University of Trento

Department of Information Engineering and Computer Science

Doctoral Programme in Information Engineering and Computer Science

$\sim \cdot \sim$

Academic Year 2024–2025

Models and Application of Question Retrieval for Natural Language Processing

Advisor
Alessandro Moschitti – Amazon AGI

Mentor
Ivano Lauriola – Amazon AGI

Ph.D. Candidate
Stefano Campese

Abstract

This thesis investigates the role of question understanding in Question Answering systems, developing methods that exploit question semantic equivalence at progressively larger scales: from individual question pairs, through equivalence clusters, to entire datasets.

The first part addresses question retrieval at scale. We introduce QUADRo, a retrieval framework operating over millions of question-answer pairs, and the Question Ranking Corpus (QRC), a large-scale resource with answer-aware annotations and challenging hard negatives. We demonstrate that incorporating answers during retrieval substantially improves accuracy, as answers serve as a semantic bridge between questions that share little lexical overlap but seek the same information. To reduce annotation costs, we develop Question Ranking Pre-training (QRP), a self-supervised method that learns question equivalence patterns without labeled data, achieving significant improvements while reducing model variance by over 50%.

The second part extends pairwise equivalence to question clusters. We analyze coherence in Large Language Models, finding that a substantial portion of question clusters exhibit incoherent behavior: models answer some phrasings correctly while failing on semantically equivalent alternatives. This reveals that understanding failures, not just knowledge gaps, limit LLM performance. We introduce Question-Augmented Generation (q-RAG), which supplements prompts with retrieved similar questions, improving accuracy by up to 9 percentage points and coherence by up to 28 points. We further show that q-RAG's benefits can be distilled into model parameters through Direct Preference Optimization (DPO) and Supervised Fine-Tuning, producing standalone models with improved coherence that surpass the inference-time approach. For retrieval systems, we apply clusters to train models for consistency: the Coherence Ranking Loss improves ranking coherence by up to 30% while simultaneously improving relevance.

The third part lifts equivalence to the dataset level. We introduce dataset declassification, a framework that replaces proprietary questions with semantically equivalent public alternatives, enabling dataset sharing without exposing sensitive content. Models trained on fully declassified data match baseline performance (WikiQA $\Delta \approx 0$, TrecQA $|\Delta| \leq 1.2$ points), and test set declassification preserves evaluation validity when high-quality mappings exist ($|\Delta| \leq 2$ on standard benchmarks), enabling the release of "shadow benchmarks" for evaluation integrity. We identify boundary conditions through experiments on adversarially-constructed benchmarks.

Together, these contributions show that question semantic equivalence, systematically exploited at multiple scales, enables substantial improvements to QA system accuracy, consistency, and evaluation integrity.

Acknowledgements

This milestone is the result of a long and challenging journey, which would not have been possible without the support of extraordinary people.

I would like to thank the University of Trento and the Department of Information Engineering and Computer Science for providing me with the opportunity to pursue this PhD.

My first thanks go to my advisor, Alessandro Moschitti, for guiding me through these years with expertise and patience, and for giving me the opportunity to grow as a researcher. Alessandro has been both my academic advisor and my manager during my time at Amazon, and I am deeply grateful for his mentorship in both roles. A special thanks to Ivano Lauriola, mentor and friend, who supported me during my time at Amazon and throughout my entire research journey: his help has been invaluable.

I am grateful to my labmates, Luca Di Liello and Matteo Gabburo, with whom I shared not only the work but also the struggles and rewards of this path. A special thought goes to my former colleagues at the Department of Physics and Astronomy of the University of Padova, who cheered me on when I decided to change paths and move to Trento. Thanks also to Federico Agostini, Lorenzo Barbiero, and Ema Baci, colleagues from other labs with whom I had the pleasure of collaborating. And to all the friends who have been by my side over these years.

To Sara, my wife, goes my deepest gratitude. You supported me through the difficult moments, believed in me when I doubted myself, and had the courage to move with me to Los Angeles during my time abroad. This achievement is as much yours as it is mine. And to our daughter Viola Jane, who will join us soon: you are already giving me strength and purpose. This thesis is for you too.

Thanks to my sister Martina, who has always supported me and has always been there when I needed her.

Finally, a thought for my parents, Giancarlo and Anna Maria. Dad, it has been almost ten years since you left us, but your example has guided me every day of this journey. Mom, you passed away just a year ago, and I wish more than anything that you could be here today. This PhD is dedicated to you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Question Answering (QA) is a core task in Natural Language Processing (NLP) that enables machines to understand and respond to human questions. From web search engines to virtual assistants, the ability to match questions with relevant answers drives some of the most widely used applications in computing (Voorhees and Tice, 1999). Recent advances with Large Language Models (LLMs) have dramatically improved QA capabilities (Brown et al., 2020; Touvron et al., 2023), but fundamental challenges remain. Retrieving semantically equivalent questions at scale lacks adequate infrastructure; LLMs and retrieval systems exhibit inconsistent behavior across equivalent phrasings; and benchmark integrity is increasingly compromised by data contamination while dataset privacy limits collaboration.

This thesis investigates these challenges through the lens of *question semantic equivalence*: the relation that holds between questions seeking the same information. We demonstrate that understanding and exploiting this equivalence at multiple scales, from individual question pairs to entire datasets, enables substantial improvements to QA systems.

