in Section 2.3.3). Our goal in this chapter is to extend the analysis of negative sampling

This chapter was published as T. C. Rajapakse, A. Yates, and M. de Rijke. Negative sampling techniques for dense passage retrieval in a multilingual setting. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 575–584, 2024.

methods to multilingual retrieval and determine which negative sampling strategy is best-suited for this understudied setting.

Monolingual retrieval beyond English. Information access in languages other than English is a topic with a long history in information retrieval, with resources, benchmarking activities and algorithm development going back decades; see [e.g., 84] for an early survey. In contrast, research on DPR has mainly been focused on English [50, 51, 131], even though some work has been done on monolingual DPR for other languages, such as Arabic, Japanese, and Russian [146]. These models have been trained on monolingual corpora and have achieved high performance on monolingual retrieval tasks.

Multilingual DPR for monolingual retrieval. Using a *multilingual* DPR model for *monolingual* purposes has some clear advantages. It allows for using cross-lingual information transfer, which can improve performance on low-resource languages [146]. Furthermore, a multilingual model can perform zero-shot retrieval on languages for which it has not been explicitly trained. For example, Zhang et al. [146] show that a multilingual dense retrieval model can be used on a new language in a zero-shot manner with some success, enabling retrieval even in languages for which no retrieval training data is available. Other work [e.g., 5, 15, 60, 70, 101, 144] supports this finding. Using a single *multilingual* model for *monolingual* retrieval for many different languages is more cost-effective and scalable than training a separate *monolingual* model for each language of interest. Thus, the *zero-shot* setting is of particular interest in this work. We explore the capabilities of *multilingual* bi-encoders in a *monolingual* setting. We train our models to perform retrieval for multiple languages (one model for many languages, i.e., *multilingual*), and we test them on *monolingual* datasets (queries and passages in the same language, i.e., *monolingual*).

Generalizability. We address the generalizability of dense retrieval models to new data and new languages. This is of particular interest because dense retrieval models are known to struggle with out-of-distribution data [111, 144], often falling behind traditional sparse methods when tested in zero-shot settings. Given this drawback, we consider the retrieval performance across three settings: (i) *in-distribution*, (ii) *out-of-distribution*, and (iii) *zero-shot*. The in-distribution setting gives us an idea of how well a model learns the distribution of data similar to the training data. Out-of-distribution testing demonstrates how we may expect a model to perform when exposed to new types of queries and passages. The zero-shot performance of the models is of particular interest as this represents the real-world use-case of using a multilingual retrieval model on a language that it is not trained on as no training data was available for that language.

Negative sampling. Negative sampling is the process of selecting negative examples (passages that are not relevant to a given query) for training a dense retrieval model. Negative examples are used to train the model to differentiate between relevant and non-relevant passages. Negative sampling that includes *hard* negatives (passages that are similar to the query but are not relevant) is crucial for the effectiveness of dense retrieval models [39, 131]. Given the importance of negative sampling, we study the effectiveness of negative sampling methods in a multilingual setting.

Simple methods to select negative examples for training dense retrieval models include random selection from the corpus (DPR_{base}) or from BM25's top-ranked docu-

ments (DPR_{BM}). However, these approaches do not ensure that the negative examples are hard negatives, which has motivated other work.

Hofstätter et al. [39] cluster queries and select queries from the same cluster for a given batch to increase the probability of in-batch negatives being hard negatives (TAS-Q). Similarly, passages can be clustered, and training batches can be built from the same cluster of passages (TAS-P). Xiong et al. [131] iteratively update a dense index of the full collection by periodically re-computing representations of all passages and select passages that are ranked highly (but not at the top) for each query as negative examples (ANCE).

As part of our reproducibility work, we identify a gap left by these methods and consider a combination of these two approaches that combines clustered training with iterative updates produced using a subset of the collection, which we refer to as *iterative clustered training* (ICT). Unlike the work in [39], this method uses the representations from the model being trained to perform clustering instead of a separate teacher model. The passages are clustered at the start of every training epoch to ensure that the training objective remains challenging even as the model learns to differentiate between similar passages better (ICT-P). Similarly, this method can also be applied to query representations (ICT-Q). This method is complementary to existing methods and combines insights from the methods proposed by Hofstätter et al. [39] and Xiong et al. [131]. Sections 2.3.2 and 2.3.3 provide detailed descriptions of these negative sampling methods.

Based on these existing negative sampling techniques, as well as the ICT technique introduced in this chapter, we pose the following research questions in order to answer the broader research question (**RQ 1**): How do negative sampling strategies affect the generalization of dense retrievers under distribution shift across domains and languages?

RQ 1.1 Do prior findings from negative sampling studies in English language dense retrieval remain valid for multilingual retrieval?

RQ 1.2 Which negative sampling method offers the best overall performance in multilingual dense retrieval?

Main findings. We find that the use of negative sampling methods yields significant improvements in a multilingual retrieval setting, reproducing the lessons from prior work in English. The ICT methods perform the best overall, showing the best results in both out-of-distribution and zero-shot conditions, while achieving the second-highest scores under the in-distribution condition. ICT-P performs best out of the two ICT methods. DPR_{BM} shows the best results under the in-distribution conditions.

Furthermore, we see that TAS style clustering is less effective in a multilingual setting than the other methods. This contradicts the lessons learned from English only retrieval, where TAS is competitive with other negative sampling methods (such as ANCE). Thus, we find that ANCE generalizes better to our new multilingual setting than TAS.

Our results demonstrate that the clustered training method we propose leads to the best overall retrieval quality in a multilingual retrieval setting. Finally, we provide recommendations on which negative sampling method should be used in different scenarios.

2.2 Task Definition

Our task is to use *multilingual* (the same model used with multiple languages) DPR models to perform *monolingual* (queries and passages in the same language) dense retrieval. We study the effectiveness of existing negative sampling techniques as well as our proposed technique, clustered training, for this task. Extending beyond the findings of prior work on English language (monolingual) retrieval with English models, we explore *monolingual* retrieval with *multilingual* models and test whether these findings can be reproduced in this new setting. We investigate the effectiveness of each negative sampling technique under three conditions: (i) in-distribution, (ii) out-of-distribution, and (iii) zero-shot.

- The *in-distribution condition* uses test data from the same datasets used to train the models. The in-distribution test datasets consists of languages the models have been trained on for retrieval. As the training and test data were all gathered using the same methods at the same time, we consider this to be the in-distribution setting.
- The *out-of-distribution condition* uses test data from datasets that are different from the models' training datasets. The test sets employed under this condition solely consist of languages the models have been trained on for retrieval. As these test datasets were built using different methods, at different times, and by different contributors compared to the training data for the models, we call this the out-of-distribution setting. This setting is out-of-distribution with respect to the testing datasets.
- Similar to the out-of-distribution setting the *zero-shot testing condition* uses test datasets that were built using different methods, at different times, and by different contributors compared to the training data for the models. However, the test sets under this condition consist solely of languages the models have not been trained on for retrieval. This setting is out-of-distribution with respect to both the test datasets *and* the languages being tested. Hence, this is our zero-shot test setting.

2.3 Negative Sampling for Dense Retrieval

We recall DPR and negative sampling techniques that have been considered for DPR. We discover a natural but "missing" approach for negative sampling, which we then describe in detail.

2.3.1 Dense passage retriever (DPR)

Our work uses the DPR model [50]; one of the first effective dense retrieval models. The DPR model consists of two BERT encoders, a passage encoder $E_p(\cdot)$ and a query encoder $E_q(\cdot)$, used to encode passages and queries separately. The passage encoder $E_p(\cdot)$ is used to encode all passages into d-dimensional vectors, and a dense retrieval index is built with FAISS [48] for all M passages [50].

During retrieval, the query encoder $E_q(\cdot)$ is used to encode a query to a d-dimensional vector and a desired number of passages are retrieved from the index where the passage vectors are most similar to the query vector. The similarity is simply defined as the dot product of two vectors [50]:

$$\text{sim}(q, p) = E_Q(q)^\intercal E_p(p). \tag{2.1}$$

The training goal is to learn encoders $E_p(\cdot)$ and $E_q(\cdot)$ such that the encoded representations for relevant queries and passage pairs have higher similarity relative to irrelevant query and passage pairs. Consider $D = \{\langle q_i, p_i^+, p_{i,1}^-, \ldots, p_{i,n}^-\rangle\}_{i=1}^m$, where D is a training batch consisting of m instances. Each such instance contains a question q_i and a relevant passage p_i^+, as well as n irrelevant passages $p_{i,j}^-$ [50].

In our work, we use the in-batch negative [50] strategy when training the models. Therefore, the irrelevant passages are the relevant passages for the other queries in the batch.

The loss function is optimized as the negative log likelihood of the relevant passage:

$$L(q_i, p_i^+, p_{i,1}^-, \ldots, p_{i,n}^-) = -\log \frac{\mathrm{e}^{\text{sim}(q_i, p_i^+)}}{\mathrm{e}^{\text{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} \mathrm{e}^{\text{sim}(q_i, p_{(i,j)}^-)}}. \tag{2.2}$$

The original DPR model was initialized from BERT [26] (English). However, the models in this work are initialized with Multilingual BERT (mBERT), following [146], to facilitate multilingual retrieval.

2.3.2 Key characteristics of negative sampling methods

We observe three key dimensions of negative sampling techniques for dense retrieval:

Iterative or non-iterative: whether the negative samples are updated periodically during training;

Negative mining model: the model used to find the hard negatives; and

Hard negative source: what is the source of the hard negatives, i.e., whether the hard negatives are sampled from the corpus or from the training passages or queries (the size of the corpus is much bigger than the number of training queries/passages).

Table 2.1 summarizes the negative sampling methods we have listed so far and the design decisions made for the three dimensions listed above. The top part of the table shows the decisions of the existing negative sampling techniques. We observe that there is a gap in the existing methods: they do not consider the use of clustering self-generated (the model being trained) representations periodically to generate hard negatives. The bottom part of the table characterizes these gaps, which we describe in detail in Section 2.3.3.

Table 2.1: Overview of negative sampling methods used for dense retrieval DPR and their features. Top: previously published. Bottom: newly proposed.

Model	Source	Negative mining model	Hard negative source	Iterative updates
Base	[50]	N/A	N/A	N/A
BM25	[50]	BM25	Full corpus	No
TAS-Q	[39]	Teacher model	Training queries	No
TAS-P	[39]	Teacher model	Training passages	No
ANCE	[131]	Self	Full corpus	Yes
ICT-Q	This work	Self	Training queries	Yes
ICT-P	This work	Self	Training passages	Yes

2.3.3 Current negative sampling techniques

Random negatives (DPR_{base}). A dense retrieval model can easily be trained with random negatives in an inbatch-negative contrastive loss training scheme [50]. Here, a single training sample s consists of a query q, out of the full set of queries Q, and its relevant passage p_q. Then, a single training batch B of batch size b out of the full set of n training batches D is built as follows:

$$D = \sum_{i=1}^{n} B_i \tag{2.3}$$

$$B = \{(q, p_q) \mid q \in random(Q, b)\}. \tag{2.4}$$

Here, B_i is the i-th batch of the full set D and $random(Q, b)$ are b queries randomly sampled from Q without replacement. Then the dense retriever can be trained as described in Section 2.3.1.

BM25 negatives (DPR_{BM}). Negatives can be sampled from the corpus using the BM25 algorithm [95] by sampling passages from the top k retrieved passages. The BM25 algorithm is a bag-of-words retrieval function that ranks a set of documents (or passages) based on the query terms appearing in each document. We refer the reader to [50, Section 3.2] for a detailed description of using BM25 to sample negatives for training dense retrievers.

Topic aware sampling (TAS-Q and TAS-P). Topic aware sampling (TAS) [39] is a technique that aims to improve the effectiveness of dense retrievers by building training batches where all in-batch negatives are hard negatives for any given query. TAS achieves this by clustering queries once at the beginning of training and then sampling from those clusters to build training batches (TAS-Q). TAS style negative sampling requires a teacher model (any model trained to generate representations) to generate the initial representations used for clustering. For completeness, we also study the effect of sampling from passage clusters (TAS-P) in addition to query clusters. See [39, Section 3] for a detailed description of the TAS algorithm. Note that, the original TAS method also used knowledge distillation in a dual-teacher setup (see [39, Section 2.3]), but we only consider the negative sampling strategy proposed in the same work.

ANCE. Approximate nearest neighbor negative contrastive estimation (ANCE) [131] is a technique that aims to improve the effectiveness of dense retrievers by using hard negatives during training. ANCE accomplishes this by periodically identifying false positive examples using the retrieval model currently being trained. As the hard negatives are periodically updated, we refer to this as an *iterative method*. The false positive examples are then used as hard negatives for the next training epoch. ANCE requires maintaining a continuously updated dense index, which requires significant compute resources. Further details on the ANCE algorithm can be found in [131, Section 4].

Iterative clustered training (ICT-Q and ICT-P). Considering the established need to ensure the presence of hard negatives [131] when training DPR models, we combine intuitions from ANCE and TAS to use clustering to place similar training samples in each training batch. However, unlike with TAS style negative sampling, text representations are generated by the model itself, thus eliminating the need for a teacher model. The representations used in clustering can be either passage or query representations. Similar to ANCE, we also iteratively update the representations used to perform clustering. But, clustered training methods are more efficient than ANCE since they only cluster the training queries or passages (unlike ANCE where a full index of the corpus is built to update the hard negatives). In a typical information retrieval setup, the number of training queries or passages is much smaller than the total number of documents in the corpus.

We provide a formal description of the clustered training method below. Note that we provide the method for clustering passages ICT-P, but the process for clustering queries remains the same with queries ICT-Q replacing passages in the method.

Before each training epoch, we group all training samples S into k clusters with k-means clustering based on the passage representations generated by the passage encoder $E_p(\cdot)$. The objective of the clustering is to minimize the following:

$$\arg_C \min \sum_{i=1}^{k} \sum_{p \in C_i} \mid p - \mu_i \mid^2 . \tag{2.5}$$

Here, μ_i is the centroid of the cluster C_i and p is a passage representation generated by $E_p(\cdot)$. Now, the training samples S are grouped into k clusters C_i where $i \in \{1, \ldots, k\}$.

Next, we split each cluster C containing $|C|$ samples, where $|C| > b$ into sub-clusters c_j such that $|c_j| \leq b$. For a cluster C_i:

$$C_i = \left\{ c \in \{1, \ldots, j\} \mid j = \left\lceil \frac{|C|}{b} \right\rceil \right\}. \tag{2.6}$$

Then, $|c_j| \leq b$. Finally, we combine all sub-clusters containing less than b samples such that each combined cluster contains b or fewer samples until no further combinations are possible. The set of all sub-clusters of size b, all combined sub-clusters, and any sub-clusters that could not be combined becomes the set of training batches for a training epoch.

Then, the training dataset consisting of the set of training batches, built according to the above procedure, is used to train a dense retrieval model. The clustering representations are refreshed periodically during training. We refer to this method as *iterative*

clustered training (ICT), and it comes in two flavors: ICT-Q for clustered training on queries and ICT-P for clustered training on passages.

2.3.4 Summary of negative sampling techniques

We first looked at the in-batch negative sampling technique used in the original DPR paper [50]. Then, we summarized the BM25 negative sampling technique [50]. Next, we described the topic-aware sampling technique [39] and, finally, the ANCE technique [131]. We also introduced iterative clustered training, which combines ideas from [39] and [131] and fills a gap left by previously proposed methods. Our next step is to perform a systematic comparison of these negative sampling methods for dense retrieval under three conditions: *in-distribution*, *out-of-distribution*, and *zero-shot*.

2.4 Experimental Design

We now describe the training and evaluation processes, the datasets used at query and passage level, and the models that we used in the systematic comparison of negative sampling methods for dense retrieval promised at the end of the previous section,

2.4.1 Process

We describe the process of training and evaluating the models below.

Training. The DPR models discussed in this work are trained in two steps. First, the model is pre-finetuned on English and then finetuned on the combined training sets of all available languages. We follow this procedure to train models using each of the negative sampling techniques discussed in Section 2.3.

Evaluation. Each model is evaluated in the three settings described in Section 2.2: in-distribution, out-of-distribution, and zero-shot.

The specific implementation details for each of these processes are discussed in the remainder of this section.

2.4.2 Datasets

Training datasets. All models evaluated in this work are trained on the same datasets. The MS MARCO (MAchine Reading COmprehension) dataset (English) [76] is used for pre-finetuning, followed by fine-tuning on the Mr. TyDi collection of datasets [144].

The Mr. TyDi [144] dataset is a multilingual retrieval benchmark based on the TyDi dataset [20]. Mr. TyDi contains data from eleven typologically diverse languages, some of which are written in Latin script, while the others are written in other scripts (with no two languages sharing the same non-Latin script) [144]. Table 2.2 shows the languages and the number of associated queries and passages for each language.

Testing datasets. Three collections of datasets/benchmarks are used for testing the models in three conditions: *in-distribution*, *out-of-distribution*, and *zero-shot*. For the in-distribution setting, we use the test sets from the Mr. TyDi dataset, as all models were trained on the Mr. TyDi train sets.

Table 2.2: Mr. TyDi languages and the associated number of queries and passages.

Language	# Train queries	# Test queries	# Corpus size
Arabic	12,377	1,081	2,106,586
Bengali	1,713	111	304,059
English	3,547	744	32,907,100
Finnish	6,561	1,254	1,908,757
Indonesian	4,902	829	1,469,399
Japanese	3,697	720	7,000,027
Korean	1,295	421	1,496,126
Russian	5,366	995	9,597,504
Swahili	2,072	670	136,689
Telugu	3,880	646	548,224
Thai	3,319	1,190	568,855

The mMARCO [15] dataset consists of 13 different languages created using machine translation from the MS MARCO dataset. Four of these languages (Arabic, Indonesian, Japanese, Russian) are common to both mMARCO and Mr. TyDi. These four languages are used to evaluate the models in the out-of-distribution setting as they represent languages the models are trained on but created using different methods.

The remaining nine languages from mMARCO (Chinese, Dutch, French, German, Hindi, Italian, Portuguese, Spanish, Vietnamese) are not found in Mr. TyDi, and thus, the models have not been trained on these languages for retrieval. Since mMARCO consists of machine translated datasets, we include human annotated datasets in our test datasets for the languages where we were able to find a retrieval dataset. These datasets are Multi-CPR (E-commerce and entertainment), BSARD (Legal IR), and GerDaLIR (Legal IR) for Chinese, French, and German, respectively. In addition to being unknown languages, these datasets are out-of-domain in terms of data distribution as the retrieved documents are from different domains. Therefore, these languages are used to evaluate the models in the zero-shot setting.

The datasets that are used in this work, and their purpose, are summarized in Table 2.3.

Analysis datasets. Finally, we also use two additional datasets for further analysis (see Section 2.6) beyond our main results. We use the unknown languages from MIRACL [145], an updated version of the Mr. TyDi dataset, to form an in-distribution, unknown language setting. Then, we use the nine smallest datasets (for faster evaluation) from BEIR [111], designed to evaluate out-of-domain performance of retrieval models, to form an out-of-domain, known language setting.

2.4.3 Models

We train DPR models with different negative sampling methods using the Simple Transformers[1] framework, which is based on Huggingface Transformers [128]. All

[1] `https://github.com/ThilinaRajapakse/simpletransformers`

Table 2.3: The datasets used at each stage of the study.

Stage	Condition	Dataset
Training	pFT (pre-finetuning)	MS MARCO
	FT (finetuning)	Mr. TyDi
Testing	In-distribution	Mr. TyDi
	Out-of-distribution	mMARCO (known languages)
	Zero-shot	mMARCO (unknown languages) BSARD (French) GerDaLIR (German) Multi-CPR E-com (Chinese) Multi-CPR video (Chinese)

models we train use mBERT[2] as the starting point, are then pre-finetuned on MS MARCO, and finetuned on the complete training set of Mr. TyDi. They consist of a DPR transformer bi-encoder, with distinct encoders for the queries and passages, initialized from mBERT (*bert-base-multilingual-cased*).

The models trained with TAS negative sampling require two teacher models to perform negative sampling. The first, used for the English pretraining step, is a publicly available DistilBERT[3] model. The second, used for multilingual finetuning, is a publicly available BERT model[4] trained on the Mr. TyDi training set.

To specify the DPR models that we train, we use the abbreviations and acronyms introduced for the corresponding negative sampling methods in Section 2.3.3 and 2.3.3, and summarized in Table 2.1:

- **DPR$_{\text{base}}$**: A DPR model trained without any negative sampling.
- **DPR$_{\text{BM}}$**: A DPR model trained with BM25[5] negatives.
- **TAS-Q**: A DPR model trained with TAS negative sampling on queries.
- **TAS-P**: A DPR model trained with TAS negative sampling on passages.
- **ANCE**: A DPR model trained with ANCE negative sampling.
- **ICT-Q**: A DPR model trained with ICT using training queries.
- **ICT-P**: A DPR model trained with ICT using training passages.

2.4.4 Implementation

Training pipeline. Using the Adam optimizer, each model is trained for 40 epochs with a learning rate of 1e-5 and a batch size of 16. Negative log likelihood loss is used as the

[2] https://github.com/google-research/bert
[3] https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b
[4] https://huggingface.co/castorini/mdpr-tied-pft-msmarco-ft-all
[5] https://github.com/castorini/pyserini

Table 2.4: Results on the Mr. TyDi (in-distribution) datasets. Each dataset is evaluated on two metrics: MRR@100 and Recall@100. Best scores per metric are in bold; second-best are italicized.

Dataset	Metric	ANCE	BM25	ICT-P	ICT-Q	DPR_{BM}	DPR_{base}	TAS-P	TAS-Q
ar	MRR@100	*0.524*	0.247	0.457	0.417	**0.586****	0.305	0.422	0.413
	Recall@100	0.868	0.636	*0.884*	0.882	**0.907****	0.856	0.865	0.859
bn	MRR@100	0.446	0.333	0.492	0.469	**0.563**	0.398	0.454	*0.494*
	Recall@100	0.847	0.730	**0.919**	**0.919**	0.901	0.892	0.901	*0.910*
fi	MRR@100	0.419	0.161	*0.431*	0.396	**0.471****	0.259	0.373	0.357
	Recall@100	0.828	0.507	0.856	*0.866*	**0.881**	0.823	0.836	0.854
id	MRR@100	*0.475*	0.288	0.427	0.413	**0.502***	0.321	0.402	0.382
	Recall@100	0.870	0.742	0.877	*0.878*	**0.903****	0.864	0.876	0.876
ja	MRR@100	0.333	0.173	*0.362*	0.319	**0.430****	0.215	0.314	0.284
	Recall@100	0.794	0.624	0.835	*0.844*	**0.864***	0.808	0.814	0.815
ko	MRR@100	*0.354*	0.196	0.343	0.323	**0.399****	0.239	0.316	0.306
	Recall@100	0.753	0.360	0.753	*0.760*	**0.805****	0.720	0.732	0.724
ru	MRR@100	*0.410*	0.209	0.350	0.326	**0.436***	0.241	0.317	0.294
	Recall@100	0.821	0.462	0.843	*0.859*	**0.871**	0.813	0.826	0.838
sw	MRR@100	0.397	0.363	*0.530*	0.492	**0.530**	0.386	0.507	0.438
	Recall@100	0.785	0.743	0.863	**0.881**	*0.870*	0.852	0.854	0.867
te	MRR@100	0.677	0.186	*0.703*	0.579	**0.774****	0.469	0.594	0.461
	Recall@100	0.921	0.426	**0.967**	0.964	*0.966*	0.947	0.966	0.966
th	MRR@100	0.416	0.161	*0.423*	0.384	**0.489****	0.288	0.392	0.353
	Recall@100	0.807	0.489	0.887	**0.905**	0.842	0.863	0.873	*0.893*

loss function. This procedure is followed separately for both the pre-finetuning (pFT) and the finetuning (FT) steps. The model, initialized from mBERT, is pre-finetuned on the MS MARCO dataset for 40 epochs and is then finetuned for another 40 epochs on the combined training sets of Mr. TyDi following the setup in [146]. The representations are updated every 10 epochs for the iterative methods.

Evaluation and testing. Following [144, 146], we report the MRR and Recall@100 scores for each test dataset. We test the seven DPR models on two datasets, the Mr. TyDi benchmark and the mMARCO dataset, under three settings. We report results under the three conditions, in-distribution (Mr. TyDi test sets), out-of-distribution (mMARCO languages that are present in Mr. TyDi), and zero-shot (mMARCO languages that are not present in Mr. TyDi).