1.1 Research Gaps and Contributions

This thesis examines core challenges in question understanding and proposes methods to enhance the accuracy, consistency, and reliability of QA systems. We identify four major research gaps:

1. **(Research Gap 1) Insufficient Resources and Methods for Question Retrieval at Scale.** Many QA scenarios require matching user questions against large databases of previously answered questions, a paradigm known as Database-QA (DBQA) (Fader et al., 2014). However, existing resources are insufficient: datasets for question matching are small-scale and domain-specific (Lei et al., 2016; Iyer et al., 2017), lack challenging hard negatives, and do not incorporate answer information into annotations. Creating large-scale resources is prohibitively expensive due to annotation costs, yet self-supervised pre-training methods that could reduce this dependency have not been developed for question ranking. Lexical methods like BM25 (Robertson and Zaragoza, 2009a) fail to capture semantic equivalence across different surface forms, while neural approaches have not been optimized for question-to-question matching. Crucially, the role of pre-computed answers in improving retrieval remains underexplored: answers can serve as a semantic bridge between questions that share little lexical overlap but

seek the same information. This gap is addressed in Chapter 3 (Sections 3.2–3.4) and Chapter 4 (Section 4.1).

2. **(Research Gap 2) Incoherent Behavior of LLMs on Equivalent Questions.** LLMs often produce inconsistent answers when presented with semantically equivalent questions (Elazar et al., 2021; Raj et al., 2023). A model may correctly answer "*What is the capital of France?*" but fail on "*France's capital city is?*" despite possessing the required knowledge. This *coherence failure* is distinct from accuracy failures where the model lacks required information. Such inconsistencies challenge user trust and indicate failures of question *understanding* rather than missing *knowledge*. Current approaches focus primarily on improving accuracy through scaling (Kaplan et al., 2020) or retrieval augmentation (Lewis et al., 2020b), but neglect consistency across equivalent phrasings. This gap is addressed in Chapter 5 (Sections 5.2.2–5.2.3).

3. **(Research Gap 3) Inconsistent Retrieval for Equivalent Queries.** Information retrieval systems exhibit similar coherence problems: equivalent queries often produce substantially different ranked lists (Wang et al., 2021). When a user searches for "*climate change effects*" versus "*impacts of global warming*", they expect similar results, but in this case the retrieval models may return divergent rankings. This inconsistency affects user experience and complicates downstream applications that depend on stable retrieval behavior. While systems to be robust to lexical variations (Campos et al., 2023) and query reformulation techniques (Wang et al., 2021) address related problems, they do not directly optimize for ranking consistency across semantically equivalent queries. No training objectives exist that explicitly encourage retrieval coherence. This gap is addressed in Chapter 5 (Section 5.6).

4. **(Research Gap 4) Benchmark Integrity and Dataset Privacy.** The widespread use of web-scraped training data means that benchmark questions increasingly appear in LLM training corpora, compromising evaluation validity through data contamination (Balloccu et al., 2024). Models may achieve high scores through memorization rather than generalization, making it difficult to assess true capabilities. Additionally, organizations cannot share proprietary QA datasets for research without exposing sensitive content, limiting collaboration and reproducibility. Both problems require methods to transform datasets while preserving their utility: for training sets, the transformed data must yield equivalent model performance; for test sets, it must additionally preserve task difficulty to maintain evaluation validity. Question retrieval offers a potential solution by mapping proprietary questions to semantically equivalent public alternatives, but no systematic framework exists for such transformation, and the conditions under which dataset equivalence holds have not been studied. This gap is addressed in Chapter 6 (Sections 6.2–6.5).

These challenges share a common thread: *question semantic equivalence* at different scales. Gap 1 addresses equivalence between question pairs, developing retrieval methods to find questions seeking the same information. Gaps 2 and 3 extend this to equivalence *clusters*, using groups of equivalent questions to analyze and improve model coherence. Gap 4 lifts equivalence to the *dataset level*, asking whether entire datasets can be replaced with semantically equivalent alternatives. This progression from pairwise to cluster to dataset equivalence provides a unified framework: the retrieval infrastructure of Gap 1 enables the coherence analysis of Gaps 2–3, which in turn informs the declassification methodology of Gap 4.

To address these gaps, we develop: (i) a 6.3M question-answer database with answer-aware retrieval methods and self-supervised pre-training; (ii) question-augmented generation for LLM coherence; (iii) coherence-aware training objectives for retrieval; and (iv) a declassification framework for privacy-preserving dataset transformation.

These methods are detailed in Chapters 3 through 6, with theoretical foundations provided in Chapter 2. Figure 1.1 summarizes this progression and the corresponding methods developed in each chapter.

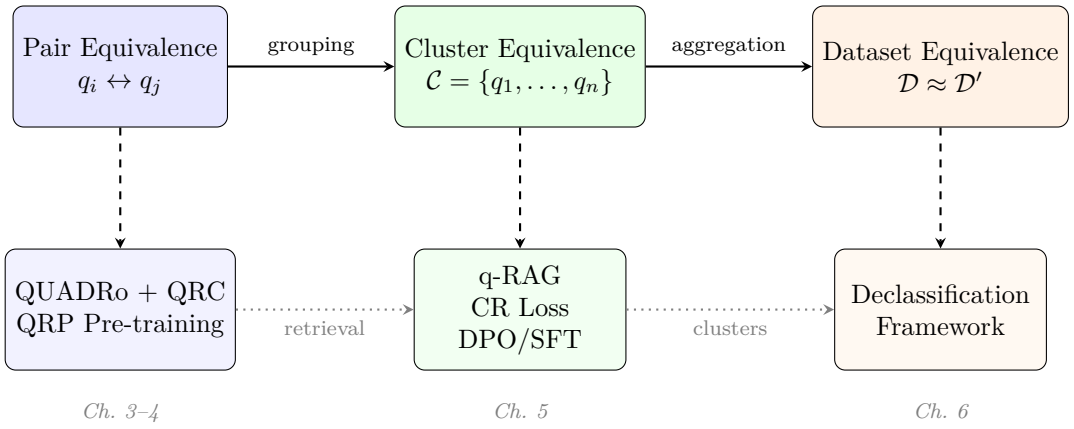

Figure 1.1: Unified view of the thesis contributions. Question semantic equivalence is studied at three scales: pairwise equivalence for retrieval (Chapters 3–4), cluster equivalence for coherence optimization (Chapter 5), and dataset equivalence for privacy-preserving transformation (Chapter 6). Each scale builds on the infrastructure developed at the previous level.

1.1.1 Question Retrieval: Resources and Methods

Retrieving semantically equivalent questions from large databases is essential for DBQA (Fader et al., 2014) and duplicate question detection (Lei et al., 2016). The task requires identifying questions that seek the same information regardless of surface form, enabling systems to reuse previously computed answers rather than generating new responses.

However, existing question matching datasets present significant limitations. Resources like Quora Question Pairs (Iyer et al., 2017) and SemEval duplicate detection tasks (Nakov et al., 2016a) contain only thousands of examples and focus on narrow domains. They typically use random negative sampling, which produces easy negatives that do not challenge models to learn fine-grained semantic distinctions. Furthermore, these datasets treat questions in isolation, ignoring the answer information that could help disambiguate equivalent questions with different surface forms.

The role of answers in question matching deserves particular attention. Two questions may appear dissimilar lexically but become clearly equivalent when their answers are considered. For instance, "*Who painted the Mona Lisa?*" and "*The creator of La Gioconda*" share few words but obviously seek the same information when paired with the answer "*Leonardo da Vinci*". Existing approaches do not systematically exploit this signal.

Finally, while pre-training has revolutionized NLP (Devlin et al., 2019a; Liu et al., 2019), generic pre-training objectives optimize for general language understanding rather than the specific requirements of question ranking. Methods that pre-train specifically for ranking

tasks exist (Di Liello et al., 2022a), but none target the unique characteristics of question-to-question matching.

Our contribution: We introduce **QUADRo** (QUestion-Answer Database Retrieval), a comprehensive framework for question retrieval operating over 6.3 million question-answer pairs aggregated from diverse high-quality sources including GooAQ (Khashabi et al., 2021), WikiAnswer (Fader et al., 2013), and various curated collections (Chapter 3). QUADRo combines dense neural retrieval with cross-encoder reranking in a two-stage architecture, achieving sub-second latency suitable for interactive applications while maintaining high accuracy.

We demonstrate that incorporating answers during retrieval substantially improves performance, with gains of +5 P@1 over answer-agnostic methods. We show that neural approaches are essential for semantic matching, outperforming BM25 by 19%. We analyze the effects of candidate ordering during training, finding that presenting candidates in query-answer-question (QAQ) order brings to optimal results by encouraging models to use answer information as a bridge between queries and candidates.

We also introduce the **Question Ranking Corpus (QRC)**, containing 15211 queries with approximately 443K annotated examples. QRC addresses limitations of existing datasets: it provides substantially larger scale, includes challenging hard negatives sampled from dense retrieval (ensuring that models must learn fine-grained distinctions), incorporates answer-aware relevance judgments, and covers open-domain factual questions rather than narrow domains.

To reduce dependence on manual annotation, we develop **Question Ranking Pre-training (QRP)**, a self-supervised method that trains models to detect ranking corruptions without access to the original query (Chapter 4). By withholding the query, QRP forces models to learn from answer-mediated relationships between questions, developing abstract patterns of equivalence that transfer to new queries. QRP achieves statistically significant improvements (+1.05% P@1, $p = 0.0005$) while reducing model variance by over 50%, improving reliability of model selection.

1.1.2 Coherence of Large Language Models

While LLMs achieve impressive accuracy on QA benchmarks (Brown et al., 2020; Touvron et al., 2023), their behavior on semantically equivalent questions reveals significant inconsistencies. Recent work has documented that models often change their predictions when questions are paraphrased (Rabinovich et al., 2023), but comprehensive analysis of this phenomenon on factual QA and methods for addressing it remain limited.

We distinguish between two types of failures. *Accuracy failures* occur when a model produces incorrect answers due to missing knowledge. These may be addressed by improving training data or retrieval augmentation. *Coherence failures* occur when a model produces inconsistent answers to equivalent questions, even when some answers are correct. A model exhibiting coherence failures possesses the required knowledge but fails to access it reliably across different phrasings. This distinction has important implications: improving coherence requires enhancing question *understanding*, not expanding factual *knowledge*.

Existing approaches to improving LLM reliability focus primarily on scaling model size (Kaplan et al., 2020), improving training data quality, or augmenting generation with retrieved documents (Lewis et al., 2020b). However, these approaches target accuracy rather than consistency. Retrieval-Augmented Generation (RAG) provides new factual information but

does not address failures where the model already possesses relevant knowledge but fails to access it due to question misunderstanding.