We consider observed differences to be statistically significant if $p < 0.05$ in a paired t-test. We write ** to indicate $p < 0.01$ and * to indicate $p < 0.05$. Statistical significance is computed between each dataset's highest and second-highest scores.

Table 2.5: Results on the MMARCO OOD datasets. Each dataset is evaluated on two metrics: MRR@100 and Recall@100. Best scores per metric are in **bold**; second-best are *italicized*.

Dataset	Metric	ANCE	BM25	ICT-P	ICT-Q	DPR_{BM}	DPR_{base}	TAS-P	TAS-Q
ar	MRR@100	0.092	0.106	**0.138****	*0.125*	0.105	0.103	0.124	0.116
	Recall@100	0.355	0.375	**0.461****	*0.441*	0.370	0.409	0.428	0.433
id	MRR@100	0.114	**0.154****	*0.140*	0.124	0.118	0.099	0.117	0.115
	Recall@100	0.425	**0.541****	*0.509*	0.487	0.439	0.435	0.474	0.470
ja	MRR@100	0.128	0.136	**0.167****	*0.157*	0.139	0.129	0.153	0.147
	Recall@100	0.462	0.469	**0.545**	*0.540*	0.474	0.506	0.522	0.524
ru	MRR@100	0.134	0.102	**0.157****	*0.143*	0.139	0.124	0.137	0.136
	Recall@100	0.480	0.354	**0.541****	*0.531*	0.482	0.498	0.503	0.509

2.5 Results

2.5.1 In-distribution results (Known language)

Table 2.4 shows the MRR@100 and Recall@100 scores obtained by each model on the Mr. TyDi test sets (the in-distribution setting).

We find that DPR_{BM} outperforms all other methods across all languages (statistically significant for all but two languages) in the in-distribution setting. This indicates that simple BM25 negatives are surprisingly effective when training multilingual dense retrievers. While Hofstätter et al. [39], who introduced TAS-style negative sampling, demonstrated impressive retrieval effectiveness, our results indicate that most of the improvements possibly came from the other techniques used in [39] (e.g., knowledge distillation).

ICT-P, obtains the second-best performance with the passage clustering approach outperforming query clustering across the board. We see the same pattern with TAS clustering where TAS-P outperforms TAS-Q. We believe the better performance of clustering passages instead of queries is likely due to passages being longer and containing more information, leading to better clusters and harder negatives.

ANCE performance is close to ICT-P performance, with ICT-P obtaining higher MRR@100 on six languages while ANCE obtains higher MRR@100 on four languages. In terms of Recall@100, ICT-P gets higher scores than ANCE on all languages except Korean (tie).

DPR_{base} with random in-batch negatives performs the worst out of all methods, confirming that effective negative sampling methods are essential to train good dense retrievers in a multilingual setting.

Based on these results, we recommend using negative sampling based on BM25 hard negatives when training a multilingual dense retrieval model if the model is primarily tasked with retrieval in an in-distribution setting. We further analyze the effectiveness of DPR_{BM} in the in-distribution setting in Section 2.6.1.

Table 2.6: Results on the MMARCO zero-shot datasets. Each dataset is evaluated on two metrics: MRR@100 and Recall@100. Best scores per metric are in **bold**; second-best are *italicized*.

Dataset	Metric	ANCE	BM25	ICT-P	ICT-Q	DPR_{BM}	DPR_{base}	TAS-P	TAS-Q
zh	MRR@100	0.136	0.119	**0.169****	*0.164*	0.145	0.135	0.154	0.154
	Recall@100	0.499	0.451	*0.576*	**0.578**	0.515	0.529	0.543	0.547
nl	MRR@100	*0.155*	0.142	**0.172****	0.154	0.150	0.128	0.148	0.142
	Recall@100	0.518	0.488	**0.582****	*0.567*	0.513	0.517	0.534	0.535
fr	MRR@100	0.159	0.149	**0.186****	*0.167*	0.157	0.139	0.159	0.154
	Recall@100	0.548	0.519	**0.611**	*0.608*	0.551	0.560	0.567	0.573
de	MRR@100	0.156	0.135	**0.175****	*0.158*	0.158	0.137	0.156	0.150
	Recall@100	0.517	0.464	**0.575****	*0.561*	0.519	0.530	0.534	0.538
hi	MRR@100	0.087	*0.134*	**0.141**	0.130	0.107	0.111	0.131	0.128
	Recall@100	0.324	*0.470*	**0.470**	0.462	0.387	0.435	0.453	0.449
it	MRR@100	0.154	0.145	**0.179****	*0.166*	0.154	0.137	0.155	0.151
	Recall@100	0.543	0.499	**0.604****	*0.593*	0.541	0.546	0.561	0.560
pt	MRR@100	0.156	0.158	**0.185****	*0.168*	0.152	0.140	0.160	0.160
	Recall@100	0.537	0.544	**0.604****	*0.593*	0.534	0.542	0.563	0.568
es	MRR@100	0.170	0.159	**0.196****	*0.177*	0.164	0.147	0.168	0.166
	Recall@100	0.566	0.551	**0.635****	*0.624*	0.558	0.578	0.590	0.594
vi	MRR@100	0.118	*0.140*	**0.141**	0.126	0.121	0.106	0.120	0.118
	Recall@100	0.423	**0.508**	*0.498*	0.481	0.439	0.444	0.461	0.459

2.5.2 Out-of-distribution results (Known language)

Table 2.5 shows the MRR@100 and Recall@100 scores for each model on the out-of-distribution language datasets from mMARCO. In this setting, the two variants of iterative clustered training, ICT-P and ICT-Q, outperform all other negative sampling methods. Similar to the in-distribution setting, we again see that passage-based clustering yields better results than query-based clustering, with the ICT-P model obtaining the highest scores of the negative sampling methods on all four languages (statistically significant).

We compare ICT-P and DPR_{BM} on BEIR datasets (English language) to confirm our findings in the out-of-distribution setting free from machine translation artifacts in Section 2.6.2.

These results indicate that the clustered training methods (both ICT-P and ICT-Q) provide superior out-of-distribution results compared to the other negative sampling methods.

2.5.3 Zero-shot results (Unknown language)

Next, Table 2.6 shows the MRR@100 and Recal@100 scores obtained by each model on the zero-shot languages. This is the setting that we are most interested in as it

Table 2.7: Results on the other zero-shot datasets. Each dataset is evaluated on two metrics: MRR@100 and Recall@100. Best scores per metric are in **bold**; second-best are *italicized*.

Dataset	Metric	ANCE	BM25	ICT-P	ICT-Q	DPR_{BM}	DPR_{base}	TAS-P	TAS-Q
BSARD	MRR@100	0.150	**0.225***	0.161	*0.168*	0.146	0.161	0.147	0.148
	Recall@100	0.310	**0.466**	0.368	0.398	0.352	*0.430*	0.383	0.348
GerDaLIR	MRR@100	0.120	**0.199****	*0.163*	0.151	0.104	0.148	0.158	0.144
	Recall@100	0.349	**0.650****	*0.422*	0.401	0.319	0.401	0.409	0.387
Multi-CPR Ecom	MRR@100	0.118	**0.293****	0.191	0.192	0.118	0.190	0.188	*0.203*
	Recall@100	0.399	**0.711****	0.530	0.549	0.409	0.550	0.542	*0.552*
Multi-CPR Video	MRR@100	0.112	**0.230**	0.203	*0.210*	0.124	0.188	0.199	0.204
	Recall@100	0.449	**0.735****	0.634	0.648	0.469	0.601	0.650	*0.666*

represents the real-world scenario of using a multilingual dense retrieval model for monolingual retrieval in a language that it has not been trained on for retrieval.

Similar to the out-of-distribution setting, the ICT methods outperform all other methods on all zero-shot languages (statistically significant). Again, we see that clustering passages yields better results compared to clustering queries for both ICT and TAS-style clustering.

We also report results on four additional retrieval datasets in zero-shot languages (French, German, and Chinese) to confirm that the results on the mMARCO datasets are not due to machine translation artifacts. In addition to these datasets being zero-shot languages, they are out-of-domain datasets as described in Section 2.4.2. Table 2.7 shows that the ICT methods outperform the other negative sampling methods on these human annotated datasets. However, we see that the baseline BM25 model outperforms the dense retrieval method on these datasets. This agrees with findings from Thakur et al. [111] that the lexical-based BM25 method can be superior to dense retrievers in an out-of-domain setting.

The results from the zero-shot language tests shows that the iterative ICT methods (ICT-P and ICT-Q) show superior domain adaptability as well as adaptability to new languages compared to the other negative sampling methods.

2.5.4 Summary of results

Finally, we summarize the findings from this section.

- DPR_{BM} demonstrates impressive results on in-distribution test sets, outperforming all other methods.
- ICT outperforms the other negative sampling methods in out-of-distribution and zero-shot settings.
- TAS-style clustering using an external model underperforms other negative sampling techniques in a multilingual setting.

Table 2.8: Results on the MIRACL zero-shot languages.

	MRR@100		Recall@100	
Dataset	ICT-P	DPR_{BM}	ICT-P	DPR_{BM}
German	0.435	**0.458**	0.767	**0.772**
Spanish	0.512	**0.572****	**0.712**	0.700
Persian	0.434	**0.461**	**0.805****	0.764
French	0.389	**0.459****	0.782	**0.798**
Hindi	0.412	**0.451***	**0.732**	0.727
Yoruba	0.540	**0.577**	**0.866**	0.861
Chinese	**0.517**	0.511	**0.854****	0.815

2.6 Analysis

We look at two variants of the three main settings from Section 2.5 and consider the two best-performing negative sampling methods, namely, ICT-P and DPR_{BM}.

2.6.1 In-distribution data, unknown language

We introduce a variant of the in-distribution setting to further analyze the effectiveness of DPR_{BM} under in-distribution testing conditions. We compare the performance of the DPR_{BM} and ICT-P models (best and second-best results in the in-distribution setting, respectively) on the MIRACL languages that do not appear in Mr. TyDi. In this setting, we test on data that is *in-distribution* in terms of data collection, annotation, and sources, but *zero-shot* in terms of the language.

In Table 2.8 we see that DPR_{BM} outperforms ICT-P on in-distribution data in terms of ranking metrics even when the language is new to the model. Interestingly, ICT-P gets better recall than DPR_{BM}, unlike what we saw in Section 2.5.1. This suggests that the better generalizability of the ICT-P model helps it adapt to the newer languages. However, DPR_{BM} still has better overall performance in the in-distribution setting which strengthens our recommendation to use DPR_{BM} for in-distribution multilingual retrieval scenarios.

2.6.2 Out-of-domain data, known language

Now, we compare the performance of ICT-P and DPR_{BM} on BEIR which presents a setting where the data is *out-of-domain* (*out-of-distribution* and new domains) but in a known language (English).

In Table 2.9, we see that the ICT-P model outperforms the DPR_{BM} model in the *out-of-domain*, known language setting. This serves to confirm our recommendation to use ICT-P models in a multilingual retrieval setting where generalizability to new data distributions or domains is needed. We also report nDCG@10 for the BEIR results as this is the official metric used in [111].

Table 2.9: Results on the BEIR datasets.

	nDCG@10		MRR@100		Recall@100	
Dataset	ICT-P	DPR_{BM}	ICT-P	DPR_{BM}	ICT-P	DPR_{BM}
ArguAna	**0.304****	0.235	**0.214****	0.165	**0.891****	0.852
CQA Dup Stack	**0.207****	0.147	**0.219****	0.154	**0.406****	0.320
DBPedia	0.230	**0.238**	0.510	**0.538**	0.324	**0.332**
FiQa	**0.205****	0.181	**0.265***	0.239	**0.475****	0.432
NFCorpus	**0.214****	0.192	**0.395**	0.386	**0.202****	0.182
Quora	**0.770****	0.264	**0.760****	0.256	**0.967****	0.613
SciDocs	**0.084***	0.077	**0.175***	0.156	0.203	**0.204**
SciFact	**0.420**	0.396	**0.399**	0.370	0.753	**0.775**
TREC-COVID	**0.464****	0.355	**0.696**	0.583	**0.057**	0.051

2.7 Related Work

Dense retrieval. Traditionally, passage retrieval has been performed using sparse retrieval methods such as BM25 [118]. Karpukhin et al. [50] show that transformer-based [115] dual-encoder models can surpass traditional sparse methods by using the ability of transformer models to represent semantic meaning, unlike classic keyword-based methods. However, later work [97, 98, 111, 138] has shown that dense retrieval models, specifically dual-encoder models, struggle to generalize to out-of-distribution data.

Improved training regimes to boost generalizability. Prior work has proposed a range of data generation, data augmentation, and data selection techniques for improving the effectiveness of dense retrieval models. Since the release of the BEIR benchmark [111], researchers have begun to specifically consider whether these techniques improve out-of-distribution generalization as well as in-domain effectiveness. Negative sampling techniques [39, 50, 131] represent one such approach to improve the generalizability of dense retrievers. We focus on their effectiveness in boosting the generalizability of dense retrievers in a multilingual setting.

Negative sampling for dense retrieval. Early dense retrieval models like DPR [50] select their negative training examples from a combination of false positives identified by BM25 and from other queries in the same training batch ("in-batch negatives"). ANCE [131] demonstrates the importance of selecting hard negative training examples, which it accomplishes by periodically identifying false positive examples using the retrieval model currently being trained. Although the original work focuses on in-distribution performance and does not explore out-of-distribution or zero-shot retrieval, later work [111] shows that out-of-distribution results also improve. ANCE requires maintaining a continuously updated dense index, which requires significant compute resources, though these requirements can be reduced by freezing the document encoder for part of training [137]. Alternatively, computational requirements can be reduced by caching negative examples rather than periodically recomputing the entire index [65, 132].

Rather than using a dense retrieval model to mine hard negative examples, TAS-B [39] creates difficult training batches by clustering queries once at the start of training and then samples from those clusters to build training batches. TAS-B combines this with knowledge distillation to improve the retrieval performance of dense retrievers. TAS-B uses a separately trained BERT model to generate the representations for the clustering. Therefore, the effectiveness of TAS-B is dependent on the availability of a teacher model, which can be a restrictive constraint in a multilingual setting.

A systematic comparison of negative sampling methods for dense retrieval under different generalizability conditions (in-distribution, out-of-distribution, zero-shot) is missing. This is the gap that we fill. We consider a rich multilingual setting that allows us to formulate all three conditions, and we discover a gap in the choices available for negative sampling for dense retrieval so far.

Multilingual retrieval. To understand the generalizability of dense retrievers we consider a multilingual setting, that naturally allows us to consider challenging in-distribution, out-of-distribution, and zero-shot settings. The literature on information retrieval in multiple languages is rich. Cross-lingual retrieval (queries in one language and passages in another), in particular, has been the focus of many publications, and we refer the reader to [29, 140] for recent surveys on this area. However, our work focuses on multilingual retrieval, where both queries and passages are in the same language (monolingual), but the models used support monolingual retrieval in many languages. However, cross-lingual and multilingual retrieval both benefit from cross-lingual transfer capabilities, particularly of large language models. The zero-shot knowledge transfer ability of large language models has previously been studied in [70, 101].

Zhang et al. [146] offer a comprehensive guide on training multilingual dense retrievers based on the Mr. TyDi benchmark and focuses on *monolingual retrieval* with *multilingual retrievers*. We also focus on the same task, however, we pay careful attention to the generalizability of the multilingual dense retrievers, both for out-of-distribution data and for new languages. We analyze the effectiveness of different negative sampling methods under the in-distribution, out-of-distribution, and zero-shot conditions in a multilingual setting, and investigate whether existing findings from English language research generalizes to these new conditions.

2.8 Discussion

We discuss the strengths and weaknesses of the negative sampling techniques we have investigated and our central reproducibility question, viz. how existing findings from English language models and datasets generalize to the setting of monolingual retrieval with multilingual models. We also discuss the implications for the use of multilingual dense retrieval models in practice.

Generalizability of English language findings to the multilingual setting. Broadly speaking, we found that existing findings on English language retrieval (the importance of the presence of hard negatives) generalize to the multilingual domain. Good hard negative sampling methods yields significant improvements in multilingual retrieval quality. However, one of the most effective negative sampling methods, TAS, is less effective in the multilingual setting. We believe that this is due to the comparative

lack of effective teacher models that can be employed to generate the query or passage representations for clustering. Therefore, the iterative negative mining methods (ICT-P, ICT-Q, and ANCE) demonstrate superior retrieval quality over the non-iterative methods as they do not require external models to perform negative sampling.

DPR_{base} (Random negatives) requires no additional data, models, or hardware resources to be used. It also requires the least training time and is the simplest to implement, but it is also the least effective of the methods we have investigated. We recommend using DPR_{base} only when no additional data, models, or hardware resources are available and training time efficiency is the primary concern.

DPR_{BM} (Non-iterative BM25 negatives) requires an external BM25 model to find negatives. However, BM25 is a sparse retrieval method and is generally much faster and cheaper than dense retrieval methods. Therefore, DPR_{BM} can be used with little additional effort in most cases. DPR_{BM} does not require any additional hardware resources to use.

Using BM25 negatives is surprisingly effective as long as there is little distributional shift between the training and test data, demonstrating superior retrieval quality in the in-distribution setting compared to the other methods. Based on these factors, we recommend using DPR_{BM} when the model is used mostly for in-domain retrieval. This contradicts in-distribution English language findings where negative sampling methods like ANCE and TAS were developed in order to improve over BM25 negatives.

ICT-Q and ICT-P (Iterative, clustering-based negatives) ICT-P demonstrated superior performance in all three settings between the two ICT methods. While ICT-Q is marginally faster in the clustering phases during training due to queries usually being shorter than passages, we do not believe this makes a practical difference. Therefore, we recommend using ICT-P over ICT-Q.

Overall, ICT-P obtained the best results in two out of three settings (out-of-distribution and zero-shot). Based on this, we recommend using ICT-P as the negative sampling method in most multilingual retrieval scenarios except for the special case detailed above (the model is intended for use in an in-distribution setting).

TAS-Q and TAS-P (Non-iterative, clustering-based negatives) While both TAS-Q and TAS-P outperform DPR_{base} across all three settings, they perform similar or inferior to the other negative sampling methods. In addition to this, these two negative sampling methods require an external model to perform the clustering in order to sample similar queries/passages for training batches.

We believe that a possible reason for TAS negative sampling to underperform in a multilingual setting is that it relies heavily on the external model used for clustering to find good hard negatives. The external models available in the multilingual retrieval setting tend to be less effective and reliable compared to the models available for English such as ColBERT [51]. Furthermore, while the approach in [39] performs well in an English language setting, it uses other techniques, such as knowledge distillation, in addition to negative sampling explored in this work. Wang and Zuccon [123] also found that TAS-style negative sampling, without knowledge distillation, can underperform random negatives in an English language setting indicating that the success of [39] could largely have been influenced by the effectiveness of knowledge distillation. Due to these limitations, we do not recommend using TAS-Q or TAS-P for negative sampling in a

multilingual retrieval setting.

ANCE (Iterative, full-corpus mined negatives) ANCE is fairly effective in multilingual retrieval in all three settings that we considered. However, it has the highest hardware resource requirements (for training) of all the methods considered in this work. Periodically building a dense index of the full document collection increases training time significantly compared to the other methods. Therefore, we recommend using ICT-P instead which builds on similar ideas as ANCE, but is more efficient to train, and also outperformed ANCE in all three settings.

2.9 Conclusion

We studied the generalizability of earlier insights into the effectiveness of negative sampling methods for multilingual retrieval under in-distribution, out-of-distribution, and zero-shot conditions. We identified a gap in the literature, and by combining earlier insights, introduced an iterative, clustering-based method, to fill this gap.

Our experiments confirmed the choice of the negative sampling method used to train dense retrievers has a significant impact on their multilingual retrieval effectiveness. This answers **RQ 1.1**, showing that prior findings from negative sampling studies in English language dense retrieval remain valid for multilingual retrieval. However, TAS, a highly effective clustering-based negative sampling method in English, underperformed other negative sampling methods in a multilingual setting contradicting existing findings. On the other hand, iterative negative sampling methods performed well in the multilingual setting, maintaining their effectiveness from prior English language work. The comparative lack of effective teacher models in a multilingual setting poses a barrier for methods that rely on external representations, such as TAS. Iterative negative sampling methods do not require external representations; instead, they use the representations from the model being trained to find hard negatives and, therefore, succeed in finding good hard negatives even as the model learns even in a multilingual setting.

Interestingly, the best negative sampling method depends on whether the model is tested on in-distribution or out-of-distribution data. For the in-distribution setting, simple BM25 negatives (DPR_{BM}) obtained the best performance and, therefore, we recommend using DPR_{BM} for in-distribution multilingual retrieval tasks. For out-of-distribution and zero-shot settings, we found that ICT-P has the best performance. Based on this, we recommend using ICT-P for out-of-distribution or zero-shot scenarios. Unless the situation clearly calls for in-distribution performance only, our overall recommendation is also to use ICT-P due to its better generalizability. These experiments also answer **RQ 1.2** and shows that ICT-P offers the best overall performance in multilingual dense retrieval. Taken together, these findings answer **RQ 1**: the choice of negative sampling strategy has a decisive effect on cross-domain and cross-lingual generalization, with iterative clustered training (ICT-P) providing the most robust performance across distribution shifts, while BM25 negatives remain optimal in purely in-distribution settings.

As to limitations of our work, we have constrained ourselves to the DPR architecture and have not explored the benefits of clustered training for other architectures. We only considered a contrastive learning setup and did not experiment with other training

methods such as knowledge distillation (consistently good teacher models are rarer in the multilingual retrieval setting than in English only retrieval). In future work we intend to generalize our findings to different architectures and training setups, and explore interactions between negative sampling methods and other training setups.

3

Improving the Generalizability of the Dense Passage Retriever Using Generated Datasets

3.1 Introduction

Recently, a number of transformer-based dense retrieval models have achieved state-of-the-art results on various benchmark datasets [50, 51, 131]. The dense passage retriever (DPR) architecture consists of two encoder models, typically BERT models [26], which encode the query and the passages separately. A simple similarity metric, such as the inner product or cosine distance, is then used to compute the relevance of a passage for a query.

An advantage of the DPR architecture is that passage representations can be pre-computed offline and built into an index with relatively small computational cost, making it a preferred model over recent proposals such as, e.g., ColBERT [51] and ANCE [131] with higher computational cost for training and/or retrieval. At runtime, the query encoder is used to compute a dense representation for the query and approximate nearest neighbor methods are used to find the most relevant passage.

A disadvantage of this approach is that a mismatch may exist between the information available to the passage encoder and the information available to the query encoder. As the training objective forces the passage and query encoders to generate representations that are similar, we hypothesize that the passage encoder (which has access to more information) learns to discard information that is not relevant to the query in a given training query-passage pair. The issue is exacerbated by the fact that most retrieval datasets and benchmarks contain far more passages with only one query from a given passage than passages with multiple queries per passage (see Table 3.1). In such situations, the model is not sufficiently penalized against learning to discard information that is not relevant to the (single) query that is asked from a given passage.

We hypothesize that a DPR model trained on datasets where a given passage typically has one associated query generalizes poorly to other datasets, new types of queries

This chapter was published as T. C. Rajapakse and M. de Rijke. Improving the generalizability of the dense passage retriever using generated datasets. In *European Conference on Information Retrieval*, pages 94–109. Springer, 2023.

or topics, or both. We investigate this hypothesis by testing the zero-shot performance of the pretrained DPR model (from [50], which is trained on NQ [54]) in both out-of-distribution and out-of-domain settings. Here, we define *out-of-distribution* to be datasets that share the same passage corpus but with queries collected at different times and/or using different methods, and *out-of-domain* to be datasets with their own unique passage collection typically focused on a particular domain (see Section 3.4.1).

Having established that a DPR model trained on datasets where a given passage typically has one associated query, generalizes poorly, we propose a treatment to help improve out-of-distribution and out-of-domain performance. We synthetically generate training datasets where the passages typically have multiple queries from any given passage. The generation pipeline consists of a NER model to tag entities, a sequence-to-sequence model to generate queries, and a question answering model to filter out bad queries (see Section 3.3.1).

Our results show that training on data with multiple queries per passage leads to a DPR model with better generalizability to both out-of-distribution and out-of-domain data. In both settings, our DPR model trained on multiple queries per passage data easily outperforms the baseline DPR model trained on mostly single query per passage data (NQ).