Our contribution: We provide the first comprehensive analysis of **LLM coherence** on factual QA (Chapter 5). Using clusters of equivalent questions constructed from paraphrase datasets and question retrieval, we evaluate whether models produce consistent answers across phrasings. Our analysis spans multiple model families (Phi, LLaMA, Mistral) and sizes, revealing significant coherence gaps where models succeed on some phrasings but fail on semantically equivalent alternatives. These failures persist even in the largest models, suggesting that scaling alone does not solve the coherence problem.

We analyze factors influencing coherence, finding that question difficulty is the strongest predictor: models are less coherent on harder questions. We also find that coherence failures often occur on questions where the model demonstrates knowledge through some phrasings but not others, confirming that understanding rather than knowledge is the bottleneck.

To address coherence gaps, we introduce **Question-Augmented Generation (q-RAG)**, a novel retrieval-augmented approach that supplements LLM prompts with clusters of similar questions retrieved from a large database. Unlike traditional RAG that retrieves documents containing new factual information, q-RAG retrieves questions expressing equivalent or related information needs. This provides redundant semantic signal about user intent without introducing potentially distracting factual content.

The intuition is that multiple phrasings of a question collectively constrain its interpretation more than any single phrasing alone. When a model sees not only "*How many calories are in a cucumber?*" but also "*Cucumber calorie content?*" and "*What is the nutritional energy value of cucumbers?*", it receives converging evidence about the intended information need.

Empirically, q-RAG improves accuracy up to 9 percentage points and coherence up to 28 points across multiple LLMs. Remarkably, q-RAG outperforms document-based RAG despite providing no new factual information, demonstrating that understanding failures rather than knowledge gaps often limit LLM performance.

We extend this analysis to **multilingual settings**, examining whether coherence patterns transfer across languages. The analysis reveals both universal patterns: (i) coherence decreases with question difficulty across all languages, and (ii) language-specific effects as coherence varies with resource availability and typological features. These findings inform development of multilingual QA systems and highlight the need for language-aware coherence optimization.

1.1.3 Coherence of Retrieval Systems

Information retrieval systems face analogous coherence challenges. When users issue semantically equivalent queries, they expect consistent results, but retrieval models often produce substantially different rankings (Bernard et al., 2007; Jansen et al., 2005). This inconsistency has practical consequences: users reformulating unsuccessful queries may receive entirely different results rather than refined versions, and downstream systems that aggregate multiple query variants may encounter contradictory evidence.

Traditional approaches address related but distinct problems. Query expansion techniques (Carpineto and Romano, 2012) add terms to improve recall but do not ensure consistency across equivalent formulations. Query reformulation methods (Wang et al., 2021) transform queries to improve retrieval but optimize for relevance rather than consistency. Robust retrieval research (Voorhees, 2006) focuses on maintaining performance across diverse queries

rather than consistency across equivalent ones.

The fundamental issue is that standard retrieval training objectives optimize only for relevance: given a query, rank relevant documents above non-relevant ones. These objectives provide no signal about consistency across queries. A model can achieve perfect relevance scores while producing completely different rankings for equivalent queries, as long as relevant documents appear near the top in each case.

Our contribution: We introduce the **Coherence Ranking (CR) Loss** for training document retrieval models that produce consistent rankings for semantically equivalent queries (Chapter 5). The loss combines two complementary components:

Query Embedding Alignment (QEA) encourages the model to produce similar embeddings for equivalent queries. By minimizing the distance between query representations, models learn to map equivalent surface forms to similar points in embedding space, promoting consistent downstream rankings.

Similarity Margin Consistency (SMC) directly optimizes for consistent relevance scores across query variants. For each document, SMC encourages the model to assign similar relevance scores regardless of which equivalent query is used, penalizing cases where the same document receives high relevance for one phrasing but low relevance for another.

Together, these components improve coherence by up to 30% (measured by Rank-Biased Overlap between rankings for equivalent queries) while simultaneously improving relevance by up to 1.69% NDCG. This demonstrates that coherence and accuracy are complementary rather than competing objectives: training for consistency provides beneficial regularization that improves overall retrieval quality.

We also show that improved coherence increases reranking opportunity by 9.3%. When initial rankings are more consistent, reranking models see more relevant candidates across query variants, enabling better final performance.

1.1.4 Dataset Equivalence and Declassification

The reliability of QA evaluation depends on the integrity of benchmark datasets. However, the widespread use of web-scraped training data has led to increasing *data contamination*: benchmark questions, answers, or discussions about them appear in training corpora (Balloccu et al., 2024). When this occurs, models may achieve high benchmark scores through memorization rather than genuine question understanding, compromising our ability to assess and compare system capabilities.

A related challenge concerns *dataset privacy*. Organizations developing QA systems often create proprietary datasets reflecting their specific domains, user populations, or annotation guidelines. Sharing these datasets could advance research but may expose sensitive content, violate user privacy, or reveal competitive information. Current practice forces a binary choice between complete sharing and complete secrecy.

Both problems can be addressed through *dataset declassification*: replacing original questions with semantically equivalent public alternatives. For privacy, organizations release the declassified version in place of the original, enabling collaboration without exposing sensitive content. For evaluation integrity, organizations release declassified versions publicly as "shadow benchmarks" while retaining originals for authoritative assessment, ensuring that models that memorize the public version gain no advantage on the protected original. In both cases, the declassified data must preserve utility: training sets must yield equivalent model performance, while test sets must additionally preserve task difficulty.

Our contribution: We introduce a **declassification framework** that enables privacy-preserving dataset transformation by replacing proprietary questions with semantically equivalent public alternatives (Chapter 6). The framework operates in two stages: **question mapping** uses QUADRo to find equivalent public questions, while **answer reconstruction** uses domain-matched corpora to generate appropriate answers for the mapped questions.

We validate the framework across multiple QA paradigms and four LLMs of varying scale. For training set declassification, models trained on fully declassified data match baseline performance: WikiQA achieves $\Delta \approx 0$, TrecQA shows $|\Delta| \leq 1.2$ points. For test set evaluation, OpenBookQA declassification preserves model rankings with $|\Delta| \leq 0.4$ points across all models.

The framework reveals important boundary conditions through experiments on SimpleQA, a benchmark deliberately designed with obscure questions. Declassification shows +5.8 to +12 point difficulty shifts, indicating that mapped questions are substantially easier than originals. However, a backtranslation baseline (Gen-BT) demonstrates that this shift stems from mapping to different questions, not from surface-level rephrasing: Gen-BT achieves $|\Delta| \leq 4.1$ while preserving semantic content. Stratified analysis further shows that *cross-encoder reranking is essential* for difficulty preservation: in the top 1% by reranker similarity, three of four models achieve $|\Delta| \leq 2.3$, while bi-encoder retrieval alone fails even at highest similarity.

We also find that *domain matching* is critical for answer reconstruction. When replacement answers come from domains similar to the original dataset (Wikipedia for WikiQA, CCnews for TrecQA), performance is maintained. Domain mismatch causes degradation regardless of question quality.

1.2 Publications

The research presented in this thesis has resulted in the following publications, listed in chronological order:

- ***QUADRo: Dataset and Models for QUestion-Answer Database Retrieval***
  Stefano Campese, Ivano Lauriola, and Alessandro Moschitti
  Findings of the Association for Computational Linguistics (EMNLP 2023)

- ***Pre-Training Methods for Question Reranking***
  Stefano Campese, Ivano Lauriola, and Alessandro Moschitti
  Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2024)

- ***Analyzing and Improving Coherence of Large Language Models in Question Answering***
  Ivano Lauriola, Stefano Campese, and Alessandro Moschitti
  Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2025)

- ***Improving Document Retrieval Coherence for Semantically Equivalent Queries***
  Stefano Campese, Ivano Lauriola, and Alessandro Moschitti
  Proceedings of the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (AACL 2025)

- ***Exploring Coherence of LLMs in Multilingual Question Answering***
  Stefano Campese, Ivano Lauriola
  Under review GEM Workshop at ACL 2026 (GEM 2026)

- ***From Proprietary to Public: Dataset Declassification for Question Answering***
  Stefano Campese, Alessandro Moschitti, and Ivano Lauriola
  Under review at Findings of the Association for Computational Linguistics (EMNLP 2026)

Additionally, during the doctoral period the candidate contributed to the following related publications:

- ***Datasets for Multilingual Answer Sentence Selection***
  Matteo Gabburo, Stefano Campese, Federico Agostini, and Alessandro Moschitti
  Findings of the Association for Computational Linguistics (EMNLP 2024)

- ***Domain-Specific and Cross-Lingual Synthetic Data Generation for Information Retrieval Training in RAG Applications***
  Lorenzo Barbiero, Federico Agostini, Ema Baci, Federico Frigo, Manuel Vianello, Davide Pozza, and Stefano Campese
  Proceedings of the 11th Intelligent Systems Conference (IntelliSys 2025)
  Best Student Paper Award

1.3 Thesis Structure

This thesis is structured into seven chapters. The first two chapters provide essential context and background, while Chapters 3 through 6 present the core contributions addressing the research gaps identified in Section 1.1.

Chapter 2: Background provides the technical foundations necessary for understanding the contributions of this thesis, covering Question Answering paradigms, neural retrieval methods, transformer architectures, and evaluation metrics.

Chapter 3: Question Retrieval at Scale introduces QUADRo, a comprehensive framework for question retrieval operating over 6.3 million question-answer pairs, and the Question Ranking Corpus (QRC), demonstrating the value of answer-aware retrieval with gains of +5 P@1 over answer-agnostic methods. This chapter is based on work published at EMNLP 2023.

Chapter 4: Specialized Pre-Training for Question Ranking introduces Question Ranking Pre-training (QRP), a self-supervised method that improves retrieval accuracy (+1.05% P@1) while reducing model variance by over 50%, decreasing annotation dependency. This chapter is based on work published at EACL 2024.

Chapter 5: Question Clustering for Model Coherence extends equivalence to clusters for analyzing and improving system coherence. We analyze LLM coherence gaps on factual QA, introduce Question-Augmented Generation (q-RAG) which improves accuracy by up to 9 points and coherence up to 28 points, extend the analysis to multilingual settings, and propose the Coherence Ranking Loss for training retrieval models. This chapter synthesizes work published at NAACL 2025 and AACL 2025, with additional contributions under review at GEM Workshop at ACL 2026.

Chapter 6: Dataset Equivalence and Declassification extends equivalence to the dataset level, introducing a framework for privacy-preserving dataset transformation. We demonstrate that models trained on declassified data match baseline performance (WikiQA $\Delta \approx 0$, TrecQA $|\Delta| \leq 1.2$ points), and that test set declassification preserves model rankings (OpenBookQA $|\Delta| \leq 0.4$ points). The framework enables shadow benchmarks for evaluation integrity, secure dataset sharing, and regulatory compliance. This chapter presents work under review at EMNLP 2026.

Chapter 7 Conclusions summarizes the contributions of the thesis, discusses unifying themes including the distinction between knowledge and understanding failures, acknowledges limitations, and outlines directions for future research.