In summary, then, this chapter asks the following research question:

RQ 2 Does training a DPR model on data containing multiple queries per passage improve the generalizability of the model?

In the out-of-distribution setting, the pre-trained DPR model [50], serving as the baseline, and our DPR model trained on generated queries with multiple queries per passage are tested, zero-shot, on six datasets. Our model achieves higher retrieval accuracy on five out of the six datasets demonstrating that training data containing multiple queries per passage does improve the generalizability of dense retrievers to out-of-distribution queries.

The picture becomes even clearer in the out-of-domain setting where our model outperforms the pretrained DPR model on 12 out of 13 datasets. Training DPR models on passages with multiple associated queries prevents the context encoder from (exclusively) focusing on a specific detail or piece of information in the passage, leading to a better generalized retrieval model.

Our analysis of increasing the size of the set of generated queries with multiple queries per passage as a way to improve the generalizability of dense retrievers indicates a subtle balance. While the model trained on the largest training dataset does achieve higher scores compared to the others, the improvements are relatively minor. But, these relatively minor improvements come at a significantly higher costs in terms of compute and training time. Even the smallest generated dataset with multiple queries per passage performs competitively with larger generated datasets and handily outperforms the pre-trained model trained on mostly single query per passage data.

3.2 Related Work

Passage retrieval. Passage retrieval has classically been performed using sparse retrieval methods such as BM25 [118]. Recently, transformer-based dense retrieval methods have garnered interest as the performance of dense retrieval methods surpasses that of traditional sparse methods [50, 51, 131]. A dense passage retriever indexes a collection of passages in a low-dimensional and continuous space, such that the top-k passages are relevant to a given query [50]. Here, the size of the passage collection is typically very large (21M passages in this work and in [50]) and k is very small (e.g., 20–100). Going beyond *in-distribution* and *in-domain* testing, we focus on generalizability to new data which can be *out-of-distribution* and *out-of-domain*.

Test collections. The Benchmarking-IR (BEIR) [111] test collection was introduced to facilitate the effectiveness of retrieval models in out-of-domain settings. It provides a collection of 18 datasets (13 of which are readily available) from diverse retrieval tasks and domains. Thakur et al. [111] also highlight considerable room for improvement in the generalization capabilities of dense retrieval models. Our work aims to improve the generalizability of dense retrievers by using synthetic datasets with specially chosen composition of data (multiple queries per passage).

Automatically generated collections. Automatically generating training, development and test collections for retrieval has a long history in information retrieval. Examples include test collections for bibliographic systems [108], known-item test collections [7], desktop search [52], web search [4], test collections for academic search [11]. Berendsen et al. [12] focus on test collection generation to improve robustness for tuning and learning. A comprehensive approach to simulated test collection building with considerable attention to privacy preservation is offered in [38]. What we add on top of this is test collection building with a specific focus on generalizability by preventing overfitting.

3.3 Methodology

We train DPR models on generated query datasets and compare their retrieval performance against the pre-trained model on the test datasets.

3.3.1 Dataset generation process

For our dataset generation process, we follow the steps below:

(1) Identify potential answers to questions to be generated;

(2) Generate queries that are answered by one of the potential answers; and

(3) Filter out bad queries, that is, queries that are unanswerable or do not end with a question mark.

Identifying potential answers. We train a token classification model to identify words or phrases from a passage that could serve as potential answers to queries. The trained

model is then used to tag potential answers for each passage in a dataset. This process enables us to find all potential answers in a passage, which is critical to ensure that there are sufficient queries from any given passage.

Generating queries. The passages, along with the tagged answers, are fed to a sequence-to-sequence model that generates a query for each passage-answer pair. Each passage can have multiple associated answers, resulting in multiple queries from the same passage. This ensures that there are queries related to most, if not all, entities found in a given passage.

Filtering queries. The generated queries are filtered to remove potentially unanswerable queries (from the originating passage). To find such queries, we feed the passages and queries to a question answering (QA) model and discard queries where the QA model answer does not match the original tagged answer. We also discard queries that contain more than one sentence or do not end with a question mark (?). This is to ensure that all the generated queries used for training are reasonable queries (see Section 3.4.2) and provide a good training signal for the model being trained on them.

3.3.2 Training the retriever

We build training datasets by generating queries following the procedure given in Section 3.3.1. The generation process ensures that most passages in the training datasets have multiple queries associated with them. We train bi-encoder retrieval models on these training datasets.

3.4 Experimental Setup

3.4.1 Datasets

Most popular open-domain retrieval datasets contain a much larger number of passages with only a single query originating from it than passages with multiple queries. Table 3.1 shows the frequency of passages with a given number of queries originating from the passage for the five datasets used in [50] as well as the five datasets that were generated. The Wikipedia collection and five of the datasets used (NQ, Trivia QA, Curated TREC, Web Questions, and SQuAD) are the same versions provided by [50] available on GitHub.[1]

Out-of-distribution test datasets. To test the models on out-of-distribution data, we use the four datasets available from [50] that were not used in training the baseline model, namely Trivia QA, Curated TREC, Web Questions, and SQuAD. In addition to these four, we include two generated test datasets. The first of these is generated from the NQ *dev* passages and the second is generated from randomly selected Wikipedia passages. This results in a total of six out-of-distribution test datasets. As these datasets use the same passage collection but contain queries collected or generated using different approaches, we consider the datasets to be *out-of-distribution* but *in-domain*.

[1] `https://github.com/facebookresearch/DPR`

Table 3.1: Frequency of passages with a given number of queries originating from the passage.

	Number of queries/passage		
Dataset	1	2	≥2
Natural Questions	32,155	4,973	3,542
Trivia QA	43,401	5,308	1,793
Curated TREC	990	41	16
Web Questions	2,019	148	46
SQuAD	8,468	6,056	11,790
Generated from NQ train	2,784	3,418	30,120
Wikipedia passages (˜58k) single	58,880	0	0
Wikipedia passages (˜58k) multi	16,634	19,641	985
Wikipedia passages (˜236k)	19,487	18,061	41,308
Wikipedia passages (˜786k)	62,264	60,472	137,266

Out-of-domain test datasets. We use the 13 readily available datasets from [111], each with their own distinct passage collection, to test the models on out-of-domain data. The datasets are as follows: TREC-COVID [117], NFCorpus [17], HotpotQA [136], FiQA-2018 [71], ArguAna [119], Touché-2020 [14], CQADupStack [40], Quora, DBPedia [37], SCIDOCS [22], FEVER [112], Climate-FEVER [27], and SciFact [120]. These datasets cover multiple domains, including bio-medical, Wikipedia/general, finance, news, and scientific domains.

3.4.2 Generation pipeline

Named entity recognition model for tagging answers. The named entity recognition model is a RoBERTa [68] model trained on the large NER dataset (1 million sentences) from Naman Jaswani on Kaggle,[2] with the tags: *Organization*, *Person*, *Location*, *Date*, *Time*, *Money*, *Percent*, *Facility*, and *Geo-Political Entity* (GPE). The RoBERTa model, trained on a large NER dataset, ensures that we find all the entities in a passage.

MACAW model for query generation. The pretrained *MACAW* [107] model (3 billion parameters) is used to generate the queries. It is a strong sequence-to-sequence question generation model (among other tasks) based on the T5 model [86]. This model is capable of generating queries for each entity found in the passage such that they are relevant to the context of the passage.

Question answering model for query filtering. A RoBERTa [68] model trained on the SQuAD dataset is used to filter out potential bad queries in the generated datasets. The RoBERTa model is a question answering model that is good at extractive question answering. We can reasonably assume that the questions the model is incapable of answering are most likely flawed.

This generation pipeline results in queries that are typically relevant and answerable

[2] `https://www.kaggle.com/namanj27/ner-dataset`

Table 3.2: Examples of generated queries and answers for a randomly sampled passage.

Passage	Generated query	Generated answer	Related	Answerable
Sirocco (play) is a play in four acts by Noël Coward. It opened at Daly's Theatre on November 24, 1927, directed by Basil Dean. Ivor Novello was part of the original cast. The London opening met with harsh reception...	Sirocco was first performed at which theater in London?	Daly's Theatre	Yes	Yes
	When did the first performance of Sirocco take place?	November 24, 1927	Yes	Yes
	Which actor played the role of Sirocco in the original production?	Ivor Novello	Yes	No
	Who wrote the play *Sirocco*?	Noël Coward	Yes	Yes
	Who directed the first production of *Sirocco*?	Basil Dean	Yes	Yes

from their passages of origin. We found 92% of queries to be relevant, and 86% to be answerable from their passages of origin, based on a randomly sampled set of 50 queries (example shown in Table 3.2).

3.4.3 Retrieval pipeline

The architecture of the retrieval model is identical to [50], i.e., a bi-encoder architecture consisting of two BERT [26] encoders, one for encoding the passages/contexts and the other for encoding the queries. We also use the same hyperparameters as [50] except for the batch size, where we use a batch size of 80 vs. a batch size of 120 due to resource limitations.

We choose the DPR [50] model as our architecture of choice to avoid introducing any confounding factors in our analysis. Other architectures, notably the late interaction based ColBERT [51] architecture, has demonstrated superior retrieval accuracy over the original DPR [50] architecture. However, ColBERT has higher latency and much larger space footprints for indices. As our work is focused on the composition of data, the simpler and more straightforward architecture of DPR is better suited to our analysis. Furthermore, the higher resource demands and complexity of ColBERT makes it a less viable option compared to DPR in any setting with even moderate computational resource constraints.

We build five training datasets by generating queries following the procedure given in Section 3.3.1. One dataset is built by generating queries from the same passages used in the NQ train set, while the other four are from randomly selected Wikipedia passages. A bi-encoder DPR model, starting from the pretrained BERT [26] weights, is trained on each of these five datasets.

While positive training examples (matching query and document pairs) are available directly in retrieval datasets, negative training examples must be selected from the set of all documents. The original DPR model is trained using a combination of in-batch negatives (the positive documents of all other queries in the batch used as negatives for a given query) and BM25 selected negatives (highest ranked document retrieved by BM25, which does not contain the answer to the query). In our work, we simply use the in-batch

negatives as the negative examples leaving improvements from more complex negative selection strategies for future work as our results demonstrate improved generalizability even without using hard negatives.

3.4.4 Experiment

We use two models trained on two different datasets to compare the generalizability of DPR models trained on data with multiple queries per passage versus DPR models trained on data with mostly a single query per passage. The pre-trained DPR model from [50], trained on NQ with mostly single query per passage data, is used as the baseline model to be compared against our model trained 58,880 generated queries containing mostly multiple queries per passage data (*58k generated*).

The two models are tested in both the out-of-distribution (6 datasets) and the out-of-domain settings (13 datasets). *Top-100 accuracy* is used as the evaluation metric for the out-of-distribution setting while *recall@100* is used as the evaluation metric for the out-of-domain setting. The decision to use two different metrics is motivated by the fact that the set of all relevant passages is only available for the out-of-domain datasets, which is necessary to calculate recall. Only the true answers are available for the out-of-distribution datasets, so we calculate top-100 accuracy by checking whether the true answer is present in any of the top-100 retrieved documents. In addition to this, we also report MRR@100 (Mean Reciprocal Rank) for all experiments.

3.5 Results

We report results from the baseline pretrained model trained on NQ (58,880 queries) against our model trained on 58,880 generated queries for the two generalizability settings; out-of-distribution and out-of-domain. Here, the generated query dataset contains mostly passages with multiple queries per passage.

3.5.1 Out-of-distribution generalizability

Table 3.3 shows the top 100 accuracy scores obtained by the baseline DPR model (trained on NQ) and our DPR model, trained on the 58k generated query dataset with multiple queries per passage (*58k generated*), on the out-of-distribution datasets. We also include the scores on the NQ dataset itself for completeness, but it should be noted that this dataset is an in-distribution dataset for the baseline model.

The model trained on *58k generated* (our model) outperforms the baseline DPR model on 5 out of 6 out-of-distribution datasets, with the Curated TREC dataset being the sole exception. However, the difference in accuracy between the two models on Curated TREC and WebQ are not statistically significant. Our model generalizes better in all four datasets (out of six) where the difference is statistically significant. The baseline DPR model does better on the NQ test dataset (in-distribution) compared to the our model trained on generated queries (out-of-distribution).

Interestingly, the baseline DPR model trails our model trained on *58k generated* even on the queries generated from the NQ passages despite being trained on fairly

Table 3.3: Top 100 accuracy scores for the model trained on *58k generated* and the baseline DPR model trained on NQ for out-of-distribution datasets. The highest score is in **bold** and ‡ indicates in-domain performance. Statistical significance with paired t-test: * indicates $p < 0.05$ and ** indicates $p < 0.01$.

	Standard datasets					Generated datasets	
Model	NQ	TriviaQA	TREC	WebQ	SQuAD	NQ dev	Wikipedia
Baseline DPR	**84.9**‡**	78.7	**90.7**	77.6	63.5	81.5	56.7
58k generated (ours)	75.0	**80.0****	89.6	**78.3**	**69.4****	**85.3****	**79.2****

Table 3.4: MRR@100 scores for the model trained on *58k generated* and the baseline DPR model trained on NQ for out-of-distribution datasets. Same notational conventions as in Table 3.3.

	Standard datasets					Generated datasets	
Model	NQ	TriviaQA	TREC	WebQ	SQuAD	NQ dev	Wikipedia
Baseline DPR	**0.512**‡**	**0.437****	**0.583****	**0.389****	0.234	**0.449****	0.240
58k generated (ours)	0.313	0.426	0.507	0.358	**0.258****	0.426	**0.415****

similar data. This indicates that the performance of DPR models trained on data with mostly a single query from each passage deteriorates rapidly when tested on new queries. This observation may be explained by our initial hypothesis. If a model trained on data with a single query per passage learns to discard information, it is logical that the model would struggle when dealing with multiple queries from a passage as this requires the context encoder to encode all information available in the passage in order to correctly match all the queries from that passage. These results indicate that training a model on data with multiple queries per passage results in improved generalizability in the out-of-distribution setting.

The baseline model outperforms the model trained on *58k generated* on 4 out of 6 out-of-distribution datasets when considering MRR@100 scores (Table 3.4). However, the *58k generated* model performs slightly better on average.

3.5.2 Out-of-domain generalizability

Table 3.5 shows the recall@100 scores obtained by the baseline DPR model (trained on NQ) and our DPR model trained on *58k generated*. The model trained on *58k generated* outperforms the baseline DPR model achieving higher recall@100 scores in 12 out of 13 out-of-domain datasets. Considering only the statistically significant results ($p < 0.05$), our model trained on multiple query per passage data outperforms the baseline DPR model on all 10 out of 10 datasets.

The MRR@100 scores (Table 3.5) follow a similar pattern, with the model trained on *58k generated* outperforming the baseline in 9 out of 10 out-of-domain datasets where the results are statistically significant.

The model trained with data containing multiple queries per passage (our model

Table 3.5: Recall@100 and MRR@100 scores for the baseline DPR model trained on NQ and the model trained on 58k generated queries for out-of-domain datasets. Same notational conventions as in Table 3.3.

	Recall@100		MRR@100	
Dataset	Baseline DPR	58k generated	Baseline DPR	58k generated
ArguAna	0.480	**0.919****	0.051	**0.213****
Climate FEVER	**0.410**	0.405	**0.258****	0.220
CQA Dup Stack	0.109	**0.139****	0.041	**0.068****
DBPedia	0.310	**0.335***	0.559	**0.564**
FEVER	0.748	**0.805****	**0.497**	0.492
FiQa	0.313	**0.369****	0.131	**0.195****
HotpotQA	0.493	**0.502**	0.419	**0.559****
NFCorpus	0.170	**0.238**	0.306	**0.377****
Quora	0.566	**0.880****	0.279	**0.590****
SciDocs	0.196	**0.253****	0.136	**0.207****
SciFact	0.581	**0.704****	0.247	**0.372****
Touche	0.276	**0.344****	0.234	**0.386****
TREC-COVID	0.096	**0.177****	0.287	**0.354**

trained on *58k generated*) dominates the baseline DPR model, trained on mostly single query per passage data, in both the out-of-distribution and out-of-domain setting. This clearly superior zero-shot generalization performance when a DPR model is trained on data with multiple queries per passage answers the research question (RQ 2) demonstrating that training a DPR model on data with multiple queries per passage does result in a better generalized model.

3.6 Analysis

3.6.1 Generation versus data composition

We conduct a further analysis to confirm that the improvements in generalizability shown in Section 3.5 is due to the composition of the dataset, specifically the number of queries per passage, rather than any artifact of the query generation process. Here, we compare the generalizability to out-of-distribution and out-of-domain data of two models trained on generated queries. The first model is trained on generated queries with multiple queries per passage (same as in Section 3.5) and the second model is trained on generated queries with only a single query from each passage.

Table 3.6 shows the top-100 accuracy scores obtained by the two models on the out-of-distribution datasets. The model trained on *58k generated (multi)* outperforms the model trained *58k generated (single)* on 5 out of 7 datasets (one loss and one tie). Four of these results are statistically significant with the model trained on *58k generated (multi)* generalizing better in all four cases. Similarly, the model trained on *58k generated (multi)* outperforms the model trained on *58k generated (single)*, in terms

Table 3.6: Top 100 accuracy scores for the models trained on *58k generated (single)* and *58k generated (multi)* for out-of-distribution datasets. Same notational conventions as in Table 3.3.

	Standard datasets					Generated datasets	
Model	NQ	TriviaQA	TREC	WebQ	SQuAD	NQ dev	Wikipedia
58k generated (single)	**75.0**	78.4	**90.2**	77.5	67.9	81.9	74.5
58k generated (multi)	**75.0**	**80.0****	89.6	**78.3**	**69.4****	**85.3****	**79.2****

Table 3.7: MRR@100 scores for the models trained on *58k generated (single)* and *58k generated (multi)* for out-of-distribution datasets. Same notational conventions as in Table 3.3.

	Standard datasets					Generated datasets	
Model	NQ	TriviaQA	TREC	WebQ	SQuAD	NQ dev	Wikipedia
58k generated (single)	0.309	0.397	0.489	0.350	0.247	0.394	0.366
58k generated (multi)	**0.313**	**0.426****	**0.507**	**0.358**	**0.258****	**0.426****	**0.415****

of MRR@100 scores (Table 3.7), on all six out-of-distribution datasets with four of the results being statistically significant. These results clearly show that having multiple queries per passage in the training data helps the model generalize better to out-of-distribution queries, as the only difference between the two models is the composition of the training data.

Table 3.8 shows the recall@100 scores obtained by the two models on the out-of-domain datasets. Again, the model trained with multiple queries per passage outperforms the model trained on single query per passage data and generalizes better to 10 out of 13 out-of-domain datasets. Looking at the statistically significant results, the model trained on *58k generated (multi)* does better on 6 out of 7 datasets. The results on the remaining six datasets are likely not statistically significant as they contain a very small number of queries.

Overall, the model trained on *58k generated (multi)* generalizes better, in both out-of-distribution and out-of-domain settings, compared to the model trained on *58k generated (single)* when all other factors are kept constant. This confirms that the composition of training data, specifically the number of queries per passage, is an important factor to consider when training dense retrieval models and that training on data with multiple queries per passage leads to a model that is capable of generalizing better to out-of-distribution and out-of-domain queries.

3.6.2 Effect of dataset size

We also investigate the effect of the total number of generated queries in a training dataset on the generalizability of DPR models. For this analysis we compare three DPR models trained on three generated query datasets, where each dataset contains 58,880 (*58k generated*), 236,444 (*236k generated*), and 786,312 (*786k generated*) queries

Table 3.8: Recall@100 and MRR@100 scores for the models trained on *58k generated (single)* and *58k generated (multi)* for out-of-domain datasets. Same notational conventions as in Table 3.3.

Dataset	Recall@100		MRR@100	
	58k generated (single)	58k generated (multi)	58k generated (single)	58k generated (multi)
ArguAna	0.885	**0.919****	0.208	**0.213**
Climate FEVER	0.378	**0.405****	0.188	**0.220****
CQA Dup Stack	0.134	**0.139****	**0.068**	**0.068**
DBPedia	0.312	**0.335****	0.545	**0.564**
FEVER	0.722	**0.805****	0.415	**0.492****
FiQa	0.358	**0.369**	0.189	**0.195**
HotpotQA	0.430	**0.502****	0.460	**0.559****
NFCorpus	0.185	**0.238**	0.376	**0.377**
Quora	**0.909****	0.880	**0.658****	0.590
SciDocs	0.246	**0.253**	0.202	**0.207**
SciFact	0.685	**0.704**	0.346	**0.372**
Touche	**0.371**	0.344	0.343	**0.386**
TREC-COVID	**0.181**	0.177	0.300	**0.354**

Table 3.9: Top 100 accuracy scores for the models trained on the three generated query datasets *58k*, *236k*, and *786k* for out-of-distribution datasets. Same notational conventions as in Table 3.3.

Model	Standard Datasets					Generated Datasets	
	NQ	TriviaQA	TREC	WQ	SQuAD	NQ dev	Wikipedia
58k Generated	75.0	80.0	89.6	78.3	69.4	85.3	79.2
236k Generated	79.5	82.5	91.7	80.6	71.6	90.1	85.4
786k Generated	**80.5***	**83.2****	**92.2**	**80.7**	**72.9****	**92.4****	**89.4****

respectively. Note that all three of these datasets contain data with multiple queries per passage. Again, we report zero-shot scores in both the out-of-distribution and out-of-domain settings.

Table 3.9 shows the top-100 accuracy scores obtained by each model on the out-of-distribution datasets. The model trained on *786k generated* generalizes better to all seven datasets, with five of the results being statistically significant. In terms of MRR@100 (Table 3.10), the model trained on *786k generated* obtains higher scores on 5 out of 6 datasets, with four being statistically significant. These results indicate that training on larger datasets, containing data with multiple queries per passage, does yield better results on out-of-distribution datasets in a zero-shot setting.

Table 3.11 shows the recall@100 scores obtained by each model on the out-of-domain datasets. Overall, the model trained on the largest dataset, *786k generated*, does marginally better than the other two models, obtaining the highest recall@100 score for

Table 3.10: MRR@100 scores for the models trained on the three generated query datasets *58k*, *236k*, and *786k* for out-of-distribution datasets. Same notational conventions as in Table 3.3.

	Standard Datasets					Generated Datasets	
Model	NQ	TriviaQA	TREC	WQ	SQuAD	NQ dev	Wikipedia
58k Generated	0.313	0.426	0.507	0.358	0.258	0.426	0.415
236k Generated	0.339	0.467	0.515	**0.381**	0.274	0.493	0.488
786k Generated	**0.360****	**0.492****	**0.526**	0.379	**0.283****	**0.522****	**0.542****

Table 3.11: Recall@100 and MRR@100 scores for the model trained on the three generated query datasets *58k generated*, *236k generated*, and *786k generated* for the out-of-domain datasets. Same notational conventions as in Table 3.3.

	Recall@100			MRR@100		
Dataset	58k *generated*	236k *generated*	786k *generated*	58k *generated*	236k *generated*	786k *generated*
ArguAna	0.919	0.939	**0.940**	**0.213**	0.209	0.202
Climate FEVER	0.405	**0.406**	0.371	0.220	**0.224**	0.198
CQA Dup Stack	0.139	**0.154**	0.153	0.068	**0.072****	0.069
DBPedia	0.335	0.362	**0.364**	**0.564**	**0.564**	**0.564**
FEVER	0.805	0.853	**0.856**	0.492	**0.508****	0.476
FiQa	0.369	**0.385**	0.377	**0.195**	0.190	0.171
HotpotQA	0.502	0.557	**0.572****	0.559	0.598	**0.603**
NFCorpus	**0.238**	0.216	0.216	0.377	**0.387**	0.382
Quora	0.880	0.897	**0.929****	0.590	0.613	**0.636****
SciDocs	0.253	0.253	**0.261**	0.207	**0.212**	0.198
SciFact	0.704	0.737	**0.790**	0.372	0.373	**0.374**
Touche	0.344	**0.366**	0.325	**0.386**	0.325	0.314
TREC-COVID	**0.177****	0.124	0.119	**0.354***	0.219	0.166

seven out of thirteen out-of-domain datasets. The other two models, trained on *236k generated* and *58k generated*, achieve the highest scores in four out of thirteen and two out of thirteen, respectively. Only three of these results are statistically significant with the model trained on *786k generated* doing better on two and the model trained on *58k generated* performing better on the other. The MRR@100 scores (Table 3.11) are even more mixed, with the model trained on *236k genrated* performing better in 2 out of 4 statistically significant results while the other two models perform better on one each.