Chapter 2

Background and Related Work

This chapter provides a comprehensive overview of the foundational concepts, methods, and prior work that is fundamental to this thesis. We begin with an examination of Question Answering (QA) systems, tracing their evolution from early rule-based approaches to modern neural architectures. We then explore question understanding and semantic similarity, which are essential for identifying equivalent questions, followed by a detailed analysis of Transformer models and dense retrieval methods that are at the basis of efficient large-scale question-answer matching. Given the central role of Database-based QA (DBQA) systems in this thesis, we dedicate substantial attention to Frequently Asked Question (FAQ) retrieval architectures and question ranking methodologies. We further examine Retrieval-Augmented Generation (RAG), which grounds Large Language Model (LLM) outputs in retrieved evidence, and analyze critical challenges in LLM coherence and robustness that motivate our clustering-based approaches. We then discuss question clustering techniques and their applications. Given the importance of dataset transformation in Chapter 6, we review privacy-preserving approaches to Natural Language Processing (NLP) data release, including anonymization, synthetic data generation, and benchmark contamination concerns. The chapter concludes with evaluation metrics essential for the experimental validation in subsequent chapters.

2.1 Question Answering Systems

QA is a fundamental task in NLP that aims to automatically provide accurate answers to questions posed in natural language. The field has evolved significantly over the past five decades, from early rule-based systems operating on structured databases to sophisticated neural architectures capable of reasoning over unstructured text and generating fluent responses (Farea and Emmert-Streib, 2025).

2.1.1 Historical Evolution

The history of QA systems dates back to the 1960s with pioneering systems like BASEBALL (Green et al., 1961) and LUNAR (Woods, 1973), which respectively answered questions about baseball statistics and moon rock samples. These early systems operated on structured databases based on built-in patterns to map natural language questions to database queries. While effective within their narrow domains, they required extensive manual effort to develop and lacked the ability to generalize to new domains or deal with linguistic variation.

The 1970s and 1980s saw continued development of knowledge-based QA systems, such as SHRDLU (Winograd, 1972), which could answer questions about a blocks world by reasoning over structured knowledge representations. These systems demonstrated advanced reasoning capabilities but were still limited to carefully and specifically controlled environments.

The 1990s marked a paradigm shift to open-domain QA, driven mainly by the TREC QA track (Voorhees and Tice, 1999) which provided standardized evaluation and guided systematic comparison of approaches. Systems in this era adopted an Information Retrieval (IR) paradigm: first, retrieve relevant documents from large corpora using keyword-based methods, then extract answers using pattern matching, named entity recognition, or shallow parsing (Moldovan et al., 2003). With the introduction of the retrieve-then-extract approach, systems were able to scale to open domains for the first time, though they remained heavily dependent on surface-level features and still struggled with questions requiring complex reasoning or inference.

The 2000s saw significant advances in structural approaches to question answering. Kernel methods allowed comparison of questions and answers based on their syntactic and semantic structures rather than solely on surface features. Convolution kernels over predicate-argument structures (Moschitti, 2004) captured richer semantic relationships beyond lexical overlap, while relational kernels (Moschitti and Zanzotto, 2007) introduced a framework for jointly modeling question-answer relationships within a kernel representation, foreshadowing the idea of joint modeling later adopted by neural cross-encoders. Work on structured lexical similarity further demonstrated that tree kernels over dependency structures could effectively capture semantic equivalence and entailment (Croce et al., 2011). Elements of these structural approaches were incorporated at scale in IBM Watson (Ferrucci et al., 2010), which combined multiple structural matching techniques with statistical ranking to compete with humans on open-domain QA. The transition to neural methods began with convolutional neural networks for answer sentence selection (Severyn and Moschitti, 2015), which learned to extract relevant features automatically rather than relying on hand-crafted kernels.

The introduction of large-scale reading comprehension datasets, such as SQuAD (Rajpurkar et al., 2016), revolutionized QA research by providing sufficient training data for neural approaches. Models like BiDAF (Seo et al., 2017) combined recurrent neural networks with attention mechanisms to identify answer spans within provided passages, while DrQA (Chen et al., 2017) demonstrated how to scale these approaches to open-domain settings by combining document retrieval with neural reading comprehension. The advent of pre-trained Transformers, such as BERT (Devlin et al., 2019b), further transformed the field, enabling models to achieve near-human performance on reading comprehension benchmarks (Rajpurkar et al., 2016). Subsequent work introduced increasingly challenging benchmarks: (i) requiring multi-hop reasoning across multiple documents like HotpotQA (Yang et al., 2018), (ii) handling diverse answer types including long-form responses such as Natural Questions (NQ) (Kwiatkowski et al., 2019; Garg et al., 2020), and (iii) maintaining coherence across conversational exchanges like CoQA (Reddy et al., 2019a).

2.1.2 Modern QA Paradigms

Modern QA systems can be broadly categorized into several paradigms, each with distinct architectural choices, training requirements, and application scenarios. Understanding these paradigms is essential for positioning the contributions of this thesis within the broader landscape.

Extractive QA. Extractive systems identify answer spans directly from provided passages, treating QA as a span prediction task (Rajpurkar et al., 2016; Kwiatkowski et al., 2019). Given a question and a context passage, the model produces two probability distributions over token positions: one for the answer start and one for the answer end. The answer is extracted as the substring between the most likely start and end positions, subject to validity constraints (start before end, reasonable length). Transformer-based models like BERT process question and passage as a single concatenated sequence, using special tokens to delimit boundaries, and apply linear classification heads over the final hidden states to predict span boundaries. While extractive QA offers interpretability through explicit source attribution and avoids hallucination by construction, it is fundamentally limited to answers that appear verbatim in the provided context and cannot synthesize information across multiple passages, such as in the case of complex questions (Gabburo et al., 2024b), or generate novel phrasings.