While larger training datasets help with zero-shot performance on out-of-distribution datasets, the benefit of more generated data is less clear with regard to zero-shot performance on out-of-domain datasets. Although the model trained on *786k generated* generalizes better than the other two models, the increase in recall scores are marginal, especially compared to the increased cost of training which increases linearly with dataset size. Overall, training DPR models on more generated queries with multiple

queries per passage can improve the generalizability of the model, but with sharply diminishing gains. This is likely due to the fact that increasing the size of the training dataset does not necessarily increase the diversity of the training data.

3.7 Conclusion and Future Work

We have shown that the generalizability of dense passage retrievers may suffer from learning to discard information from passages during training. This problem can be mitigated by using training data containing a sufficient number of passages with multiple associated queries. By exposing the dense retriever to multiple facets of information contained in the same passage, we ensure that the model does not learn to discard potentially useful information, leading to improved retrieval accuracy for out-of-domain topics and queries and a better-generalized model overall. This answers the research question (**RQ 2**) posed in this chapter and concludes that training on data with multiple queries per passage improves the generalizability of dense passage retriever models.

As a general lesson, when training a dense retrieval model, it is important to consider the number of queries per passage, or more generally, how much of the information contained in a given passage is covered by the queries. Training datasets with a large number of queries per passage can be automatically generated for training dense retrievers resulting in a better generalized model.

As to limitations, we did not use hard negative mining [131] or late interaction [51], which are known to improve the generalizability of dense retrievers. We leave their integration to future work but note that our method is trivially compatible with such techniques and is also independent of the actual dense retriever architecture that is used.

Finally, it would be interesting to use our proposed dataset generation method on a full collection of Wikipedia passages to train a DPR model. While our analysis of the effect of dataset size (Section 3.6.2) did not demonstrate meaningful gains in generalizability, a sufficiently large query collection (a generated query dataset of the full Wikipedia collection would be several orders of magnitude larger) containing diverse topics may generalize very well to most domains.

Part II

ROBUST REFUSAL AND EVIDENCE-BASED ANSWERING

4

Reward Shaping for Robust Refusal in Small Language Models for Retrieval-Augmented Question Answering

4.1 Introduction

Retrieval-augmented generation (RAG) is an effective technique to extend the knowledge of language models (LMs) beyond what was learned during pre- and post-training [16, 35, 57]. This is most useful in enterprise settings or in domains where information is continuously updated, making it infeasible to keep an LM's knowledge updated through training [16, 45]. In high-stakes domains, it is critical to ensure that *any* response from a LM is truthful [6, 43]. It is often better to conservatively refuse to answer a question rather than attempt to guess and risk hallucinating a plausible-sounding, but incorrect answer [127, 133]. In such scenarios, we want a LM deployed as part of a RAG system to answer questions based on, and *only based on*, the documents retrieved by the retrieval component.

We focus on the behavior of the LM component of a RAG system and analyze the ability of open-source LMs to (a) refuse to answer queries when sufficient evidence is not present in the provided information, (b) accurately answer complex queries given sufficient information (in one or more documents), and (c) robustly answer complex queries in a noisy retrieval scenario (e.g., in real-world RAG systems) where both relevant and non-relevant documents are present.

Small LMs. While large-scale proprietary models can often be prompted to behave reliably, small open-source LMs remain highly relevant [1, 3, 141]. They are efficient to train and deploy, making them practical in domains where computational budgets, privacy requirements, or regulatory constraints preclude the use of frontier models [122, 125, 134]. Small models are easier to steer with explicit reward shaping, providing a tractable testbed for studying how refusal behavior can be trained rather than emerged

This chapter was published as T. C. Rajapakse and M. de Rijke. Reward shaping for robust refusal in small language models for retrieval-augmented question answering. In *Under Review*, 2026.

implicitly [64, 124]. Small LMs are not just a pragmatic choice but a valuable target for building more trustworthy retrieval-augmented systems.

Observed behavior. Our analysis of current instruction-tuned LMs reveals three consistent patterns: (a) in refusal settings where relevant evidence is withheld, models often attempt to answer instead of refusing, despite explicit prompts; (b) even in vanilla conditions with gold documents provided, performance remains moderate; and (c) when distractor documents are added, accuracy drops significantly in most cases, with models failing to identify relevant evidence. General-purpose instruction-tuning and prompting alone do not suffice for reliable refusal and robustness to real-world, noisy retrieval scenarios.

Our approach. We introduce reward shaping for robust refusal (RSRR), a reinforcement learning framework that explicitly trains models to both ground their answers in evidence and to refuse to answer when evidence is insufficient. We build on proximal policy optimization (PPO) [60, 75, 148], widely used in RLHF, and extend it with reward components tailored for retrieval-augmented QA [16, 35, 57, 78]. Our design combines (a) a *relevance-based reward* that encourages models to generate reasoning more relevant than any single input document, (b) an *answer correctness reward* that schedules a penalty/bonus depending on whether the gold answer (including `NO-RES`) is present, (c) a *formatting reward* that enforces structured outputs, and (d) a KL penalty to stabilize training.

The primary research question posed in this chapter is the following: (**RQ 3**) How can relevance-based rewards be used in reward shaping to train small language models to answer based on retrieved evidence and to refuse when evidence is insufficient?

To answer **RQ 3**, we explore the following sub-questions in this chapter:

RQ 3.1 Can existing small, instruction-tuned LMs answer questions based on retrieved evidence and refuse to answer when evidence is insufficient?

RQ 3.2 Can RSRR improve the refusal accuracy of small LMs, i.e., refuse to answer when evidence is insufficient?

RQ 3.3 Can RSRR improve the robustness of small LMs to distractor documents?

Contributions. Our contributions are threefold:

(1) We evaluate small instruction-tuned LMs under vanilla, refusal, and distractor settings, showing that they struggle with calibrated refusal and robustness.

(2) We introduce RSRR, a reward shaping framework that couples evidence-grounded generation with explicit refusal training.

(3) We demonstrate that RSRR substantially improves refusal accuracy (+43.3% relative improvement) and robustness to distractor documents (+39.8% relative improvement) across multiple QA benchmarks, making small LMs more reliable components in retrieval-augmented systems.

4.2 Related Work

Recent research has paid a lot of attention to reducing hallucinations and guiding refusal behavior of LLMs, especially in RAG systems. We highlight related work that focuses on improving LLM accuracy and refusal behavior.

Sensitive domains. Hallucination mitigation and calibrated refusal is especially important in high-stakes domains such as healthcare. Pandit et al. [83] introduce MedHallu, a benchmark dataset for detecting medical hallucinations where the task is to predict whether an answer is hallucinated or grounded. They show that even state-of-the-art LMs struggle to detect medical hallucinations, with the best model achieving an F1 score of 0.625. Lee et al. [56] develop a RAG-based system to enhance the reliability of diabetes management advise provided by LMs and show that using the retrieval sources significantly improves the reliability of the provided answers. These works underscore the prevalence of hallucination, especially in the healthcare domain, and how RAG systems can be used to reduce such hallucinations. They focus on hallucination detection or reduction via better retrieval.

In contrast, we examine the behavior of LMs *after* the retrieval step. We evaluate the question answering accuracy of LMs under both ideal and noisy conditions, as well as the ability to *refuse* when sufficient evidence is absent, complementing retrieval-centric approaches with mechanisms for conservative abstention.

Refusal. Some research efforts move beyond retrieval and RAG and consider directly aligning LM refusal behavior through reward models and reinforcement learning. Xu et al. [133] propose reinforcement learning from knowledge feedback, rewarding correct answers and refusals, and penalizing hallucinations. This approach teaches models to abstain when outside their knowledge scope, thereby reducing overconfident mistakes. Similarly, Mu et al. [74] introduce rule-based rewards to refine refusal behavior, using explicit safety rules to guide reinforcement learning, resulting in safer and more consistent outputs. Other work focuses on balancing helpfulness and harmlessness, and shows that naively optimizing for refusal can lead to over-refusal, thereby hurting usefulness [109]. These works highlight the importance of calibrated refusal and introduce reward mechanisms to encourage it, but they focus on not exceeding a model's knowledge boundary.

In contrast, we focus on answering or refusing based on the evidence provided to the model instead of relying on internal knowledge acquired during training an LM. We create evaluation settings where refusal is the correct outcome and shape rewards to teach models to abstain when evidence is missing, while rewarding faithful reasoning when sufficient evidence is present. This joint emphasis on reasoning *and* refusal is crucial for reliable RAG systems.

Reward shaping and alignment. Another line of research focuses on shaping reward functions to improve factuality and alignment in LMs. Zhang et al. [139] introduce train reward models on large preference datasets annotated along axes such as hallucination, comprehensiveness, and attribution. These reward models are then used to fine-tune LMs with reinforcement learning, yielding improved generation quality and reduced hallucinations in retrieval-augmented settings. Other works emphasize alignment strate-

gies beyond factuality. E.g., Mu et al. [74] propose rule-based rewards to control safety and refusal behavior.

While these works demonstrate the effectiveness of shaping reward functions to improve reliability, they target broad aspects of alignment such as factuality, style, or safety. In contrast, we introduce reward shaping that directly couples *reasoning over retrieved evidence* with the ability to *refuse when evidence is insufficient*. By explicitly rewarding multi-document reasoning alongside conservative abstention, we complement prior alignment efforts and address a key gap in retrieval-augmented generation: ensuring that models do not only generate answers based on, *and only based on*, provided evidence, but also recognize when no answer can be supported by the available evidence.

4.3 Analysis of Instruction-Tuned Models

We analyze instruction-tuned models to investigate their ability to answer questions based on evidence and their ability to refuse to answer when sufficient evidence is not provided. We consider three settings, using the augmentation methods described in Section 4.3.3. As our focus is on the QA part of the RAG pipeline, we ensure that the gold documents (required to answer the query) are always present in the prompt, except in the refusal setting (Section 4.3.3), to prevent introducing noise from an imperfect retriever to the results of our analysis.

In the *refusal* setting, we investigate the ability of instruction-tuned models to refuse to answer when evidence is missing.

In the *vanilla* setting, we look at the ability of instruction-tuned models to answer complex queries that may require reasoning. This also serves as a baseline to assess any drop in performance in a RAG scenario where non-relevant documents (distractors) are likely to be presented alongside relevant ones.

In the setting *with distractors*, we include distractor documents by sampling from a ranked list of documents to obtain pseudo-relevant documents. This setting is analogous to a RAG setup where an imperfect retriever is unlikely to retrieve relevant, *and only relevant*, documents for a given query.

4.3.1 Models

We test a variety of instruction-tuned, open-source language models, namely, Llama 3, Gemma 2, Qwen 2.5, and DeepSeek-R1. For this analysis, we use the instruction-tuned versions of the models without any additional training.

4.3.2 Datasets

We analyze the performance of existing instruction-tuned models on four datasets (statistics summarized in Table 4.2):

PubmedQA. PubmedQA is a biomedical question answering dataset that contains 1k expert-annotated, 61.2k unlabeled, and 211.3k artificially generated QA instances. We use the official test set from [47] comprising 500 question and context pairs, where

reasoning over the context is required to answer (yes/no/maybe) the question. We use the accompanying context of each question to simulate a RAG question answering scenario.

BioASQ. BioASQ is a biomedical question answering dataset, designed to reflect the real information needs of biomedical experts [53]. The dataset contains four types of questions; binary yes/no, factoid, list, and summarization. In this work, we focus on the binary yes/no and factoid questions to enable exact match evaluation. The dataset also provides relevant snippets and documents for each question and we use these supporting texts to simulate a RAG question answering scenario.

BeerQA. BeerQA [85] is a multi-hop question answering and retrieval dataset where reasoning over one or more Wikipedia passages is required to answer the queries. In this work, we focus on the reasoning and question answering aspect of the task.

StrategyQA. StrategyQA [32] is a question answering benchmark requiring multi-hop reasoning where the reasoning steps are implicit in the question and should be inferred by the model. Again, we use the questions and evidence paragraphs to construct a RAG style question answering scenario.

4.3.3 Augmentation

Next to the *vanilla* datasets (where all queries are correctly answerable from the evidence paragraphs), we create two settings by augmenting the data. Section 4.5.2 details the process of building the *with distractors* and *refusal* settings.

Refusal (No-Res). To test a model's ability to refuse to answer a query without sufficient evidence, we replace the original evidence documents with pseudo-relevant documents by sampling from ranked lists of candidate documents.

With distractors. A real-world RAG question answering scenario would contain relevant and non-relevant (or pseudo-relevant) passages given the imperfect nature of any retrieval system. We investigate the question answering performance of models when given a mix of relevant and non-relevant documents to the query. That is, sufficient evidence is given to answer the query, but additional *distractor* documents are also present.

Summary. Across all datasets, we evaluate models in three distinct settings: *vanilla* (fully answerable queries), *refusal* (queries without sufficient evidence), and *with distractors* (queries with both relevant and irrelevant documents). These experiments are conducted on four diverse datasets to assess performance under varying evidence conditions.

4.3.4 Evaluation

We evaluate models across three retrieval-augmented QA scenarios: *No-Res* (gold evidence removed; the correct output is `NO-RES`), *Vanilla* (gold evidence present), and *Distractor* (gold evidence plus additional irrelevant documents).

Metric. We report **tagged accuracy**: an answer is considered correct if it is produced inside the required `<ANSWER>...</ANSWER>` tag and matches one of the gold

Table 4.1: Tagged accuracy of instruction-tuned models in vanilla, distractor, and No-Res scenarios (mean across prompt lengths).

Setting	Model	BioASQ	BeerQA	PubmedQA	StrategyQA	Average	Wins
No-Res	R1-7B	0.308	0.663	0.408	*0.699*	0.520	0
	Gemma-2-2B	0.479	**0.800**	0.562	0.472	0.578	1
	LLaMA-3.2-3B	*0.484*	*0.682*	*0.723*	0.617	0.627	0
	Qwen-2.5-3B	**0.683**	0.662	**0.921**	**0.927**	0.749	3
Vanilla	R1-7B	0.402	*0.588*	**0.485**	**0.460**	0.484	2
	Gemma-2-2B	**0.477**	0.474	0.340	0.138	0.357	1
	LLaMA-3.2-3B	0.321	**0.596**	*0.464*	0.351	0.433	1
	Qwen-2.5-3B	*0.402*	0.453	0.392	*0.358*	0.401	0
Distractors	R1-7B	0.329	*0.497*	**0.390**	**0.458**	0.419	2
	Gemma-2-2B	**0.482**	0.491	0.233	0.287	0.373	1
	LLaMA-3.2-3B	0.323	**0.542**	*0.366*	0.368	0.400	1
	Qwen-2.5-3B	*0.383*	0.355	0.235	*0.373*	0.336	0

answers after normalization. Normalization removes case distinctions, punctuation, and articles, and applies standard whitespace trimming. This metric emphasizes whether the model can produce a *grounded, verifiable answer* when one exists, and whether it can refuse to answer (following [24], we prompt the models to respond with `NO-RES`) when evidence is absent.

We report the average across three different prompts (Appendix 4.A) to reflect consistent performance. Some examples of the input and output to the models are shown in Appendix 4.B.

4.3.5 Results of the analysis

We observe several patterns across the evaluation settings in Table 4.1. Qwen-2.5-3B demonstrates the strongest refusal behavior, particularly on PubMedQA and StrategyQA, while other models frequently hallucinate instead of refusing. Even with explicit prompting, all models fail a significant fraction of cases, showing that instruction-tuning and prompting alone is insufficient for calibrated refusal.

In the vanilla setting, instruction-tuned models achieve only moderate performance. While LLaMA-3.2-3B performs best *on average* and R1-7B records the higher *number of wins*, no single model consistently dominates across all datasets, and overall accuracy remains limited. Hence, models at these parameter scales, are not reliable when required to integrate information across multiple documents, even when explicitly prompted.

When distractor documents are introduced, all models (with the exception of Gemma-2-2B) experience a noticeable drop in performance compared to the vanilla setting. This suggests that current instruction-tuned models are highly sensitive to non-relevant information and often fail to robustly identify and reason over the relevant evidence.

Overall, the results indicate three consistent weaknesses in current instruction-tuned

small LMs: unreliable refusal behavior when evidence is absent, limited reasoning ability even under ideal conditions, and reduced accuracy in noisy retrieval settings. These results indicate that the answer to the first research question of this chapter (**RQ 3.1**) is that small LMs *cannot* reliably answer questions based on retrieved evidence and refuse to answer when evidence is insufficient. This motivates the need for explicit reward shaping approaches such as ours, which teach models to both reason faithfully over provided evidence and refuse to answer when evidence is insufficient.

4.4 Reward Shaping for Reasoning and Refusal (RSRR)

4.4.1 Proximal policy optimization (PPO)

We adopt Proximal Policy Optimization (PPO) as the reinforcement learning algorithm for fine-tuning instruction-tuned language models with our proposed reward signals. PPO is widely used in RLHF pipelines as it provides a stable and sample-efficient method for policy optimization while preventing large, destabilizing updates. Formally, given a policy π_θ parameterized by θ, PPO maximizes the clipped surrogate objective:

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}_t\Big[\min\big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big], \tag{4.1}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between the updated and old policies, and $\hat{A}_t$ is the estimated advantage at timestep t. The clipping parameter ϵ prevents excessively large policy updates. In our setup:

- **Policy (π_θ):** the instruction-tuned LLM generating step-by-step reasoning and final answers.
- **Environment:** a query together with retrieved documents.
- **Actions:** token-level generations, terminated by end-of-sequence.
- **Rewards:** shaped signals described in Sections 4.4.2 to 4.4.6 (relevance-based reasoning, answer correctness, formatting penalty, and KL regularization).

Following standard PPO-RLHF practice, we add a scalar value head to the model to estimate state values for advantage computation. A KL penalty is applied to ensure that the updated policy remains close to the reference model, preventing catastrophic drift during training.

4.4.2 Relevance-based reward

The central component of our reward design is a relevance-based signal that encourages model reasoning to be grounded in the retrieved evidence. For a query q, the model output y is parsed to obtain a reasoning span $r = \text{extract}(y, \texttt{REASONING})$. Let $\mathcal{D} = \{d_i\}_{i=1}^{K}$ be the set of documents provided to the model. We use a learned ranking

model $Rel(q, \cdot)$ to score the relevance of the reasoning span and the documents to the query. We define the relevance reward as

$$R_{\text{rel}}(q, y, \mathcal{D}) \;=\; Rel(q, r) \;-\; \max_{d \in \mathcal{D}} Rel(q, d), \tag{4.2}$$

which is positive only when the model's reasoning is more relevant to the query than any individual document. This reward promotes reasoning that faithfully integrates information from multiple documents and resists distractors. We scale this term by λ_{rel} in the final objective.

4.4.3 Alignment reward

We include an alignment reward R_{align} from a separate reward model trained to capture general quality dimensions such as fluency and coherence. This reward complements the relevance signal by ensuring that outputs are both grounded in the retrieved evidence and well-formed. Formally,

$$R_{\text{align}}(q, y) \;=\; \lambda_{\text{align}}\, g(q, y), \tag{4.3}$$

where g is the alignment reward model and λ_{align} is a weighting coefficient.

4.4.4 Answer correctness

To encourage inclusion of the correct answer, we add a scheduled shaping term. For each example, let $a^\star$ denote the gold answer or `NO-RES` if the query is unanswerable, and define $\text{correct}(y, a^\star) = 1$ if the extracted `ANSWER` span from y matches $a^\star$ under task normalization, else 0. The reward is

$$R_{\text{correct}}(y, a^\star, t) \;=\; \begin{cases} +\phi(t), & \text{correct}(y, a^\star) = 1, \\ -\phi(t), & \text{correct}(y, a^\star) = 0, \end{cases} \tag{4.4}$$

where $\phi(t) \geq 0$ grows over steps t: small early (encouraging exploration), larger later (enforcing correctness).

4.4.5 Formatting reward

Outputs must follow the required template with `<REASONING>` and `<ANSWER>` tags. Let $\text{valid}(y) \in \{0, 1\}$ indicate conformity. With a fixed $\gamma_{\text{fmt}} > 0$, we assign

$$R_{\text{fmt}}(y) = \begin{cases} +\gamma_{\text{fmt}}, & \text{valid}(y) = 1, \\ -\gamma_{\text{fmt}}, & \text{valid}(y) = 0. \end{cases} \tag{4.5}$$

4.4.6 KL

We regularize π_θ against a reference π_{ref} with a per-token KL term:

$$r_t^{\text{KL}} = -\beta\, \text{KL}\big(\pi_\theta(\cdot|s_t) \;\big|\; \pi_{\text{ref}}(\cdot|s_t)\big), \tag{4.6}$$

where β controls penalty strength and prevents drift from the initialization.

Table 4.2: Dataset statistics for evaluation.

Dataset	Test size	Answer type
BioASQ	2,688	yes/no, span
PubMedQA	500	yes/no/maybe
BeerQA	14,121	span
StrategyQA	229	yes/no

4.4.7 Final reward

The overall reward combines all sequence-level terms with KL regularization. At the sequence level we define

$$R_{\text{seq}} = \lambda_{\text{rel}} R_{\text{rel}} + \lambda_{\text{align}} R_{\text{align}} + R_{\text{correct}} + R_{\text{fmt}}, \tag{4.7}$$

where λ_{rel} and λ_{align} weight the relevance and alignment components, and R_{correct}, R_{fmt} are scaled by their hyperparameters $\phi(t)$ and γ_{fmt}. Following standard RLHF practice, R_{seq} is added at the final token while per-token KL penalties r_t^{KL} are applied throughout, yielding

$$\mathbf{r} = (r_1^{\text{KL}},\ \ldots,\ r_{T-1}^{\text{KL}},\ r_T^{\text{KL}} + R_{\text{seq}}). \tag{4.8}$$

This vector is used for advantage estimation with GAE and PPO optimization.

4.5 Experimental Setup for Reward Shaping

4.5.1 Datasets

We use the BeerQA [85] dataset for training as it consists of multi-hop questions that require evidence from multiple documents to answer correctly. We augment the training set of BeerQA to add examples containing distractor documents along with the relevant documents, and we also generate unanswerable versions of each query by replacing all relevant documents with distractor documents. The augmentation process is detailed in Section 4.5.2.

We evaluate on four QA datasets spanning biomedical and multi-hop reasoning: PubMedQA [47], BioASQ [53], BeerQA [85], and StrategyQA [32]. We cast each example as retrieval-augmented QA with evidence documents $\mathcal{D}$. We follow the same evaluation procedure as in Section 4.3 and evaluate the models under *refusal*, *vanilla*, and *with distractors* settings.

Splits. We use the official test or held-out evaluation splits when available and reserve a subset of the remaining data for training. See Table 4.2 for the counts.

4.5.2 Data augmentation

We augment each dataset to create three settings:

(1) **NO-RES:** gold evidence is withheld; the correct target is `NO-RES`.

(2) **Vanilla:** all gold evidence is present; queries are answerable.

(3) **Distractors:** gold evidence is present *plus* k non-relevant documents sampled from a retrieval ranking (details below).

Distractor sampling. We sample k distractors per query from a ranked candidate list (excluding gold), using top-M as the pool, such that each example has *n* associated documents. The ranked candidate lists are built using BM25. To fit within the model's context length, we set $n \in [3, 20]$ and $M \in [10, 40]$, with $k = n -$ (number of relevant documents). Both n and M are held constant per dataset.

Refusals. In the *refusal* setting, all gold documents are replaced with sampled distractor documents.

4.5.3 Prompt and output template

All models are prompted with a fixed instruction and the evidence block (`Documents:`), followed by `Question:` and a required output template enforced via the formatting reward (Section 4.4.5):

```
<REASONING> ... </REASONING>
<ANSWER> ... </ANSWER>
```

4.5.4 Models

Policy and reference. We fine-tune an instruction-tuned backbone as the policy π_θ and use the same initialization as the reference π_{ref} for KL control (Section 4.4.6). We report results for ***LLaMA-3.2-3B***.

Alignment reward model. $g(q, y)$ is a sequence-level model scoring general quality (fluency, coherence, helpfulness, etc.). We use the publicly available model checkpoint *Skywork/Skywork-Reward-Llama-3.1-8B-v0.2* [66].

Relevance scorer. $Rel(q, \cdot)$ is a ranking model used to score (q, r) and (q, d_i) pairs for the relevance reward (Section 4.4.2). We use the publicly available model checkpoint *rankllama-v1-7b* [69].

Trained models. We use the publicly available, instruction-tuned LLaMA-3.2-3B as the baseline model for these experiments. We further train this model on the augmented version of BeerQA (4.5.1) using the PPO algorithm. The PPO - No Rank model is trained with the standard PPO setup without a relevance-based reward and reward shaping for robust refusal is trained using the proposed RSRR method which adds a relevance-based reward to the standard PPO setup (Section 4.4). Both models are trained using identical hyperparameters specified in Section 4.5.5.

4.5.5 Hyperparameters

All models were trained for one epoch on 8×NVIDIA L40 GPUs (48GB), corresponding to ∼2,090 PPO updates. We adopt the Hugging Face `trl` defaults for PPO unless

otherwise noted. The policy π_θ was optimized with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01, $\epsilon = 10^{-8}$) at a learning rate of 3×10^{-6}. We run two PPO epochs with a single mini-batch split, for an effective batch size of 128, and set a maximum generation length of 500 tokens.

For sequence-level shaping, we use $\lambda_{\text{rel}} = 1.0$ for R_{rel} and $\lambda_{\text{align}} = 0.8$ for R_{align}. The formatting reward R_{fmt} uses $\gamma_{\text{fmt}} = 1.0$, giving a symmetric bonus/penalty for valid vs. invalid outputs. The correctness reward R_{correct} is scheduled linearly from 3.0 to 7.0 between steps 100 and 800 ($\phi(t)$), encouraging exploration early and enforcing refusal later. A penalty of -2.0 is applied if the output does not end with an end-of-sequence marker. We retain the `trl` default clipping range of 0.2 for both policy and value updates.

4.6 Results for Reward Shaping

Overview. We compare the instruction-tuned baseline Llama model, PPO without the relevance-based reward (PPO - No Rank), and the full RSRR method across three settings: *Vanilla*, *Distractors*, and *No-Res*. Performance is measured using tagged accuracy (Table 4.3) and percentage of incorrect refusals (Table 4.4), capturing how often the model abstains despite sufficient evidence.

Tagged accuracy. Table 4.3 reports tagged accuracy scores. In the *No-Res* setting, PPO - No Rank achieves the highest scores, but this is largely due to an over-refusal strategy: the model learns to abstain excessively, inflating refusal accuracy but undermining performance on answerable questions. RSRR performs slightly lower in raw accuracy but achieves more calibrated refusal.

In the *Vanilla* setting, both PPO variants perform comparably to the baseline. RSRR improves performance on BioASQ and BeerQA, but shows drops on StrategyQA. Since StrategyQA often requires implicit multi-step reasoning whose evidence is not always directly expressed in the retrieved documents, RSRR's conservative bias can suppress otherwise valid answers.

In the *Distractor* setting, RSRR consistently outperforms PPO - No Rank and the baseline. By rewarding reasoning that is more relevant to the query than any single distractor document, RSRR achieves better robustness to noisy retrieval, maintaining higher accuracy even in the presence of irrelevant evidence.

Incorrect refusals. To better diagnose errors beyond tagged accuracy, we report the percentage of *incorrect refusals*, i.e. cases where the model abstains despite sufficient evidence being present (Table 4.4). PPO - No Rank suffers from severe over-refusal, particularly on PubMedQA and StrategyQA, where it abstains in 40–90% of answerable cases. reward shaping for robust refusal substantially reduces this failure mode, though it still exhibits higher incorrect refusal rates than the baseline. The baseline, in contrast, minimizes refusal errors but, as Table 4.3 shows, this comes at the cost of weaker robustness in distractor settings.

Summary. Overall, reward shaping for robust refusal improves robustness in distractor settings and produces more calibrated refusals than PPO - No Rank, with some cost to raw accuracy. We return to the implications of this trade-off in Section 4.7.

Table 4.3: Tagged accuracy of PPO-trained models under different QA scenarios (mean across prompt lengths).

Setting	Model	BioASQ	BeerQA	PubmedQA	StrategyQA	Average	Wins
No-Res	LLaMA-3.2-3B	0.483	0.683	*0.723*	0.617	0.627	0
	PPO - No Rank	**0.834**	**0.981**	**0.975**	**0.994**	0.946	4
	RSRR	*0.699*	*0.958*	*0.962*	*0.993*	0.903	0
Vanilla	LLaMA-3.2-3B	0.321	0.596	*0.464*	**0.351**	0.431	1
	PPO - No Rank	*0.497*	*0.634*	**0.477**	0.332	0.485	1
	RSRR	**0.610**	**0.753**	0.463	*0.351*	0.544	3
Distractors	LLaMA-3.2-3B	0.323	*0.542*	*0.366*	**0.368**	0.400	1
	PPO - No Rank	*0.483*	0.498	0.489	0.295	0.441	0
	RSRR	**0.611**	**0.678**	**0.507**	*0.345*	0.535	3

Table 4.4: Percentage of incorrect refusals. Lower values are better. Reported across datasets for the Vanilla and Distractor settings.

Setting	Model	BioASQ	BeerQA	PubMedQA	StrategyQA	Average
Vanilla	LLaMA-3.2-3B	**12.31**	**7.48**	**10.80**	**9.17**	9.94
	PPO - No Rank	27.38	11.35	40.60	64.19	35.88
	RSRR	*17.96*	*8.76*	*31.60*	*54.58*	28.22
Distractors	LLaMA-3.2-3B	**11.60**	**9.87**	29.20	**17.24**	16.98
	PPO - No Rank	30.09	28.70	36.80	92.13	46.93
	RSRR	*15.47*	*16.24*	**27.80**	*85.58*	36.27

4.7 Discussion

We have seen that small language models can be explicitly trained to base their answers on retrieved evidence and to abstain when the available evidence is insufficient. This is achieved through relevance-based reward shaping, which evaluates the reasoning process against retrieved documents, combined with scheduled correctness and format rewards. The method substantially improves refusal in the No-Res setting and robustness in the presence of distractors, moving beyond mere accuracy improvements toward more reliable system behavior.

These results provide an answer to the second and third research questions posed in this chapter, demonstrating that training a small LM with RSRR substantially improves its refusal accuracy (**RQ 3.2**) and its robustness to distractor documents (**RQ 3.3**).

There is a trade-off between accuracy and conservative behavior. While reward shaping for robust refusal improves calibrated refusal and distractor robustness, it sometimes lowers raw tagged accuracy, most clearly on StrategyQA when going from the vanilla to the distractor setting. This dataset requires implicit reasoning steps whose supporting evidence is not always explicitly surfaced in the retrieved documents. In such cases, reward shaping for robust refusal's conservative bias toward refusal can suppress

otherwise valid yes/no answers. We view this trade-off as intentional: in high-stakes domains, avoiding unsupported answers may be preferable to maximizing raw accuracy. Nonetheless, balancing caution with coverage remains an important challenge for future work.

4.8 Conclusion

We have introduced RSRR, a reward shaping framework for retrieval-augmented QA that explicitly trains models to both ground their answers in evidence and refuse when evidence is missing. By applying a relevance-margin reward on the reasoning process, combined with scheduled correctness and format shaping, RSRR improves refusal accuracy and robustness to distractors. Experiments show that small language models, when trained with RSRR, can move beyond surface-level accuracy toward more calibrated and trustworthy behavior. This, taken together with the answers to **RQ 3.1**, **3.2**, and **3.3**, resolves the primary research question posed in this chapter (**RQ 3**) and demonstrates that small LMs can be trained to answer based on retrieved evidence and refuse to answer when evidence is insufficient using RSRR.

Reliable refusal is a critical property for deploying language models in high-stakes domains such as healthcare and law, where overconfident but incorrect outputs can lead to harmful consequences. By improving refusal, our approach reduces the risk of misinformation and helps foster user trust. At the same time, explicit refusal shaping introduces its own risks: poorly calibrated penalties may lead models to refuse too often, withholding useful information. We therefore view this work as a step toward more trustworthy RAG systems, while emphasizing the need for careful calibration and evaluation in practice.

Our approach has some limitations. First, it relies on a structured output format with explicit tags and a designated `NO-RES` string. This makes evaluation straightforward, but may not fully capture more open-ended or conversational use cases. Second, the quality of the relevance and alignment rewards is bounded by the underlying reward models: if these models misjudge relevance, the shaped policy may learn undesired behaviors. Third, our evaluation is limited to QA datasets with clear gold answers. Real-world information-seeking often involves multiple valid answers, partially supported claims, or inherently uncertain evidence. Finally, we focus on tagged accuracy, leaving the reasoning spans unevaluated; this avoids noisy judgments, but leaves open whether improvements in reasoning quality extend beyond surface-level answer correctness.

One direction for future research is to extend refusal training to tasks beyond QA, such as dialogue, summarization, or code generation, where abstention is equally important. Another is to integrate uncertainty estimation methods (e.g., entropy or variance-based signals) with reward shaping to provide a richer training signal for calibrated refusal. Scaling the approach and comparing against prompt-only baselines would clarify whether reward shaping provides complementary benefits at scale. Finally, exploring alternative output formats that are less brittle than fixed tags could make refusal behavior more robust in less constrained settings.

4.A Prompts

We used three prompts (of varying length) tailored to each dataset to evaluate the models and reported the mean value. The prompts are shown below.

4.A.1 BioASQ

Prompt A.

> Answer the question using only the given documents. If the answer cannot be inferred, respond with NO-RES.
> Format your response as follows:
> <REASONING> Step-by-step reasoning </REASONING>
> <ANSWER> Your final answer / NO-RES </ANSWER>

Prompt B.

> You are given a question and a set of documents. Your task is to reason through the information step by step and answer the question based only on what is in the documents.
> Some documents may be irrelevant. If you cannot find enough evidence to answer, respond with NO-RES.
> Follow these steps:
> 1. Understand the question and identify any supporting facts needed. 2. Review all documents to find relevant entities and facts. 3. Connect information across documents using multi-hop reasoning. 4. Derive a final answer that is grounded in the documents (max 10 tokens). 5. If the documents do not support an answer, respond with NO-RES.
> Your response should follow this format, with no extra text:
> <REASONING>Step-by-step reasoning</REASONING>
> <ANSWER> Your final answer / NO-RES </ANSWER>

Prompt C.

> You are given: - A question - A set of documents
> Your task: 1. Parse the question; note entities and facts required to answer it. 2. Read all documents; ignore irrelevant content and avoid outside knowledge. 3. Extract relevant entities/facts and note supports/contradictions. 4. Connect information across documents with multi-hop reasoning to infer the answer. 5. Decide the final answer strictly grounded in the documents. Keep the answer concise ($\leq$ 10 tokens). If support is missing or conflicting without a clear resolution, output NO-RES.
> Your output must be in the following format with no extra commentary:
> <REASONING> Step-by-step reasoning </REASONING>
> <ANSWER> Your final answer / NO-RES </ANSWER>

4.A.2 BeerQA

Prompt A.

Answer the question using only the given documents. If the answer cannot be inferred, respond with NO-RES.
Format your response as follows:
<REASONING> Step-by-step reasoning </REASONING>
<ANSWER> Your final answer / NO-RES </ANSWER>

Prompt B.

You are given a question and supporting documents. Your task is to answer the question based only on the information in the documents.
Some documents may be irrelevant. If you cannot infer the answer from the provided documents, respond with NO-RES.
Follow these steps:
1. Understand the question and identify any supporting facts needed. 2. Review all documents to find relevant facts and entities. 3. Connect information using multi-hop reasoning. 4. Derive the final answer, grounded in the documents (max 10 tokens). 5. If the documents do not support an answer, respond with NO-RES.
Your response must follow this format, with no extra text:
<REASONING> Step-by-step reasoning </REASONING>
<ANSWER> Your final answer / NO-RES </ANSWER>

Prompt C.

You are given: - A question - A set of supporting documents
Your task: 1. Read the question carefully; note the facts and entities needed to answer. 2. Examine all documents, ignoring irrelevant ones and avoiding outside knowledge. 3. Identify relevant facts and entities. 4. Connect them across documents using multi-hop reasoning. 5. Produce a concise final answer (leq 10 tokens) strictly grounded in the documents. 6. If the documents do not contain enough information, respond with NO-RES.
Your output must be in the following format with no extra commentary:
<REASONING> Step-by-step reasoning </REASONING>
<ANSWER> Your final answer / NO-RES </ANSWER>

4.A.3 PubMedQA

Prompt A.

Answer the question with yes, no, maybe, or NO-RES, using only the given documents.
Format your response as follows:
<REASONING> Step-by-step reasoning </REASONING>
<ANSWER> yes / no / maybe / NO-RES </ANSWER>

Prompt B.

You are given a question and a set of documents. Your task is to answer the question with either yes, no, maybe, or NO-RES, based only on the information in the documents.
Some documents may be irrelevant. If you cannot infer the answer from the provided documents, respond with NO-RES.
Follow these steps:
1. Understand the question and identify any supporting facts needed. 2. Review all documents to find relevant facts and entities. 3. Connect information using multi-hop reasoning. 4. If the documents support an answer, conclude with yes, no, or maybe (if the information in the documents is inconclusive). 5. If the documents do not support an answer, respond with NO-RES.
Your response must follow this format, with no extra text:
<REASONING>Step-by-step reasoning</REASONING>
<ANSWER> yes / no / maybe / NO-RES </ANSWER>

Prompt C.

You are given: - A question - A set of documents
Your task: Decide if the answer is ”yes”, ”no”, ”maybe”, or ”NO-RES” (if the answer cannot be inferred), using ONLY the provided documents.
Detailed steps: 1. Read the question carefully to understand the entities and facts required. 2. Examine all provided documents, ignoring irrelevant ones. 3. Identify direct or indirect evidence supporting or contradicting the answer. 4. If needed, combine information from multiple documents using multi-hop reasoning. 5. Decide: - ”yes” if the documents clearly confirm the statement - ”no” if they clearly refute it - ”maybe” if evidence is present but inconclusive - ”NO-RES” if no relevant evidence exists in the documents 6. Do not use external knowledge. 7. Your output must be in the following format with no extra commentary:
<REASONING> Step-by-step reasoning </REASONING>
<ANSWER> yes / no / maybe / NO-RES </ANSWER>

4.A.4 StrategyQA

Prompt A.

Answer the question using only the given documents. If the answer cannot be inferred, respond with NO-RES.
Format your response as follows:
<REASONING> Step-by-step reasoning </REASONING>
<ANSWER> true / false / NO-RES </ANSWER>

Prompt B.

You are given a question and a set of documents. Your task is to answer the question with either yes, no, maybe, or NO-RES, based only on the information in the documents.
Some documents may be irrelevant. If you cannot infer the answer from the provided documents, respond with NO-RES.
Follow these steps:
1. Understand the question and identify any supporting facts needed. 2. Review all documents to find relevant facts and entities. 3. Connect information using multi-hop reasoning. 4. If the documents support an answer, conclude with true or false. 5. If the documents do not support an answer, respond with NO-RES.
Your response must follow this format, with no extra text:
<REASONING> Step-by-step reasoning </REASONING>
<ANSWER> true / false / NO-RES </ANSWER>

Prompt C.

You are given: - A question - A set of documents
Your task: 1. Parse the question; identify entities and supporting facts needed. 2. Read all documents; ignore irrelevant content and avoid outside knowledge. 3. Extract relevant entities/facts and note supports/contradictions. 4. Connect information across documents using multi-hop reasoning. 5. Conclude with true or false, strictly grounded in the documents. If evidence is insufficient, output NO-RES.
Your output must be in the following format with no extra commentary:
<REASONING> Step-by-step reasoning </REASONING>
<ANSWER> true / false / NO-RES </ANSWER>

4.B Input and Output Examples

In this appendix we show a few examples of the input and output formats.

Example 1. Correct response from RSRR.

Prompt
You are given a question and supporting documents. Your task is to answer the question based only on the information in the documents. Some documents may be irrelevant. If you cannot infer the answer from the provided documents, respond

with NO-RES.

Follow these steps: 1. Understand the question and identify any supporting facts needed. 2. Review all documents to find relevant facts and entities. 3. Connect information using multi-hop reasoning. 4. Derive the final answer, grounded in the documents (max 10 tokens). 5. If the documents do not support an answer, respond with NO-RES.

Your response must follow this format, with no extra text: `<REASONING>` Step-by-step reasoning `</REASONING>` `<ANSWER>` Your final answer / NO-RES `</ANSWER>`

Documents
[0] (Title: *Edwin Thomas Smith*) Sir Edwin Thomas Smith (6 April 1830 – 25 December 1919) was an English-born South Australian brewer, businessman, councillor, mayor, politician and benefactor.
[1] (Title: *South Australian Brewing Company*) Provisional directors were Sir E. T. Smith, W. K. Simms, MLC., W. B. Rounsevell, MP., R. A. Stock, Alfred Simms, Charles H. T. Hart, and Frank Rymill of Adelaide, and Hon. N. Fitzgerald, MLC., John Robb, JP., M. D. McEacharn, John B. Watson, and John McIlwraith, of Melbourne.
[2] (Title: *South Australian Brewing Company*) In 1927 T. A. Nation was the brewer and G. B. Bryant the general manager. His board of directors comprised S. J. Jacobs (later Managing Director), Sir Lancelot Stirling, K.C.M.G., Edward Fitzgerald, LL.D., and H. W. Morphett.

Model Output
`<REASONING>` Based on the provided documents, the English-born South Australian brewer whose brewery was amalgamated with W. K. Simms's West End Brewery was Sir Edwin Thomas Smith. According to Document [0], he was born on 6 April 1830.`</REASONING>`
`<ANSWER>` 6 April 1830 `</ANSWER>`

Gold: 6 April 1830

Example 2. Incorrect response from PPO - No Rank.

Prompt
You are given a question and supporting documents. Your task is to answer the question based only on the information in the documents. Some documents may be irrelevant. If you cannot infer the answer from the provided documents, respond with NO-RES.

Follow these steps: 1. Understand the question and identify any supporting facts needed. 2. Review all documents to find relevant facts and entities. 3. Connect information using multi-hop reasoning. 4. Derive the final answer, grounded in the documents (max 10 tokens). 5. If the documents do not support an answer, respond with NO-RES.

Your response must follow this format, with no extra text: `<REASONING>` Step-by-step reasoning `</REASONING>` `<ANSWER>` Your final answer / NO-RES `</ANSWER>`

Documents
[0] (Title: *Edwin Thomas Smith*) Sir Edwin Thomas Smith (6 April 1830 – 25 December 1919) was an English-born South Australian brewer, businessman, councillor, mayor, politician and benefactor.
[1] (Title: *South Australian Brewing Company*) Provisional directors were Sir E. T. Smith, W. K. Simms, MLC., W. B. Rounsevell, MP., R. A. Stock, Alfred Simms, Charles H. T. Hart, and Frank Rymill of Adelaide, and Hon. N. Fitzgerald, MLC., John Robb, JP., M. D. McEacharn, John B. Watson, and John McIlwraith, of Melbourne.
[2] (Title: *South Australian Brewing Company*) In 1927 T. A. Nation was the brewer and G. B. Bryant the general manager. His board of directors comprised S. J. Jacobs (later Managing Director), Sir Lancelot Stirling, K.C.M.G., Edward Fitzgerald, LL.D., and H. W. Morphett.

Question
When was the English-born South Australian brewer whose brewery was amalgamated with W. K. Simms's West End Brewery born?

Model Output
There is no information in the provided documents about the English-born South Australian brewer whose brewery was amalgamated with W. K. Simms's West End Brewery. The documents only mention Sir Edwin Thomas Smith, the provisional directors of the South Australian Brewing Company, and the board of directors in 1927. Therefore, the answer cannot be derived from the provided documents. `<ANSWER>` NO-RES `</ANSWER>`

Gold: 6 April 1830

Example 2. Incorrect response from *LLaMA-3.2-3B*.

Prompt
You are given a question and supporting documents. Your task is to answer the question based only on the information in the documents. Some documents may be

irrelevant. If you cannot infer the answer from the provided documents, respond with NO-RES.

Follow these steps: 1. Understand the question and identify any supporting facts needed. 2. Review all documents to find relevant facts and entities. 3. Connect information using multi-hop reasoning. 4. Derive the final answer, grounded in the documents (max 10 tokens). 5. If the documents do not support an answer, respond with NO-RES.

Your response must follow this format, with no extra text: `<REASONING>` Step-by-step reasoning `</REASONING>` `<ANSWER>` Your final answer / NO-RES `</ANSWER>`

Documents
[0] (Title: *Edwin Thomas Smith*) Sir Edwin Thomas Smith (6 April 1830 – 25 December 1919) was an English-born South Australian brewer, businessman, councillor, mayor, politician and benefactor.
[1] (Title: *South Australian Brewing Company*) Provisional directors were Sir E. T. Smith, W. K. Simms, MLC., W. B. Rounsevell, MP., R. A. Stock, Alfred Simms, Charles H. T. Hart, and Frank Rymill of Adelaide, and Hon. N. Fitzgerald, MLC., John Robb, JP., M. D. McEacharn, John B. Watson, and John McIlwraith, of Melbourne.
[2] (Title: *South Australian Brewing Company*) In 1927 T. A. Nation was the brewer and G. B. Bryant the general manager. His board of directors comprised S. J. Jacobs (later Managing Director), Sir Lancelot Stirling, K.C.M.G., Edward Fitzgerald, LL.D., and H. W. Morphett.

Question
When was the English-born South Australian brewer whose brewery was amalgamated with W. K. Simms's West End Brewery born?

Model Output
`<REASONING>` Step-by-step reasoning `</REASONING>`
`<ANSWER>` 6 April 1830 `</ANSWER>`

Gold: 6 April 1830

Part III

ACCESSIBILITY AND SHARED PRACTICE

5 Simple Transformers

5.1 Introduction

Many of us see the goal of research as the improvement of people's lives everywhere. As researchers, we have the opportunity to make new discoveries and create innovations that help advances in dealing with challenges in climate change, education, healthcare, media, mobility, culture, government, and business [102] – language technology in general, and information retrieval in particular, is likely to impact all of these areas. In an open society, technology with such a wide and pervasive reach should be in the hands of many, not of a few [106]. For us, this point of view entails three things: (i) a focus on shared innovation, where the knowledge and technology being created and examined is shared and owned by multiple, and multiple types of, stakeholders – academic, industrial, governmental, and societal [73]; (ii) a focus on empowering people, equipping them with new tools, rather than replacing them [2, 103]; and (iii) a focus on getting many voices around the table, as the technology is likely to affect many [13, 105].

In our view, open-source software is key in each of those three dimensions. Clearly, open-source software feeds and is fed by shared innovation ecosystems, thereby fostering collaboration [104]. By fostering accessibility, open-source software empowers people, not just by code, data, and tool sharing but also by reducing the need for repeated large-scale investments [36]. Finally, open-source software drives transparency, which helps to build trust, which in turn helps to increase engagement.

None of the points that we have made so far is new. The open-source community has made impressive, valuable contributions towards advancing the goal of shared innovation (i). For example, in language technology, the BLOOM initiative [129] organized by the BigScience-community produced the first multilingual large language model (LLM) trained in complete transparency as a result of the largest collaboration of AI researchers ever involved in a single research project [129]. And in information retrieval, Lucene, the open-source search engine software library, has been in active usage for over 25 years [33].

In this chapter, we draw attention to the second focus we outlined: empowering

This chapter was published as T. C. Rajapakse, A. Yates, and M. de Rijke. Simple Transformers: Open-source for all. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 209–215, 2024.

people by equipping them with new (information retrieval and language technology) tools, not just specialists and people who focus on language technology. We hope that by providing people with accessible tools to experiment with and use information retrieval and language technology, we can also advance on the third focus, i.e., getting many voices around the table to discuss and make decisions about information retrieval language technology that will impact the lives of many.

This chapter describes one instance of this idea: **Simple Transformers**. Transformers [115] and transformer-based architectures have found usage across, and far beyond, information retrieval and language technology. Simple Transformers is an open-source library for training, evaluating, and using transformer models. The core design philosophy for the library is that using these transformative technologies should not be restricted to only those with expertise in the field. Instead, the library is designed to be accessible to and used by as broad a community as possible.