Generative QA. Rather than extracting verbatim spans, generative QA systems produce free-form answers by conditioning on the question and provided context. This approach treats reading comprehension as a sequence-to-sequence task: given question and passage, generate an appropriate answer. Early work employed encoder-decoder architectures such as T5 (Raffel et al., 2020a) and BART (Lewis et al., 2020a), but modern decoder-only models including GPT (Achiam et al., 2023) and Claude (Anthropic, 2024) perform this task effectively through in-context learning, where question and passage are formatted as a prompt. Generative QA offers flexibility that extraction cannot: answers can synthesize information from multiple sentences, rephrase verbose content concisely, handle yes/no questions naturally, or perform simple reasoning over the passage. However, generation can introduce the risk of hallucination, producing fluent but unsupported content, and reduces interpretability since answers cannot be directly traced to source spans. Generative QA with provided context differs from closed-book QA, which is generation from parametric memory alone, and RAG which adds retrieval to select the context as it represents the reading and answering component that these other paradigms build upon.

Open-book QA. While the paradigms above assume a context passage is provided, open-book or open-domain QA removes this assumption, requiring systems to find relevant information from large corpora before answering. This *retrieve-then-read* approach (Chen et al., 2017) decomposes the problem into two stages: a retriever searches millions or billions of documents to identify potentially relevant passages, then a reader, which may be extractive or generative, processes the retrieved content to produce an answer. The retrieval component typically employs BM25 (Robertson and Zaragoza, 2009b) for efficiency or dense methods like DPR (Karpukhin et al., 2020) for semantic matching. The main challenge in open-book QA is the error propagation of the retriever-reader pipeline: if relevant documents are not retrieved, even a perfect reader cannot produce correct answers. Recent work addresses this through iterative retrieval, where the reader's partial understanding guides additional retrieval rounds, and through end-to-end training that jointly optimizes retriever and reader (Ni et al., 2019).

Closed-book QA. These systems rely entirely on knowledge encoded in model parameters during pre-training, answering questions without accessing external documents at inference time (Roberts et al., 2020). The approach treats the language model itself as a knowledge base, querying it through natural language prompts. Large pre-trained models like GPT (Achiam et al., 2023), LLaMA (Touvron et al., 2023), and Mistral (Jiang et al., 2023b) demonstrate remarkable factual recall, particularly for frequently occurring facts in pre-training data. However, closed-book QA faces fundamental limitations: knowledge cur-

rency degrades as real-world information changes after training cutoffs. Additionally, models may hallucinate plausible but incorrect information with high confidence; there is no mechanism for source attribution or verification; and performance correlates strongly with fact frequency in training data, disadvantaging rare or specialized knowledge. These limitations motivate retrieval augmentation approaches that ground generation in external evidence.

Knowledge Base QA (KBQA). KBQA systems answer questions by querying structured knowledge bases such as Freebase or Wikidata (Berant et al., 2013; Yih et al., 2015). The core challenge is *semantic parsing*: converting natural language questions into formal query languages like SPARQL that can be executed against the knowledge graph. This requires: (i) mapping natural language expressions to entities, known as entity linking, (ii) identifying the relations being queried, which are the relation detection, and (iii) composing these into valid graph queries that respect the schema. Neural approaches learn to generate query graphs or logical forms directly from question text, while retrieval-based methods identify candidate subgraphs and rank them (Bordes et al., 2014; Yih et al., 2015). KBQA offers precise answers grounded in curated, structured facts with clear provenance, but is inherently limited to information explicitly represented in the knowledge base and struggles with questions requiring inference, aggregation, or common-sense reasoning beyond the graph structure.

Conversational QA. These systems answer questions within multi-turn dialogues. The main challenge in this approach is maintaining conversation history and resolving dependencies across different turns of the conversation (Choi et al., 2018; Reddy et al., 2019b). Differently from single-turn QA where each question is self-contained, conversational questions build on previous turns. Consider a dialogue about Marie Curie: after answering "*When was she born?*" with "*1867*," a follow-up "*And where?*" requires recognizing that "where" refers to her birthplace and that "she" still denotes Marie Curie. Such questions involve coreference (e.g., "she," "her husband"), ellipsis, where key elements of the question, such as the subject or predicate, are omitted because they are recoverable from prior context (e.g., "*And in 1990?*" implicitly meaning "*What happened to her in 1990?*"), as well as topic shifts that require discourse tracking. Models must therefore maintain representations of the dialogue history through concatenation, hierarchical attention, or explicit memory mechanisms. Due to the sequential nature of conversational QA also introduces evaluation challenges: an incorrect answer early in a conversation can propagate errors to the end of the conversation.