Simple Transformers has a special focus on information retrieval (IR) techniques and other use cases of transformer models. It provides an easy interface to train, test, and use neural IR methods such as dense retrieval. For example, building a dense vector index of a document collection (using Faiss [28, 48]) is encapsulated to reduce complexity for the user. We demonstrate how to train a dual-encoder dense retrieval model [50] in Section 5.4.3.

The remainder of this chapter focuses on the Simple Transformers library. We introduce the library in Section 5.3, elaborate on the design, and provide code examples in Section 5.4. In Section 5.6, we look at instances where Simple Transformers has been used outside the core information retrieval and language technology communities, demonstrating our contributions towards empowering people with new language technology tools. Finally, we discuss some limitations and our commitments for future work and reflect on our ambition of "open-source for all" in Section 5.7.

5.2 Related Work

The IR community has a long tradition of foundational open-source. In addition to Apache Lucene (mentioned in the introduction), prominent examples include the Terrier IR platform that implements state-of-the-art indexing and retrieval functionalities, with a focus on the rapid development and evaluation of large-scale retrieval applications [81]. Furthermore, Pyserini [61], Tevatron [30], the LSR library [77], and Sentence Transformers [94] are among popular libraries supporting the training and evaluation of IR methods and models.

These libraries are valuable tools widely used in the IR community. Our goal with Simple Transformers, however, is to reach a broader audience. As we describe in Section 5.6, and as evidenced by the popularity of the Hugging Face suite of libraries [34, 58, 128], NLP methods have been adopted in many fields beyond language technology. By integrating IR techniques alongside NLP methods with similar design principles and a familiar interface (as described in Section 5.4), we aim to make IR methods and techniques more accessible to users who may not possess expert knowledge of IR. While the Huggingface libraries support the *architectures* of the models used in IR, they do not provide methods of directly training or evaluating dense retrieval models, which

can be a significant barrier to users unfamiliar with the (often complex) pipelines used in IR tasks. On the other hand, Simple Transformers provide accessible functions to directly implement IR pipelines as we discuss in Section 5.4.

5.3 Simple Transformers

Transformer models have become the most ubiquitous neural network architecture for natural language processing, eclipsing previous architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [128]. The typical process for using a transformer model on a new task involves identifying a suitable pre-trained model (e.g., BERT, RoBERTa, BART), preprocessing the training data and converting the text into input tokens that can be processed by a transformer (tokenization), training the transformer on new data starting from the pre-trained weights (finetuning), and finally making predictions on similarly preprocessed test data.

While transformers can be trained to perform exceedingly well on many IR and NLP tasks, the finetuning process can be complicated depending on the specific models used and the task at hand. The Simple Transformers library was created to simplify this process and offer a streamlined and straightforward way to train, finetune, and perform predictions with many different transformer models on various IR and NLP tasks. Building on top of the Hugging Face Transformers library [128], thereby enabling access to the vast collection of Transformer models available on the Hugging Face Hub,[1] Simple Transformers provides an interface that is both easy to use and easy to understand (see Section 5.4.3 for examples).

As of 2024, the Simple Transformers library has accumulated over 4,000 stars on GitHub and has been downloaded over 3 million times, with an average of nearly 50,000 downloads per month according to PePy. Additionally, over 1,500 other GitHub repositories use the Simple Transformers library. These statistics bear testament to the usability and popularity of Simple Transformers, both within and without the core IR and NLP communities.

Simple Transformers is primarily designed for ease of use and accessibility for users who may not be familiar with IR, NLP or Deep Learning in general. We believe that this focus makes Simple Transformers a valuable resource for communities outside IR and NLP, such as researchers from other scientific disciplines and industry practitioners with broader software development skill sets, e.g., software engineers who are not specialized in IR and NLP.

The fields of IR and NLP encompass various tasks such as similarity search, text classification, question answering, and language generation. All these tasks and more are supported in the Simple Transformers library (see Table 5.1), with each task mapped to its own class for convenience. While the task-specific classes follow the same pipeline of training, evaluation, and prediction, there are necessary differences between different classes (e.g., data formats). The following section elaborates on this pipeline approach.

[1] `https://huggingface.co/models`

Table 5.1: Overview of tasks and corresponding model classes in the Simple Transformers library.

Task	Model Class
Information retrieval (dense retrieval)	`RetrievalModel`
(Large) language models (training, fine-tuning, and generation)	`LanguageModelingModel`
Encoder model training and fine-tuning	`LanguageModelingModel`
Sequence classification	`ClassificationModel`
Token classification (NER)	`NERModel`
Question answering	`QuestionAnsweringModel`
Language generation	`LanguageGenerationModel`
Sequence to sequence (incl. Mono-T5 [80])	`T5Model`, `Seq2SeqModel`
Text representation	`RepresentationModel`

5.4 Design and Implementation

5.4.1 Setup

Simple Transformers is available as a Python package on PyPi[2] and can be installed through *pip*.

5.4.2 Design

Data. Simple Transformers defines input and output data formats according to each task. For example, the RetrievalModel class expects input data to contain the columns *query_text*, *gold_passage*, and an optional *title* column, while the LanguageModelingModel class expects the input data to contain a single column *text*. Data may be passed as a path to a file containing the required columns or as an in-memory Pandas dataframe. The specific input and output data formats for each task is defined in the documentation[3] of the Simple Transformers library.

Configuring models. Each Simple Transformers class associated with a task has a configuration class (named after the task class, e.g., ClassificationArgs) that controls all hyperparameters and configuration options for that task. Crucially, the configuration class for each task comes with reasonable defaults that a novice user (a user unfamiliar with the task and/or model) can rely on to provide good results even without hyperparameter tuning. On the other hand, a more experienced user can perform hyperparameter tuning or otherwise change the configuration to improve the final performance of the model. Again, the documentation of the library details the configuration options available for each task.

Training models. Each Simple Transformers model class (task) has a train_model () method that accepts a dataset (in the correct input format) and initiates the train-

[2] `https://pypi.org/project/simpletransformers/`
[3] `https://simpletransformers.ai/docs/installation/`

ing loop for the model. The hyperparameters for training can be configured when initializing the model class or alternatively can be passed to the train_model() function itself to update the initial configuration. Simple Transformers also has a built-in evaluate_during_training option to facilitate model training progress tracking by performing evaluation (optionally on a separate validation set) at set intervals as well as out-of-the-box early stopping functionality.

Evaluating models. Similar to training, a model can be evaluated by calling the eval_model() function that accepts an evaluation dataset (in the correct input format). This method will compute a set of default metrics chosen to fit a particular task (e.g., classification models report *accuracy*, *F1*, etc., while retrieval models report *nDCG*, *recall*, etc.), but a user may also pass any additional metric functions that they wish to compute in addition to the default metrics. The eval_model() function also gets called internally when using the evaluate_during_training feature.

Predicting with models. The predict() function of each model is called to make predictions on input data passed to the function. Unlike the eval_model() function, the predict() function does not compute metrics (and therefore does not require labels) and instead outputs the predictions from the model.

Visualization. The Simple Transformers library integrates two open-source visualization libraries, Weights & Biases[4] and Tensorboard.[5] These integrations can be used to track training and evaluation metrics easily. For example, simply setting the wandb_project option to a non-null value will automatically track training and evaluation metrics using the Weights & Biases library.

5.4.3 Examples

This section demonstrates two minimal examples of training and using transformer models with the Simple Transformers library. Example code for other models can be found in the Simple Transformers GitHub repository, and minimal start examples can be found in the documentation. Advanced configuration and techniques, such as custom parameter groups and hyperparameter tuning, are described in the documentation (and omitted from the thesis for brevity).

Dense retrieval

This example shows how to train a simple dense retrieval model like the dense passage retriever (DPR) [50] on the MSMARCO dataset [76]. The full code can be found in the Simple Transformers repository.[6]

Data format. The format for the training data can be seen in Table 5.2. Note that the hard_negative column is optional, but using hard negatives typically yields better performance [131]. When the column is not present, in-batch negatives are used.

[4] https://github.com/wandb/wandb

[5] https://github.com/tensorflow/tensorboard

[6] https://github.com/ThilinaRajapakse/simpletransformers/tree/master/examples/retrieval

Table 5.2: Data format to train a dense retrieval model.

query_text	**gold_passage**	**hard_negative**
what are the liberal arts?	liberal arts. 1. the academic course of instruction at a college intended to...	Liberal Education: An approach to college learning that empowers individuals...
what is the mechanism of action of fibrinolytic or thrombolytic drugs?	Baillière's Clinical Haematology. 6 Mechanism of action of the...	Be able to diagram the coagulation and fibrinolytic pathways and the...

Training. To train a dense retrieval model with Simple Transformers, the RetrievalArgs configuration class is instantiated and the preferred hyperparameter values are set.

```
# Path to the training data
train_data_path = "msmarco-train.tsv"

model_args = RetrievalArgs()
model_args.use_hard_negatives = True
model_args.num_train_epochs = 40
model_args.train_batch_size = 16
model_args.learning_rate = 1e-6
```

Then, we define the pretrained model to be used for initializing the weights and initialize the RetrievalModel with the pretrained weights and the configured hyperparameter values.

```
model_type = "custom"
model_name = None
context_name = "bert-base-cased"
question_name = "bert-base-cased"

# Instantiate RetrievalModel
model = RetrievalModel(
    model_type,
    model_name,
    context_name,
    question_name,
    args=model_args,
)
```

The train_model() function is called as shown below to initiate the training of the dense retriever.

```
model.train_model(train_data)
```

Evaluation. Simple Transformers can be used to perform evaluation on datasets in TREC or BEIR [111] formats, as well as Pandas dataframes containing query_text and gold_passage columns. Simple Transformers automatically builds and stores a dense vector index, using Faiss, to perform retrieval.

When save_as_experiment is enabled, Simple Transformers will save per query metrics to facilitate easy statistical testing. Furthermore, the built-in analyze_experiment () function can be used to automatically perform statistical tests and generate latex table code for results saved with save_as_experiment.

```
model.eval_model(
    eval_data_path,
    save_as_experiment=True,
    pytrec_eval_metrics=[
        "recip_rank",
        "recall_100",
        "ndcg_cut_10",
    ],
)
```

Adapter tuning a LLaMA model for retrieval augmented generation

This example demonstrates how to easily adapter tune [41] a LLaMA 7B[7] model [113] for retrieval augmented generation [57, 93]. Simple Transformers uses the Hugging Face PEFT [72], bitsandbytes, and accelerate [34] libraries to enable adapter tuning a LLaMA 7B model on consumer-grade GPUs (this example requires a GPU with ∼12 GB of memory). Again, the full code can be found in the Simple Transformers repository.[8]

We use the pretrained DRAGON dense retriever model [63] and the MSMARCO collection as the dense retriever and document collection for retrieval, respectively.

Data format. The data format for adapter tuning a large language model (LLM) is shown in Table 5.3. The text column contains the text that the language model will be (adapter) trained on, while the rag_query column contains the questions that will be passed to the retriever model. Note that the data format is identical for training or finetuning a language model without retrieval augmentation, except that the rag_query column is not required in this scenario.

Training. Following the same procedure as in Section 5.4.3, we instantiate the LanguageModelingArgs object and set the preferred hyperparameter values.

```
# Path to the training data
train_data_path = "train.jsonl"

model_args = LanguageModelingArgs()
model_args.peft = True
```

[7] https://huggingface.co/meta-llama/Llama-2-7b-hf

[8] https://github.com/ThilinaRajapakse/simpletransformers/tree/master/examples/llms

Table 5.3: Data format to adapter tune a retrieval-augmented large language model (LLM).

text	**rag_query**
Question: To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France? Answer: Saint Bernadette Soubirous	To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?
Question: What is in front of the Notre Dame Main Building? Answer: a copper statue of Christ	What is in front of the Notre Dame Main Building?

```
model_args.nf4 = True
model_args.loftq_bits = 4
model_args.lora_config = {"r": 8}
model_args.data_format = "jsonl"
model_args.optimizer = "Adam8bit"
model_args.max_seq_length = 500
```

Then, we define the LLM to adapt and the retriever we will use. Note that retrieval_model here has been instantiated similarly to the example in Section 5.4.3, and we omit its instantiation for brevity.

```
model_type = "causal"
model_name = "meta-llama/Llama-2-7b-hf"

model = LanguageModelingModel(
    "causal",
    "meta-llama/Llama-2-7b-hf",
    args=model_args,
    retrieval_model=retrieval_model,
)
```

Adapter tuning of the LLM is initiated by calling the train_model() function, again identical to the procedure followed to train the dense retrieval model earlier.

```
model.train_model(train_data)
```

Language generation. The predict() function of the model class can be used to generate the LLM responses to an input, as shown below.

```
responses, _ = model.predict(
    to_predict,
    rag_queries=rag_queries,
)
```

Here, to_predict is a list of input prompts to the LLM, and the rag_queries is a list of queries used by the retriever to retrieve documents relevant to the question from the

document collection. In practice, rag_queries are essentially the question itself without the prompt for the LLM (i.e., "*What is in front of the Notre Dame Main Building*" without the surrounding "*Question: ... Answer:*" found in the LLM prompt).

5.5 Experiments

In this section, we replicate the results from DPR [50] by training and evaluating a DPR dense retrieval model[9] [10] using the Simple Transformers library. Following [50], we train the model on the Natural Questions (NQ) [54] dataset and evaluate on NQ, TriviaQA [49], WebQuestions (WQ) [10], CuratedTREC (TREC) [9], and SQuAD [91]. Following Karpukhin et al. [50], we train the model using a learning rate of $1e^{-5}$, and a batch size of 128 for 40 epochs, without early stopping.

Table 5.4: Top-20/Top-100 accuracy on the five evaluation datasets used in DPR [50].

Framework	NQ	TriviaQA	WQ	TREC	SQuAD
DPR [50]	78.4/85.4	79.4/85.0	73.2/81.4	79.8/89.1	63.2/77.2
Tevatron	79.8/86.9	80.2/85.5	75.4/82.9	84.0/90.7	62.3/77.0
ST	76.9/85.6	79.5/85.2	72.2/80.7	84.2/91.6	61.9/77.1

Table 5.4 shows the top-20 and top-100 accuracy metrics obtained by DPR models trained using the original DPR repository [50], the Tevatron library [30], and the Simple Transformers library. Overall, the top-*k* metrics are comparable for all three frameworks, indicating that the results were replicated successfully.

5.6 Adoption

This section looks at the adoption of Simple Transformers in research areas other than information retrieval and NLP. We believe this to be an important benefit of open-source libraries and technologies.

Materials science. Cruse et al. [23] use transformer models via the Simple Transformers library to build a publicly available dataset of codified gold nanoparticle synthesis protocols and outcomes extracted from nearly 5 million existing nanoparticle materials science publications. They suggest that this data could help understand the underlying mechanisms controlling the size and shape of gold nanoparticles.

Environmental social sciences. Coan et al. [21] study the role of misinformation in shaping the debate on climate change. They used Simple Transformers to build a model to detect specific contrarian claims regarding climate change, as opposed to broad topics or themes [21]. They specifically note that interdisciplinary approaches are required to combat online misinformation efforts at the required scale. We believe that this highlights the need for accessible open-source tools that could be employed by researchers less familiar with IR and NLP tools and methods.

[9] `https://huggingface.co/thilina/dpr-nq-st-query-encoder`
[10] `https://huggingface.co/thilina/dpr-nq-st-context-encoder`

Healthcare. Owen et al. [82] observe that key symptoms of major depressive disorder can manifest in the structure of written language and that early diagnosis is critical for achieving accurate diagnosis and improving patient outcomes. Therefore, they find that automatic analysis of rich and regular written communications, such as social media or web forums, may provide opportunities for early intervention. They employ various transformer models, through Simple Transformers, for the purpose of text analysis to detect symptoms of major depressive disorder.

Economics. Trust et al. [114] study the potential of NLP and machine learning methods to understand how decision-making in society is influenced by news. They propose a transformer-based method using weak supervision for the identification of news articles about economic uncertainty and adapt to the calculation of the economic policy uncertainty (EPU) index [8] to their proposed strategy. The transformer-based method proposed by Trust et al. [114] yields significant improvement over the existing keyword search based method. Simple Transformers was used to train the transformer models.

Human resources. Robson et al. [96] explore the use of AI in bridging the gap between workforce reskilling demands and the offerings of local college training programs. Their project, SkillSync™, uses AI technologies, including a variation of the Siamese Multidepth Transformer-based Hierarchical Encoder (SMITH) [135] and other natural language understanding methods, to facilitate the alignment of job descriptions and course information with skills taxonomies.

5.7 Limitations, Future Work, and Reflections

5.7.1 Limitations

The primary limitation of the Simple Transformers library is that we do not focus on the deployment of transformer models once they are trained and tested. While this is less of a concern for researchers, industry users need to deploy, scale, and maintain transformer models in production. The `predict()` function of Simple Transformers models can be used to perform ad hoc predictions but integrating this with, e.g., web servers is beyond the scope of the library. However, any model trained with Simple Transformers is fully compatible with any tools that work with Hugging Face models.

Furthermore, Simple Transformers does not support the full range of models available on Hugging Face, but instead focuses on the most commonly used tasks such as information retrieval, classification, and language generation. These tasks are also commonly found in other fields of research, and we believe that the ease of use for these core tasks is crucial to democratize access to powerful IR and NLP capabilities without overwhelming users with the complexity required to manage and deploy a wider array of models. By streamlining the process for these key IR and NLP tasks, the Simple Transformers library aims to lower the barrier to entry, enabling more researchers and developers to use the power of transformer models. We believe that both general-purpose, large-scale tools like the Hugging Face suite of libraries and more specialized, beginner-friendly tools such as Simple Transformers play a crucial role in driving shared innovation. These tools empower a diverse community to benefit from open-source language technology, and their continued popularity highlights their impact

and importance.

While this focus limits the library's scope in terms of model variety and deployment capabilities, it ensures that users can quickly and effectively train, test, and use transformer models without requiring extensive expertise in IR and NLP. This approach allows the Simple Transformers library to maintain its simplicity and accessibility while offering users pathways to deploy their models in real-world applications through full compatibility with Hugging Face libraries.

5.7.2 Future work

We aim to keep the Simple Transformers library up to date with the latest developments in information retrieval and natural language processing and to democratize access to the latest models and methodologies to a broader audience beyond the deep learning community.

5.7.3 Reflections on "Open-source for All"

We reflect on the three key focuses we outlined in Section 5.1 and how the Simple Transformers library contributes towards them.

(1) **Shared innovation:** We believe that most of the progress by the open-source community has been made towards this focus. The pace of advances in IR and NLP over the past few years bears testament to this fact. Simple Transformers is part of a large and growing movement, based on open-source developments and open science, dedicated to shared innovation in both industry and academia. However, we believe that there is still ample room to involve governmental and societal stakeholders, sectors where IR and language technology are less prevalent and even less understood, in this focus.

(2) **Empowering people:** The adoption of the Simple Transformers library, particularly outside the IR and NLP communities, and the research being conducted in other fields powered by IR and NLP technologies is evidence of progress towards the focus of empowering people with new tools. We believe that further progress can be made by increasing the availability of accessible tools, as well as increasing access to training materials and educational resources.

(3) **Diversity:** While it is difficult to directly estimate the impact of Simple Transformers on our third focus, we believe that bringing more people and diverse opinions to the discussion starts with empowering people to use language technology in the first place. Therefore, we hope that by pushing towards the second focus, we indirectly move towards the third focus as well.

6 Conclusions

In this chapter, we first recall the research questions that formed the basis of this thesis and summarize the answers to the research questions, along with findings and conclusions. In the second section of the chapter, we discuss possible future research directions.

6.1 Main Findings

RQ 1 How do negative sampling strategies affect the generalization of dense retrievers under distribution shift across domains and languages?

RQ 1 investigates how negative sampling strategies shape the generalization of dense retrievers under domain and language shift. Chapter 2 compares a range of negatives, lexical (e.g., BM25), in-batch and mined "hard" negatives, clustering-based variants, and iterative/ICT-style procedures, across in-distribution, out-of-distribution, and multilingual settings. The main findings are: (i) lexical negatives tend to yield the strongest in-distribution performance and provide a stable starting point; (ii) iterative clustered training (ICT-style) consistently improves robustness under distribution shift, delivering the best balance for cross-domain and multilingual generalization; and (iii) the general principle observed in English retrieval, that the inclusion of hard negatives is essential, extends to multilingual retrieval, with the exception of TAS-style (external teacher) clustering methods, which generalize less effectively.

Implications. Negative sampling is a primary driver of robustness. For production or evaluation scenarios that must handle unseen domains or languages, ICT-style negatives should be preferred; for purely in-domain workloads with abundant relevance signals, lexical negatives remain competitive and simpler to maintain.

Limitations and scope. The observed gains depend on corpus characteristics (topic granularity, language coverage) and mining budgets; extremely low-resource languages or highly specialized domains may require re-tuned mining curricula. We also note that improvements saturate when negatives become too adversarial relative to the model's capacity.

Conclusion. Careful choice of negative sampling materially affects dense retriever generalization: iterative/ICT-style negatives offer the most reliable improvements under

domain and language shift, whereas lexical negatives are strongest in-distribution. Consequently, sampling should be selected to match the deployment regime, i.e., robust ICT-style for shift-prone settings, lexical for stable in-domain retrieval.

RQ 2 Does training a dense passage retriever (DPR) model on data containing multiple queries per passage improve the generalizability of the model?

RQ 2 investigates whether training a dense passage retriever on data containing multiple queries per passage improves the model's generalizability. Chapter 3 introduces a data generation pipeline that uses query generation to create synthetic datasets where most passages are paired with multiple diverse queries. Models trained on these datasets are evaluated in both out-of-distribution and out-of-domain settings.

The main findings are: (i) DPR models trained on multiple-queries-per-passage data consistently outperform baselines trained on single-query datasets in five out of six out-of-distribution and twelve out of thirteen out-of-domain benchmarks; (ii) these improvements hold across heterogeneous domains and corpora, indicating that the gains are not dataset-specific; and (iii) the approach increases generalization without substantially increasing training cost or reducing in-domain performance.

Implications. Training with multiple queries per passage increases the diversity of passage–query alignments seen during learning, encouraging the model to encode more general semantic representations. This broader coverage mitigates overfitting to individual phrasings and improves robustness to unseen query formulations; an essential property for real-world retrieval systems deployed across dynamic or evolving domains.

Limitations and scope. The quality of generated queries depends on the underlying generation and filtering models. Weak filtering can introduce noise, potentially diluting the intended effect.

Conclusion. Training dense retrievers on datasets containing multiple queries per passage improves generalization across domains and distributions while maintaining in-domain performance. By exposing the model to a wider range of query-passage relationships during training, datasets with multiple queries per passage yield more robust retrieval models without significant computational or architectural changes.

RQ 3 How can relevance-based rewards be used in reward shaping to train small language models to answer based on retrieved evidence and to refuse when evidence is insufficient?

RQ 3 investigates how relevance-based rewards can be used in reward shaping to train small language models (LMs) to answer questions based on retrieved evidence and to refuse when evidence is insufficient. Chapter 4 introduces the reward shaping for robust refusal (RSRR) framework, which extends proximal policy optimization (PPO) with task-specific reward components designed for retrieval-augmented question answering. These components include relevance-based ranking rewards, correctness and refusal rewards, formatting rewards, and a KL-divergence penalty for stability.

The main findings are: (i) instruction-tuned small LMs struggle to reliably distinguish between answerable and unanswerable queries, frequently hallucinating answers

even under explicit refusal prompts; (ii) applying RSRR substantially improves both correct refusals and answer accuracy, yielding a relative gain of over 40% in correct refusals and a similar improvement in robustness to distractor documents; and (iii) the framework generalizes across multiple QA datasets (BEERQA, BIOASQ, PUBMEDQA, and STRATEGYQA), demonstrating that explicit reward shaping can reliably induce grounded answering and appropriate refusal behavior in small LMs.

Implications. RSRR shows that refusal behavior and answers based on retrieved evidence can be trained for explicitly rather than emerging implicitly from large-scale instruction tuning. By rewarding evidence-grounded reasoning and penalizing unsupported answers, small open-source LMs can be aligned for high-stakes retrieval-augmented applications where correctness and caution are equally important.

Limitations and scope. Improvements depend on the quality and diversity of retrieved evidence. Reward balance is also sensitive: excessive refusal penalties can lead to over-conservatism, while overly generous relevance rewards can reintroduce hallucinations. Scaling RSRR to larger models or longer reasoning chains may require adaptive or hierarchical reward scheduling.

Conclusion. Explicit relevance-based reward shaping enables small language models to reason over retrieved evidence and to refuse when that evidence is insufficient. The RSRR framework improves robustness to distractor documents and correct refusal, demonstrating that fine-grained reward design can make small LMs both more useful and more trustworthy in retrieval-augmented generation tasks.

RQ 4 Can an open-source framework like Simple Transformers lower the technical barriers to training and reproducing transformer-based retrieval and QA models?

RQ 4 investigates whether an open-source framework like *Simple Transformers* can lower the technical barriers to training and reproducing transformer-based retrieval and question-answering models. Chapter 5 presents the design and implementation of *Simple Transformers*, an open-source library that abstracts the complexity of model training, evaluation, and deployment through a unified interface and standardized configuration schema.

The main findings are: (i) modular task classes, covering retrieval, question answering, and related transformer applications, allow users with limited engineering expertise to train and evaluate models with minimal code; (ii) consistent configuration management and sensible defaults reduce friction in experimental workflows, promoting reproducibility across setups; and (iii) evidence of adoption, thousands of GitHub stars, millions of downloads, and widespread downstream use, indicates that the framework has had a measurable impact on accessibility and open experimentation in language model research.

Implications. Simple Transformers demonstrates that accessible open-source infrastructure can democratize the use of advanced transformer architectures beyond specialized research groups. Lowering the engineering threshold expands participation, accelerates iteration, and strengthens reproducibility practices across both academia and industry.

Limitations and scope. While *Simple Transformers* simplifies the process of training and evaluating models, it does not replace the need for expertise in model selection, dataset

design, and evaluation methodology. Its abstraction layers also trade off some flexibility, making highly customized architectures or distributed setups less straightforward to implement. Future work can address these limitations through modular extensions and tighter integration with emerging retrieval-augmented and evaluation pipelines.

Conclusion. Open-source frameworks like *Simple Transformers* can substantially lower the technical barriers to transformer-based retrieval and question answering. By abstracting routine engineering details and standardizing workflows, such frameworks foster more inclusive and reproducible research ecosystems, enabling a wider community to build, evaluate, and share transformer models effectively.

6.2 Future Work

Future work arising from this thesis spans three broad directions, specifically, advancing retrieval robustness, improving grounded answering and correct refusal in small language models, and broadening accessibility through open infrastructures.

Retrieval robustness and generalization

Chapters 2 and 3 focused on data-centric strategies for improving retriever robustness: negative sampling and training data composition. Several extensions remain open. First, while iterative clustered training (ICT) and training data containing multiple queries per passage improve zero-shot and cross-lingual generalization, their joint effect has not yet been studied. Combining diverse negatives with the data augmentation introduced in Chapter 3 could further reduce overfitting and improve transfer to unseen domains. Second, future work can explore adaptive sampling strategies that respond to the model's evolving uncertainty during training, selecting negatives or additional queries in a curriculum fashion. Such adaptive pipelines would better reflect real-world retrieval, where data availability and relevance signals change over time. Finally, scaling these analyses to larger multilingual or multimodal corpora, where passages may contain images, tables, or structured data, would test the limits of current retriever architectures and highlight how retrieval robustness interacts with modality alignment.

Grounded reasoning and calibrated refusal

Chapter 4 introduced reward shaping for robust refusal (RSRR), demonstrating that small language models can be explicitly trained to balance correctness and abstention. Future work can extend this in several ways. First, scaling RSRR to larger open models or multi-turn reasoning tasks will require dynamic reward scheduling and better credit assignment across reasoning steps. Second, integrating retrieval uncertainty directly into the reward function, so that the model learns to reason not only over evidence but also over confidence, could yield more stable refusal calibration. Another promising direction is cross-domain transfer: examining whether models trained to refuse unsupported answers in scientific QA also generalize to safety-critical domains such as healthcare, finance, or law. In these settings, nuanced refusal (distinguishing lack of evidence from lack of knowledge) remains largely unexplored. Lastly, RSRR can

be paired with human feedback or preference modeling to refine its refusal thresholds, bridging reinforcement learning with alignment research.

Open-source infrastructure and reproducibility

Chapter 5 emphasized lowering technical barriers through the *Simple Transformers* framework. Future work can extend this mission by improving interoperability with existing evaluation frameworks and retrieval-QA benchmarks, ensuring that experiments can be reproduced end-to-end with minimal configuration effort. Further development could focus on modular extensions for emerging transformer architectures, lightweight adapters for multilingual or low-resource settings, and tighter integration with experiment tracking and version control tools to enhance transparency. Another promising direction lies in usability research: studying how accessible tooling influences adoption, collaboration, and reproducibility across institutions. Understanding these effects empirically would help quantify the scientific and societal impact of open-source infrastructure in information retrieval and language modeling.

Towards unified retrieval-generation systems

Taken together, the chapters of this thesis suggest a broader agenda: moving from isolated retrieval and generation components toward unified, grounded reasoning systems. Future work can explore reinforcement learning setups that jointly optimize retrieval selection, reasoning quality, and refusal calibration. Future research in this direction may yield systems that know when to search, when to reason, and when to abstain, aligning robustness, faithfulness, and responsibility within a single framework.

Bibliography

[1] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann, J. R. Lee, Y. T. Lee, Y. Li, W. Liu, C. C. T. Mendes, A. Nguyen, E. Price, G. de Rosa, O. Saarikivi, A. Salim, S. Shah, X. Wang, R. Ward, Y. Wu, D. Yu, C. Zhang, and Y. Zhang. Phi-4 Technical Report. *arXiv preprint arXiv:2412.08905*, 2024. (Cited on page 47.)

[2] Z. Akata, D. Balliet, M. de Rijke, F. Dignum, V. Dignum, G. Eiben, A. Fokkens, D. Grossi, K. Hindriks, H. Hoos, H. Hung, C. Jonker, C. Monz, M. Neerincx, F. Oliehoek, H. Prakken, S. Schlobach, L. van der Gaag, F. van Harmelen, H. van Hoof, B. van Riemsdijk, A. van Wynsberghe, R. Verbrugge, B. Verheij, P. Vossen, and M. Welling. A research agenda for hybrid intelligence: Augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer*, 53(08): 18–28, Aug. 2020. (Cited on page 71.)

[3] L. B. Allal, A. Lozhkov, E. Bakouch, G. M. Blázquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlíček, A. P. Lajarín, V. Srivastav, J. Lochner, C. Fahlgren, X.-S. Nguyen, C. Fourrier, B. Burtenshaw, H. Larcher, H. Zhao, C. Zakka, M. Morlon, C. Raffel, L. von Werra, and T. Wolf. SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model. *arXiv preprint arXiv:2502.02737*, 2025. (Cited on page 47.)

[4] N. Asadi, D. Metzler, T. Elsayed, and J. Lin. Pseudo test collections for learning web search ranking functions. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1073–1082. ACM, 2011. (Cited on page 33.)

[5] A. Asai, X. Yu, J. Kasai, and H. Hajishirzi. One question answering model for many languages with cross-lingual dense passage retrieval. *Advances in Neural Information Processing Systems*, 34: 7547–7560, 2021. (Cited on page 12.)

[6] E. Asgari, N. Montaña-Brown, M. Dubois, S. Khalil, J. Balloch, J. A. Yeung, and D. Pimenta. A framework to assess clinical safety and hallucination rates of LLMs for medical text summarisation. *npj Digital Medicine*, 8(1):274, 2025. (Cited on pages 2 and 47.)

[7] L. Azzopardi and M. de Rijke. Automatic construction of known-item finding test beds. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*, pages 603–604. ACM, 2006. (Cited on page 33.)

[8] S. R. Baker, N. Bloom, and S. J. Davis. Measuring economic policy uncertainty. *The Quarterly Journal of Economics*, 131(4):1593–1636, 2016. (Cited on page 80.)

[9] P. Baudiš and J. Šedivỳ. Modeling of the question answering task in the YodaQA system. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 222–228. Springer, 2015. (Cited on page 79.)

[10] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, 2013. (Cited on page 79.)

[11] R. Berendsen, M. Tsagkias, M. de Rijke, and E. Meij. Generating pseudo test collections for learning to rank scientific articles. In *CLEF 2012: Conference and Labs of the Evaluation Forum*, pages 42–53. Springer, 2012. (Cited on page 33.)

[12] R. Berendsen, M. Tsagkias, W. Weerkamp, and M. de Rijke. Pseudo test collections for training and tuning microblog rankers. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 53–62. ACM, 2013. (Cited on page 33.)

[13] C. L. Bockting, E. A. M. van Dis, R. van Rooij, W. Zuidema, and J. Bollen. Living guidelines for generative AI — why scientists must oversee its use. *Nature*, 622:693–696, 2023. (Cited on page 71.)

[14] A. Bondarenko, M. Fröbe, J. Kiesel, S. Syed, T. Gurcke, M. Beloucif, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, et al. Overview of Touché 2022: Argument retrieval. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 311–336. Springer, 2022. (Cited on page 35.)

[15] L. Bonifacio, V. Jeronymo, H. Q. Abonizio, I. Campiotti, M. Fadaee, R. Lotufo, and R. Nogueira. mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset. *arXiv preprint arXiv:2108.13897*, 2022. (Cited on pages 12 and 19.)

[16] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, pages 2206–2240. PMLR, 2022. (Cited on pages 1, 47, and 48.)

[17] V. Boteva, D. Gholipour, A. Sokolov, and S. Riezler. A full-text learning to rank dataset for medical information retrieval. In N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio,

C. Hauff, and G. Silvello, editors, *Advances in Information Retrieval*, pages 716–722, Cham, 2016. Springer International Publishing. ISBN 978-3-319-30671-1. (Cited on page 35.)

[18] T. Breuer and M. Maistro. Toward Evaluating the Reproducibility of Information Retrieval Systems with Simulated Users. In *Proceedings of the 2nd ACM Conference on Reproducibility and Replicability*, ACM REP '24, pages 25–29, New York, NY, USA, July 2024. Association for Computing Machinery. ISBN 979-8-4007-0530-4. doi: 10.1145/3641525.3663619. (Cited on page 3.)

[19] L. Chen, R. Zhang, J. Guo, Y. Fan, and X. Cheng. Controlling Risk of Retrieval-augmented Generation: A Counterfactual Prompting Framework. *arXiv preprint arXiv:2409.16146*, 2024. (Cited on page 1.)

[20] J. H. Clark, E. Choi, M. Collins, D. Garrette, T. Kwiatkowski, V. Nikolaev, and J. 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. (Cited on pages 1 and 18.)

[21] T. G. Coan, C. Boussalis, J. Cook, and M. O. Nanko. Computer-assisted classification of contrarian claims about climate change. *Scientific Reports*, 11(1):22320, Nov. 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-01714-4. (Cited on page 79.)

[22] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. Weld. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.207. (Cited on page 35.)

[23] K. Cruse, A. Trewartha, S. Lee, Z. Wang, H. Huo, T. He, O. Kononova, A. Jain, and G. Ceder. Text-mined dataset of gold nanoparticle synthesis procedures, morphologies, and size entities. *Scientific Data*, 9(1):234, 2022. (Cited on page 79.)

[24] F. Cuconasu, G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonellotto, and F. 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*, pages 719–729, 2024. (Cited on page 52.)

[25] M. de Rijke, B. van den Hurk, F. Salim, A. A. Khourdajie, N. Bai, R. Calzone, D. Curran, G. Demil, L. Frew, N. Gießing, M. K. Gupta, M. Heuss, S. Hobeichi, D. Huard, J. Kang, A. Lucic, T. Mallick, S. Nath, A. Okem, B. Pernici, T. Rajapakse, H. Saleem, H. Scells, N. Schneider, D. Spina, Y. Tian, E. Totin, A. Trotman, R. Valavandan, D. Workneh, and Y. Xie. Report on the 1st workshop on information retrieval for climate impact (MANILA24) at SIGIR 2024. *SIGIR Forum*, 59(1):1–23, Oct. 2025. ISSN 0163-5840. doi: 10.1145/3769733.3769737.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. (Cited on pages 1, 11, 15, 31, and 36.)

[27] T. Diggelmann, J. Boyd-Graber, J. Bulian, M. Ciaramita, and M. Leippold. CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims. *arXiv preprint arXiv:2012.00614*, 2021. (Cited on page 35.)

[28] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. THE FAISS LIBRARY. *IEEE Transactions on Big Data*, pages 1–17, 2025. ISSN 2332-7790. doi: 10.1109/TBDATA.2025.3618474. (Cited on pages 2 and 72.)

[29] P. Galuščáková, D. W. Oard, and S. Nair. Cross-language Information Retrieval. *arXiv preprint arXiv:2111.05988*, 2022. (Cited on page 27.)

[30] L. Gao, X. Ma, J. Lin, and J. 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, pages 3120–3124, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 978-1-4503-9408-6. doi: 10.1145/3539618.3591805. (Cited on pages 72 and 79.)

[31] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997*, 2024. (Cited on page 1.)

[32] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. (Cited on pages 51 and 55.)

[33] O. Gospodnetic and E. Hatcher. *Lucene in Action*. Manning Publications, 2004. (Cited on page 71.)

[34] S. Gugger, L. Debut, T. Wolf, P. Schmid, Z. Mueller, S. Mangrulkar, M. Sun, and B. Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable, 2022. (Cited on pages 72 and 77.)

[35] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR, 2020. (Cited on pages 47 and 48.)

[36] S. Haefliger, G. von Krogh, and S. Spaeth. Code reuse in open source software. *Management Science*, 54(1):180–193, 2007. (Cited on page 71.)

[37] F. Hasibi, F. Nikolaev, C. Xiong, K. Balog, S. E. Bratsberg, A. Kotov, and J. 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, pages 1265–1268, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080751. (Cited on page 35.)

[38] D. Hawking, B. Billerbeck, P. Thomas, and N. Craswell. *Simulating Information Retrieval Test Collections*. Morgan & Claypool, 2020. (Cited on page 33.)

[39] S. Hofstätter, S.-C. Lin, J.-H. Yang, J. Lin, and A. Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122, 2021. (Cited on pages 11, 12, 13, 16, 18, 22, 26, 27, and 28.)

[40] D. Hoogeveen, K. M. Verspoor, and T. Baldwin. CQADupStack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium*, pages 1–8, 2015. (Cited on page 35.)

[41] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. (Cited on page 77.)

[42] J. Hsia, A. Shaikh, Z. Z. Wang, and G. Neubig. RAGGED: Towards Informed Design of Scalable and Stable RAG Systems. In *Forty-Second International Conference on Machine Learning*, June 2025. (Cited on page 1.)

[43] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025. (Cited on pages 1, 3, and 47.)

[44] G. Izacard and E. Grave. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. *arXiv preprint arXiv:2007.01282*, 2021. (Cited on page 1.)

[45] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023. (Cited on pages 1 and 47.)

[46] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, 55(12):248:1–248:38, Mar. 2023. ISSN 0360-0300. doi: 10.1145/3571730. (Cited on page 1.)

[47] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, and X. Lu. PubMedQA: A dataset for biomedical research question answering. 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 2567–2577, 2019. (Cited on pages 50 and 55.)

[48] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. (Cited on pages 1, 14, and 72.)

[49] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In R. Barzilay and M.-Y. 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. doi: 10.18653/v1/P17-1147. (Cited on page 79.)

[50] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. 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. Association for Computational Linguistics, Nov. 2020. (Cited on pages 1, 2, 11, 12, 14, 15, 16, 18, 26, 31, 32, 33, 34, 36, 37, 72, 75, and 79.)

[51] O. Khattab and M. Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48, 2020. (Cited on pages 1, 2, 11, 12, 28, 31, 33, 36, and 43.)

[52] J. Kim and W. B. Croft. Retrieval experiments using pseudo-desktop collections. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1297–1306. ACM, 2009. (Cited on page 33.)

[53] A. Krithara, A. Nentidis, K. Bougiatiotis, and G. Paliouras. BioASQ-QA: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170, 2023. (Cited on pages 51 and 55.)

[54] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. (Cited on pages 32 and 79.)

[55] C. Lassance and S. Clinchant. The Tale of Two MSMARCO - and Their Unfair Comparisons. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 2431–2435, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 978-1-4503-9408-6. doi: 10.1145/3539618.3592071. (Cited on page 11.)

[56] J. Lee, H. Cha, Y. Hwangbo, and W. Cheon. Enhancing large language model reliability: Minimizing hallucinations with dual retrieval-augmented generation based on the latest diabetes guidelines. *Journal of Personalized Medicine*, 14(12):1131, 2024. (Cited on page 49.)

[57] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. (Cited on pages 1, 47, 48, and 77.)

[58] Q. Lhoest, A. Villanova del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, L. Tunstall, J. Davison, M. Šaško, G. Chhablani, B. Malik, S. Brandeis, T. Le Scao, V. Sanh, C. Xu, N. Patry, A. McMillan-Major, P. Schmid, S. Gugger, C. Delangue, T. Matussière, L. Debut, S. Bekman, P. Cistac, T. Goehringer, V. Mustar, F. Lagunas, A. Rush, and T. Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. (Cited on page 72.)

[59] X. Li, J. Jin, Y. Zhou, Y. Zhang, P. Zhang, Y. Zhu, and Z. Dou. From Matching to Generation: A Survey on Generative Information Retrieval. *ACM Transactions on Information Systems*, 43(3):1–62, May 2025. ISSN 1046-8188, 1558-2868. doi: 10.1145/3722552. (Cited on page 1.)

[60] Y. Li, M. Franz, M. A. Sultan, B. Iyer, Y.-S. Lee, and A. Sil. Learning Cross-Lingual IR from an English Retriever. *arXiv preprint arXiv:2112.08185*, 2021. (Cited on pages 12 and 48.)

[61] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. 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. (Cited on pages 2 and 72.)

[62] S. Lin, J. Hilton, and O. Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods. *arXiv preprint arXiv:2109.07958*, 2022. (Cited on page 1.)

[63] S.-C. Lin, A. Asai, M. Li, B. Oguz, J. Lin, Y. Mehdad, W.-t. Yih, and X. Chen. How to Train Your DRAGON: Diverse Augmentation Towards Generalizable Dense Retrieval. *arXiv preprint arXiv:2302.07452*, 2023. (Cited on page 77.)

[64] Y. Lin, H. Lin, W. Xiong, S. Diao, J. Liu, J. Zhang, R. Pan, H. Wang, W. Hu, H. Zhang, H. Dong, R. Pi, H. Zhao, N. Jiang, H. Ji, Y. Yao, and T. Zhang. Mitigating the Alignment Tax of RLHF. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 580–606, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.35. (Cited on page 48.)

[65] E. Lindgren, S. Reddi, R. Guo, and S. Kumar. Efficient training of retrieval models using negative cache. *Advances in Neural Information Processing Systems*, 34:4134–4146, 2021. (Cited on page 26.)

[66] C. Y. Liu, L. Zeng, J. Liu, R. Yan, J. He, C. Wang, S. Yan, Y. Liu, and Y. Zhou. Skywork-Reward: Bag of Tricks for Reward Modeling in LLMs. *arXiv preprint arXiv:2410.18451*, 2024. (Cited on page 56.)

[67] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the Middle: How Language Models Use Long Contexts. *arXiv preprint arXiv:2307.03172*, 2023. (Cited on pages 1 and 3.)

[68] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019. (Cited on page 35.)

[69] X. Ma, L. Wang, N. Yang, F. Wei, and J. 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, pages 2421–2425, New York, NY, USA, July 2024. Association for Computing Machinery. ISBN 979-8-4007-0431-4. doi: 10.1145/3626772.3657951. (Cited on page 56.)

[70] S. MacAvaney, L. Soldaini, and N. Goharian. Teaching a new dog old tricks: Resurrecting multilingual retrieval using zero-shot learning. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 246–254. Springer, 2020. (Cited on pages 12 and 27.)

[71] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, and A. Balahur. WWW'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of the the Web Conference 2018*, Www '18, pages 1941–1942, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5640-4. doi: 10.1145/3184558.3192301. (Cited on page 35.)

[72] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan. PEFT: State-of-the-art parameter-efficient fine-tuning methods, 2022. (Cited on page 77.)

[73] M. Mazzucato. *Mission Economy*. Penguin Random House, 2021. (Cited on page 71.)

[74] T. Mu, A. Helyar, J. Heidecke, J. Achiam, A. Vallone, I. D. Kivlichan, M. Lin, A. Beutel, J. Schulman, and L. Weng. Rule based rewards for language model safety. In *The Thirty-Eighth Annual Conference on Neural Information Processing Systems*, 2024. (Cited on pages 49 and 50.)

[75] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, and J. Schulman. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2022. (Cited on page 48.)

[76] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ Nips*, 2016. (Cited on pages 18 and 75.)

[77] T. Nguyen, S. MacAvaney, and A. Yates. A unified framework for learned sparse retrieval. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part III*, pages 101–116. Springer, 2023. (Cited on page 72.)

[78] T. Nguyen, P. Chin, and Y.-W. Tai. Reward-RAG: Enhancing RAG with Reward Driven Supervision. *arXiv preprint arXiv:2410.03780*, 2024. (Cited on page 48.)

[79] C. Niu, Y. Wu, J. Zhu, S. Xu, K. Shum, R. Zhong, J. Song, and T. Zhang. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. *arXiv preprint arXiv:2401.00396*, 2024. (Cited on pages 1 and 3.)

[80] R. Nogueira, Z. Jiang, and J. Lin. Document Ranking with a Pretrained Sequence-to-Sequence Model. *arXiv preprint arXiv:2003.06713*, 2020. (Cited on page 74.)

[81] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and D. Johnson. Terrier information retrieval platform. In *Advances in Information Retrieval*, pages 517–519. Springer, 2005. (Cited on page 72.)

[82] D. Owen, D. Antypas, A. Hassoulas, A. F. Pardiñas, L. Espinosa-Anke, J. C. Collados, et al. Enabling early health care intervention by detecting depression in users of web-based forums using language models: Longitudinal analysis and evaluation. *JMIR AI*, 2(1):e41205, 2023. (Cited on page 80.)

[83] S. Pandit, J. Xu, J. Hong, Z. Wang, T. Chen, K. Xu, and Y. Ding. MedHallu: A comprehensive benchmark for detecting medical hallucinations in large language models. *arXiv preprint arXiv:2502.14302*, 2025. (Cited on page 49.)

[84] C. Peters and P. Sheridan. Multilingual information access. In *Lectures on Information Retrieval: Third European Summer-School, ESSIR 2000 Varenna, Italy, September 11–15, 2000 Revised Lectures*, pages 51–80. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-45368-0. doi: 10.1007/3-540-45368-7_3. (Cited on page 12.)

[85] P. Qi, H. Lee, O. T. Sido, and C. D. Manning. Retrieve, rerank, read, then iterate: Answering open-domain questions of arbitrary complexity from text. *arXiv preprint arXiv:2010.12527*, 2020. (Cited on pages 1, 51, and 55.)

[86] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. 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. ISSN 1533-7928. (Cited on pages 1 and 35.)

[87] T. C. Rajapakse and M. de Rijke. Improving the generalizability of the dense passage retriever using generated datasets. In *European Conference on Information Retrieval*, pages 94–109. Springer, 2023. (Cited on pages 1 and 2.)

[88] T. C. Rajapakse and M. de Rijke. Reward shaping for robust refusal in small language models for retrieval-augmented question answering. In *Under Review*, 2026. (Cited on page 2.)

[89] T. C. Rajapakse, A. Yates, and M. de Rijke. Negative sampling techniques for dense passage retrieval in a multilingual setting. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 575–584, 2024. (Cited on pages 1 and 2.)

[90] T. C. Rajapakse, A. Yates, and M. de Rijke. Simple Transformers: Open-source for all. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 209–215, 2024. (Cited on page 2.)

[91] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In J. Su, K. Duh, and X. Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. (Cited on page 79.)

[92] P. Rajpurkar, R. Jia, and P. Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. *arXiv preprint arXiv:1806.03822*, 2018. (Cited on page 3.)

[93] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, Nov. 2023. ISSN 2307-387X. doi: 10.1162/tacl_a_00605. (Cited on pages 1 and 77.)

[94] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. (Cited on page 72.)

[95] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, Apr. 2009. ISSN 1554-0669. doi: 10.1561/1500000019. (Cited on page 16.)

[96] R. Robson, E. Kelsey, A. Goel, S. Nasir, E. Robson, M. Garn, M. Lisle, J. Kitchens, S. Rugaber, and F. Ray. Intelligent links: AI-supported connections between employers and colleges. *AI Magazine*, 43 (1):75–82, 2022. (Cited on page 80.)

[97] G. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, M. Fadaee, R. Lotufo, and R. Nogueira. In Defense of Cross-Encoders for Zero-Shot Retrieval. *arXiv preprint arXiv:2212.06121*, 2022. (Cited on page 26.)

[98] G. M. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, M. Fadaee, R. Lotufo, and R. Nogueira. No Parameter Left Behind: How Distillation and Model Size Affect Zero-Shot Retrieval. *arXiv preprint arXiv:2206.02873*, 2022. (Cited on page 26.)

[99] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. (Cited on page 3.)

[100] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. Hidden Technical Debt in Machine Learning Systems. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. (Cited on page 3.)

[101] P. Shi, R. Zhang, H. Bai, and J. Lin. Cross-lingual training of dense retrievers for document retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 251–253, 2021. (Cited on pages 12 and 27.)

[102] B. Shneiderman. *Twin-Win Research*. Springer, 2018. (Cited on page 71.)

[103] B. Shneiderman. *Human-Centered AI*. Oxford University Press, 2022. (Cited on page 71.)

[104] Y. R. Shrestha, G. von Krogh, and S. Feuerriegel. Building open-source AI. *Nature Computational Science*, 3:908–911, 2023. (Cited on page 71.)

[105] D. Siddarth, D. Acemoglu, D. Allen, K. Crawford, J. Evans, M. Jordan, and E. G. Weyl. How AI fails us. Technical report, Justic, Health, and Democracy Impact Initiative & Carr Cenetr for Human Rights Policy, 2021. (Cited on page 71.)

[106] J. Susskind. *The Digital Republic: On Freedom and Democracy in the 21st Century*. Bloomsbury, 2022. (Cited on page 71.)

[107] O. Tafjord and P. Clark. General-Purpose Question-Answering with Macaw. *arXiv preprint arXiv:2109.02593*, 2021. (Cited on page 35.)

[108] J. Tague, M. Nelson, and H. Wu. Problems in the simulation of bibliographic retrieval systems. In *Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval*, pages 236–255. Butterworth & Co., 1980. (Cited on page 33.)

[109] Y. Tan, Y. Jiang, Y. Li, J. Liu, X. Bu, W. Su, X. Yue, X. Zhu, and B. Zheng. Equilibrate RLHF: Towards balancing helpfulness-safety trade-off in large language models. *arXiv preprint arXiv:2502.11555*, 2025. (Cited on page 49.)

[110] Y. Tay, V. Tran, M. Dehghani, J. Ni, D. Bahri, H. Mehta, Z. Qin, K. Hui, Z. Zhao, and J. Gupta. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843, 2022. (Cited on page 1.)

[111] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *arXiv preprint arXiv:2104.08663*, 2021. (Cited on pages 1, 2, 12, 19, 24, 25, 26, 33, 35, and 77.)

[112] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. 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. doi: 10.18653/v1/N18-1074. (Cited on page 35.)

[113] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023. (Cited on pages 1, 3, and 77.)

[114] P. Trust, A. Zahran, and R. Minghim. Understanding the influence of news on society decision making: Application to economic policy uncertainty. *Neural Computing and Applications*, 35(20): 14929–14945, 2023. (Cited on page 80.)

[115] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. ukasz Kaiser, and I. Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. (Cited on pages 26 and 72.)

[116] N. Vermeer, V. Provatorova, D. Graus, T. Rajapakse, and S. Mesbah. Using Robbert and extreme multi-label classification to extract implicit and explicit skills from Dutch job descriptions. *Compjobs' 22: Computational Jobs Marketplace*, 1(1):2–6, 2022.

[117] E. Voorhees, T. Alam, S. Bedrick, D. Demner-Fushman, W. R. Hersh, K. Lo, K. Roberts, I. Soboroff, and L. L. Wang. TREC-COVID: Constructing a pandemic information retrieval test collection. *Proceedings of The Twelfth Annual International Acm Sigir Conference On Research and Development in Information Retrieval*, 54(1), Feb. 2021. ISSN 0163-5840. doi: 10.1145/3451964.3451965. (Cited on page 35.)

[118] E. M. Voorhees. The TREC-8 question answering track report. In *Proceedings of TREC-8*, pages 77–82, 1999. (Cited on pages 26 and 33.)

[119] H. Wachsmuth, S. Syed, and B. Stein. Retrieval of the best counterargument without prior topic knowledge. In *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. doi: 10.18653/v1/P18-1023. (Cited on page 35.)

[120] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. 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, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.609. (Cited on page 35.)

[121] J. Wallat, M. Heuss, M. de Rijke, and A. Anand. Correctness is not Faithfulness in RAG Attributions. *arXiv preprint arXiv:2412.18004*, 2024. (Cited on page 1.)

[122] Z. Wan, X. Wang, C. Liu, S. Alam, Y. Zheng, J. Liu, Z. Qu, S. Yan, Y. Zhu, Q. Zhang, M. Chowdhury, and M. Zhang. Efficient Large Language Models: A Survey. *arXiv preprint arXiv:2312.03863*, 2024. (Cited on page 47.)

[123] S. Wang and G. Zuccon. Balanced topic aware sampling for effective dense retriever: A reproducibility study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sigir '23, pages 2542–2551, New York, NY, USA, 2023. Association for Computing Machinery. (Cited on pages 11 and 28.)

[124] S. Wang, S. Zhang, J. Zhang, R. Hu, X. Li, T. Zhang, J. Li, F. Wu, G. Wang, and E. Hovy. Reinforcement Learning Enhanced LLMs: A Survey. *arXiv preprint arXiv:2412.10400*, 2025. (Cited on page 48.)

[125] X. Wang, Z. Tang, J. Guo, T. Meng, C. Wang, T. Wang, and W. Jia. Empowering edge intelligence: A comprehensive survey on on-device ai models. *ACM Computing Surveys*, 57(9):1–39, 2025. (Cited on page 47.)

[126] J. Wei, H. Zhou, X. Zhang, D. Zhang, Z. Qiu, W. Wei, J. Li, W. Ouyang, and S. Sun. Retrieval is Not Enough: Enhancing RAG Reasoning through Test-Time Critique and Optimization. *arXiv preprint arXiv:2504.14858*, 2025. (Cited on page 1.)

[127] S. Whitehead, S. Petryk, V. Shakib, J. Gonzalez, T. Darrell, A. Rohrbach, and M. Rohrbach. Reliable visual question answering: Abstain rather than answer incorrectly. In *European Conference on Computer Vision*, pages 148–166. Springer, 2022. (Cited on page 47.)

[128] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. Transformers: State-of-the-art natural language processing. In Q. Liu and D. Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. (Cited on pages 2, 19, 72, and 73.)

[129] B. Workshop, T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, J. Tow, A. M. Rush, S. Biderman, A. Webson, P. S. Ammanamanchi, T. Wang, B. Sagot, N. Muennighoff, A. V. del Moral, O. Ruwase, R. Bawden, S. Bekman, A. McMillan-Major, I. Beltagy, H. Nguyen, L. Saulnier, S. Tan, P. O. Suarez, V. Sanh, H. Laurençon, Y. Jernite, J. Launay, M. Mitchell, C. Raffel, A. Gokaslan, A. Simhi, A. Soroa, A. F. Aji, A. Alfassy, A. Rogers, A. K. Nitzav, C. Xu, C. Mou, C. Emezue, C. Klamm, C. Leong, D. van Strien, D. I. Adelani, D. Radev, E. G. Ponferrada, E. Levkovizh, E. Kim, E. B. Natan, F. D. Toni, G. Dupont, G. Kruszewski, G. Pistilli, H. Elsahar, H. Benyamina, H. Tran, I. Yu, I. Abdulmumin, I. Johnson, I. Gonzalez-Dios, J. de la Rosa, J. Chim, J. Dodge, J. Zhu, J. Chang, J. Frohberg, J. Tobing, J. Bhattacharjee, K. Almubarak, K. Chen, K. Lo, L. V. Werra, L. Weber, L. Phan, L. B. allal, L. Tanguy, M. Dey, M. R. Muñoz, M. Masoud, M. Grandury, M. Šaško, M. Huang, M. Coavoux, M. Singh, M. T.-J. Jiang, M. C. Vu, M. A. Jauhar, M. Ghaleb, N. Subramani, N. Kassner, N. Khamis, O. Nguyen, O. Espejel, O. de Gibert, P. Villegas, P. Henderson, P. Colombo, P. Amuok, Q. Lhoest, R. Harliman, R. Bommasani, R. L. López, R. Ribeiro, S. Osei, S. Pyysalo, S. Nagel, S. Bose, S. H. Muhammad, S. Sharma, S. Longpre, S. Nikpoor, S. Silberberg, S. Pai, S. Zink, T. T. Torrent, T. Schick, T. Thrush, V. Danchev, V. Nikoulina, V. Laippala, V. Lepercq, V. Prabhu, Z. Alyafeai, Z. Talat, A. Raja, B. Heinzerling, C. Si, D. E. Taşar, E. Salesky, S. J. Mielke, W. Y. Lee, A. Sharma, A. Santilli, A. Chaffin, A. Stiegler, D. Datta, E. Szczechla, G. Chhablani, H. Wang, H. Pandey, H. Strobelt, J. A. Fries, J. Rozen, L. Gao, L. Sutawika, M. S. Bari, M. S. Al-shaibani, M. Manica, N. Nayak, R. Teehan, S. Albanie, S. Shen, S. Ben-David, S. H. Bach, T. Kim, T. Bers, T. Fevry, T. Neeraj, U. Thakker, V. Raunak, X. Tang, Z.-X. Yong, Z. Sun, S. Brody, Y. Uri, H. Tojarieh, A. Roberts, H. W. Chung, J. Tae, J. Phang, O. Press, C. Li, D. Narayanan, H. Bourfoune, J. Casper, J. Rasley, M. Ryabinin, M. Mishra, M. Zhang, M. Shoeybi, M. Peyrounette, N. Patry, N. Tazi, O. Sanseviero, P. von Platen, P. Cornette, P. F. Lavallée, R. Lacroix, S. Rajbhandari, S. Gandhi, S. Smith, S. Requena, S. Patil, T. Dettmers, A. Baruwa, A. Singh, A. Cheveleva, A.-L. Ligozat, A. Subramonian, A. Névéol, C. Lovering, D. Garrette, D. Tunuguntla, E. Reiter, E. Taktasheva, E. Voloshina, E. Bogdanov, G. I. Winata, H. Schoelkopf, J.-C. Kalo, J. Novikova, J. Z. Forde, J. Clive, J. Kasai, K. Kawamura, L. Hazan, M. Carpuat, M. Clinciu, N. Kim, N. Cheng, O. Serikov, O. Antverg, O. van der Wal, R. Zhang, R. Zhang, S. Gehrmann, S. Mirkin, S. Pais, T. Shavrina, T. Scialom, T. Yun, T. Limisiewicz, V. Rieser, V. Protasov, V. Mikhailov, Y. Pruksachatkun, Y. Belinkov, Z. Bamberger, Z. Kasner, A. Rueda, A. Pestana, A. Feizpour, A. Khan, A. Faranak, A. Santos, A. Hevia, A. Unldreaj, A. Aghagol, A. Abdollahi, A. Tammour, A. HajiHosseini, B. Behroozi, B. Ajibade, B. Saxena, C. M. Ferrandis, D. McDuff, D. Contractor, D. Lansky, D. David, D. Kiela, D. A. Nguyen, E. Tan, E. Baylor, E. Ozoani, F. Mirza, F. Ononiwu, H. Rezanejad, H. Jones, I. Bhattacharya, I. Solaiman, I. Sedenko, I. Nejadgholi, J. Passmore, J. Seltzer, J. B. Sanz, L. Dutra, M. Samagaio, M. Elbadri, M. Mieskes, M. Gerchick, M. Akinlolu, M. McKenna, M. Qiu, M. Ghauri, M. Burynok, N. Abrar, N. Rajani, N. Elkott, N. Fahmy, O. Samuel, R. An, R. Kromann, R. Hao, S. Alizadeh, S. Shubber, S. Wang, S. Roy, S. Viguier, T. Le, T. Oyebade, T. Le, Y. Yang, Z. Nguyen, A. R. Kashyap, A. Palasciano, A. Callahan, A. Shukla, A. Miranda-Escalada, A. Singh, B. Beilharz, B. Wang, C. Brito, C. Zhou, C. Jain, C. Xu, C. Fourrier, D. L. Periñán, D. Molano, D. Yu, E. Manjavacas, F. Barth, F. Fuhrimann, G. Altay, G. Bayrak, G. Burns, H. U. Vrabec, I. Bello, I. Dash, J. Kang, J. Giorgi, J. Golde, J. D. Posada, K. R. Sivaraman, L. Bulchandani, L. Liu, L. Shinzato, M. H. de Bykhovetz, M. Takeuchi, M. Pàmies, M. A. Castillo, M. Nezhurina, M. Sänger, M. Samwald, M. Cullan, M. Weinberg, M. D. Wolf, M. Mihaljcic, M. Liu, M. Freidank, M. Kang, N. Seelam, N. Dahlberg, N. M. Broad, N. Muellner, P. Fung, P. Haller, R. Chandrasekhar, R. Eisenberg, R. Martin, R. Canalli, R. Su, R. Su, S. Cahyawijaya, S. Garda, S. S. Deshmukh, S. Mishra, S. Kiblawi, S. Ott, S. Sang-aroonsiri, S. Kumar, S. Schweter, S. Bharati, T. Laud, T. Gigant, T. Kainuma, W. Kusa, Y. Labrak, Y. S. Bajaj, Y. Venkatraman, Y. Xu, Y. Xu, Y. Xu, Z. Tan, Z. Xie, Z. Ye, M. Bras, Y. Belkada, and T. Wolf. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv preprint arXiv:2211.05100*, 2023. (Cited on page 71.)

[130] S. Xiao, Z. Liu, Y. Shao, and Z. Cao. RetroMAE: Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022*

Conference on Empirical Methods in Natural Language Processing, pages 538–548, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.35. (Cited on page 1.)

[131] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. Bennett, J. Ahmed, and A. Overwijk. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *arXiv preprint arXiv:2007.00808*, 2020. (Cited on pages 2, 11, 12, 13, 16, 17, 18, 26, 31, 33, 43, and 75.)

[132] C. Xu, D. Guo, N. Duan, and J. McAuley. LaPraDoR: Unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3557–3569, 2022. (Cited on page 26.)

[133] H. Xu, Z. Zhu, S. Zhang, D. Ma, S. Fan, L. Chen, and K. Yu. Rejection improves reliability: Training llms to refuse unknown questions using RL from knowledge feedback. *arXiv preprint arXiv:2403.18349*, 2024. (Cited on pages 1, 2, 47, and 49.)

[134] J. Xu, Z. Li, W. Chen, Q. Wang, X. Gao, Q. Cai, and Z. Ling. On-Device Language Models: A Comprehensive Review. *arXiv preprint arXiv:2409.00088*, 2024. (Cited on page 47.)

[135] L. Yang, M. Zhang, C. Li, M. Bendersky, and M. Najork. Beyond 512 Tokens: Siamese Multi-depth Transformer-based Hierarchical Encoder for Long-Form Document Matching. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 1725–1734, New York, NY, USA, Oct. 2020. Association for Computing Machinery. ISBN 978-1-4503-6859-9. doi: 10.1145/3340531.3411908. (Cited on page 80.)

[136] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. 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, Tenth century–2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. (Cited on page 35.)

[137] J. Zhan, J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512, 2021. (Cited on page 26.)

[138] J. Zhan, X. Xie, J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. Evaluating Interpolation and Extrapolation Performance of Neural Retrieval Models. *arXiv preprint arXiv:2204.11447*, 2022. (Cited on pages 2 and 26.)

[139] H. Zhang, J. Song, J. Zhu, Y. Wu, T. Zhang, and C. Niu. RAG-reward: Optimizing RAG with reward modeling and RLHF. *arXiv preprint arXiv:2501.13264*, 2025. (Cited on page 49.)

[140] L. Zhang and X. Zhao. An overview of cross-language information retrieval. In *Artificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part I 6*, pages 26–37. Springer, 2020. (Cited on page 27.)

[141] P. Zhang, G. Zeng, T. Wang, and W. Lu. TinyLlama: An Open-Source Small Language Model. *arXiv preprint arXiv:2401.02385*, 2024. (Cited on pages 2, 3, and 47.)

[142] W. Zhang, S. Vakulenko, T. Rajapakse, Y. Xu, and E. Kanoulas. Tackling query-focused summarization as a knowledge-intensive task: A pilot study. *arXiv preprint arXiv:2112.07536*, 2023.

[143] W. Zhang, J.-H. Huang, S. Vakulenko, Y. Xu, T. Rajapakse, and E. Kanoulas. Beyond relevant documents: A knowledge-intensive approach for query-focused summarization using large language models. In A. Antonacopoulos, S. Chaudhuri, R. Chellappa, C.-L. Liu, S. Bhattacharya, and U. Pal, editors, *Pattern Recognition*, pages 89–104, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-78495-8. doi: 10.1007/978-3-031-78495-8_6.

[144] X. Zhang, X. Ma, P. Shi, and J. Lin. Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval. In D. Ataman, A. Birch, A. Conneau, O. Firat, S. Ruder, and G. G. Sahin, editors, *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.mrl-1.12. (Cited on pages 1, 2, 12, 18, and 21.)

[145] X. Zhang, N. Thakur, O. Ogundepo, E. Kamalloo, D. Alfonso-Hermelo, X. Li, Q. Liu, M. Rezagholizadeh, and J. Lin. Making a MIRACL: Multilingual Information Retrieval Across a Continuum of Languages. *arXiv preprint arXiv:2210.09984*, 2022. (Cited on pages 2 and 19.)

[146] X. Zhang, K. Ogueji, X. Ma, and J. Lin. Toward Best Practices for Training Multilingual Dense Retrieval Models. *ACM Trans. Inf. Syst.*, 42(2):39:1–39:33, Sept. 2023. ISSN 1046-8188. doi: 10.1145/3613447. (Cited on pages 12, 15, 21, and 27.)

[147] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Liu, Z. Dou, and J.-R. Wen. Large Language Models for Information Retrieval: A Survey. *ACM Transactions on Information Systems*, page 3748304, Sept. 2025. ISSN 1046-8188, 1558-2868. doi: 10.1145/3748304. (Cited on page 1.)

[148] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-Tuning Language Models from Human Preferences. *arXiv preprint arXiv:1909.08593*, 2020. (Cited on pages 3 and 48.)

Summary

Information retrieval increasingly relies on systems that must find evidence, reason over it, and avoid answering when support is lacking. This thesis studies how to train and evaluate such systems across three strands: (i) making dense retrievers robust under distribution shifts, (ii) training small language models to ground answers and refuse when evidence is insufficient, and (iii) improving accessibility and reproducibility through open-source tooling.

Part I: Robust retrieval under distribution shifts. Chapter 2 investigates negative sampling for multilingual dense passage retrieval and introduces iterative clustered training (ICT), which refreshes hard negatives by clustering query or passage representations during training. In multilingual experiments, BM25-based negatives yield the strongest in-distribution effectiveness, while ICT, especially the passage-clustered variant (ICT-P), provides the best out-of-distribution and zero-shot performance, with lower resource demands than full-corpus iterative mining. Chapter 3 further shows that data composition matters: training on synthetically generated datasets with multiple queries per passage improves zero-shot and out-of-domain generalization compared to single-query training with all other factors controlled.

Part II: Robust refusal and evidence-based answering. Chapter 4 studies retrieval augmented QA with small instruction-tuned models in three evidence conditions: (i) *Vanilla* (gold evidence present), (ii) *Distractors* (gold evidence plus irrelevant passages), and (iii) *No-Res* (gold evidence removed; correct behavior is refusal). Baselines tend to over-answer in noisy contexts and under-refuse when evidence is absent. Chapter 4 introduces reward shaping for robust refusal (RSRR), a PPO-based approach that combines a relevance reward (comparing reasoning to retrieved documents) with scheduled correctness, formatting, and KL terms. Across BioASQ, BeerQA, PubMedQA, and StrategyQA, RSRR improves robustness under distractors and produces more calibrated refusals than PPO without relevance, with a conservative trade-off that can reduce raw accuracy on some tasks.

Part III: Accessibility and shared practice. We present the *Simple Transformers* library, which standardizes training and evaluation for transformer models across tasks (e.g., dense retrieval, classification, QA) with minimal configuration. The library's adoption across domains underscores its role in lowering the engineering burden and supporting reproducible experimentation.

Across retrieval, reasoning, and refusal, the findings highlight that robustness emerges not from scale alone but from careful design of negatives, data, and rewards. The thesis provides a reproducible foundation for retrieval-augmented systems that generalize broadly and respond with grounded, reliable behavior.

Samenvatting

Het opzoeken van informatie (information retrieval) steunt in toenemende mate op systemen die bewijs moeten vinden, dit bewijs moeten kunnen interpreteren, en moeten afzien van antwoorden wanneer er onvoldoende onderbouwing is. Dit proefschrift onderzoekt hoe dergelijke systemen getraind en geëvalueerd kunnen worden langs drie lijnen: (i) het robuuster maken van dense retrievers onder distributieverschuivingen, (ii) het trainen van kleinere taalmodellen om antwoorden te onderbouwen met bewijs en om te weigeren wanneer dat bewijs ontbreekt, en (iii) het verbeteren van toegankelijkheid en reproduceerbaarheid via open-source hulpmiddelen.

Deel I: Robuuste retrieval onder distributieverschuivingen. Hoofdstuk 2 onderzoekt negatieve sampling voor meertalige dense passage retrieval en introduceert *iterative clustered training* (ICT), een methode die harde negatieven vernieuwt door query- of passage-representaties te clusteren tijdens het trainen. In meertalige experimenten leveren BM25-negatieven de beste prestaties binnen de trainingsdistributie, terwijl ICT, en met name de passage-geclusterde variant (ICT-P), de beste prestaties behaalt buiten de distributie en in zero-shot scenario's, met een lager rekenverbruik dan volledige iteratieve mijnbouw. Hoofdstuk 3 toont bovendien aan dat de samenstelling van trainingsdata belangrijk is: trainen op synthetische datasets met meerdere queries per passage verbetert de generalisatie naar nieuwe domeinen ten opzichte van training met slechts één query per passage, bij verder identieke omstandigheden.

Deel II: Robuuste weigering en bewijs-gebaseerd antwoorden. Hoofdstuk 4 onderzoekt retrieval-augmented vraag-antwoordsystemen met kleine, instructie-getunede taalmodellen onder drie bewijssituaties: (i) *Vanilla* (gouden standaard aanwezig), (ii) *Distractors* (gouden standaard plus irrelevante passages), en (iii) *No-Res* (gouden standaard verwijderd; het juiste gedrag is weigering). Basismodellen neigen ertoe om te vaak te antwoorden in ruisachtige contexten en te weinig te weigeren wanneer bewijs ontbreekt. Hoofdstuk 4 introduceert *reward shaping for robust refusal* (RSRR), een PPO-gebaseerde aanpak die een relevantiebeloning (vergelijking tussen redenering en opgehaalde documenten) combineert met geplande correctheid-, opmaak- en KL-componenten. Over de datasets BioASQ, BeerQA, PubMedQA en StrategyQA verbetert RSRR de robuustheid bij afleidende documenten en zorgt het voor beter gekalibreerde weigeringen dan PPO zonder relevantiebeloning, met een voorzichtige afruil die soms tot lagere ruwe nauwkeurigheid leidt.

Deel III: Toegankelijkheid en gedeelde praktijk. Het proefschrift presenteert de *Simple Transformers*-bibliotheek, die training en evaluatie van transformermodellen voor uiteenlopende taken (zoals dense retrieval, classificatie en het beantwoorden van vragen) standaardiseert met minimale configuratie-inspanning. De brede adoptie van de bibliotheek onderstreept haar rol in het verlagen van de technische drempel en het bevorderen van reproduceerbare experimentatie.

Over retrieval, redenering en weigering heen laat dit werk zien dat robuustheid niet enkel voortkomt uit schaal, maar uit zorgvuldige ontwerpkeuzes in negatieve voor-

beelden, data en beloningen. Het proefschrift biedt zo een reproduceerbare basis voor retrieval-augmented systemen die breed generaliseren en reageren met onderbouwd en betrouwbaar gedrag.