2.1.3 Answer Sentence Selection

The reading component in open-book QA must identify relevant content from retrieved candidates, a capability studied extensively as a standalone task. Answer Sentence Selection (AS2) is a ranking problem: given a question q and candidate set $C = \{c_1, c_2, \ldots, c_n\}$, the system must rank answer-bearing candidates above non-answers (Wang et al., 2007; Yang et al., 2015). Unlike extractive QA which locates precise spans within a given passage, AS2 operates over discrete candidate units and produces a ranking rather than an extraction. Formally, AS2 models learn a scoring function $s(q, c)$ that assigns higher scores to relevant candidates. The training objectives include pointwise cross-entropy, pairwise margin losses, and listwise ranking losses, with evaluation using Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision@K (P@K) (Moschitti, 2006; Severyn and Moschitti, 2015). This formulation naturally fits many real-world scenarios including selecting relevant passages from search results, identifying answer-bearing sentences in documents, and ranking pre-computed

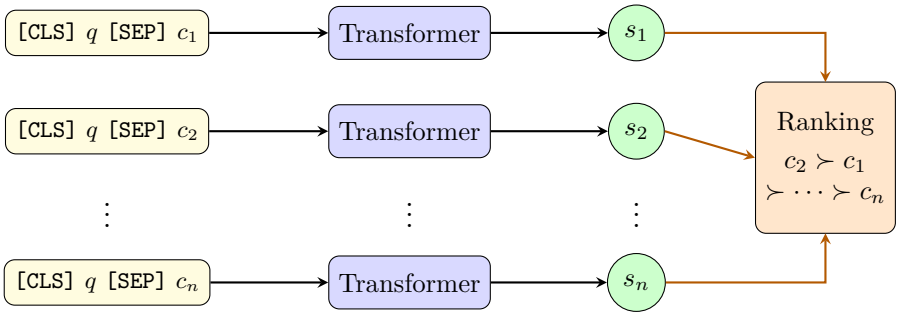

Figure 2.1: Cross-encoder architecture for Answer Sentence Selection. Each question-candidate pair (q, c_i) is concatenated and jointly encoded by a shared Transformer, producing relevance scores s_i used to rank candidates.

responses in FAQ systems.

Early AS2 systems relied on feature engineering combining lexical overlap, syntactic structure, and semantic similarity. Tree kernel methods proved particularly effective by measuring similarity through structural matching over parse trees (Moschitti, 2006; Severyn and Moschitti, 2013), capturing that "*Who invented the telephone?*" matches "*Bell invented...*" through structural correspondence rather than lexical overlap. Deep learning transformed the field by enabling end-to-end learning: convolutional architectures captured local patterns (Severyn and Moschitti, 2015), attention mechanisms focused on relevant candidate portions (Tan et al., 2016), and Siamese (Karpukhin et al., 2020) networks projected questions and candidates into shared embedding spaces.

Pre-trained Transformers brought substantial improvements. Cross-encoders jointly encode question-candidate pairs through self-attention, achieving high accuracy but requiring $O(n)$ forward passes for n candidates (Garg et al., 2019). Bi-encoders address efficiency by independently encoding questions and candidates into fixed embeddings, enabling candidate pre-computation and sub-linear retrieval (Reimers and Gurevych, 2019a). This accuracy-efficiency trade-off motivates two-stage pipelines where bi-encoders retrieve top-k candidates efficiently, then cross-encoders rerank this smaller set accurately. The contextual extension of the AS2 task demonstrates that incorporating surrounding sentences from the source document significantly improves ranking, particularly for candidates requiring coreference resolution (Lauriola and Moschitti, 2021a,b).

AS2 architectures, visible in Figure 2.1, directly inform the question ranking approaches in this thesis. Question ranking in DBQA (Chapter 3) shares the retrieve-then-rerank structure, and the pre-training methodology in Chapter 4 extends ideas from AS2 to question-question similarity. The key difference is that while AS2 ranks answer candidates given a question, question ranking evaluates similarity between questions themselves, but the architectural patterns and efficiency considerations transfer directly.

2.1.4 Database-based QA and FAQ Systems

Database-based Question Answering (DBQA) retrieves answers from pre-computed collections of question-answer pairs rather than generating them from scratch or extracting them from documents as depicted in Figure 2.2. The main idea is that many user questions have been asked before, possibly in different formulations. In this scenario, by finding a semantically equivalent question in the database, systems can return its associated answer directly. This

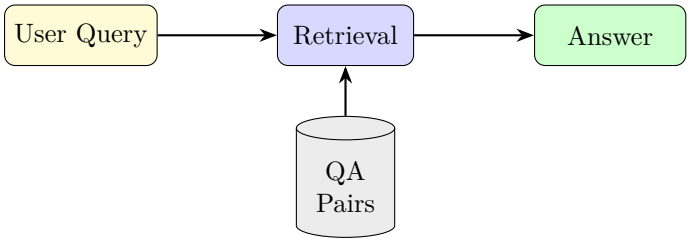

Figure 2.2: Database-based QA: a user query is matched against stored question-answer pairs, and the answer from the most similar entry is returned.

paradigm offers different advantages: guaranteed factual answers from curated sources, low inference latency, easy knowledge updates by modifying the database, and transparent source attribution.

Frequently Asked Questions (FAQ) systems represent the earliest implementations of this paradigm (Burke et al., 1997). Traditional FAQ retrieval relied on keyword matching and similarity heuristics (Sneiders, 2002), approaches that struggled with linguistic variability when users phrased questions differently from stored entries. Modern systems leverage neural retrieval models that encode queries and stored questions into dense vector representations, enabling semantic matching that transcends lexical overlap (Sakata et al., 2019; Mass et al., 2020). This evolution from lexical to semantic matching dramatically improved robustness to paraphrase and linguistic variation.

Question Similarity. A foundational capability for DBQA is recognizing when two questions are semantically equivalent, enabling retrieval of stored answers for questions phrased differently from database entries. This task, known as Duplicate Question Detection (DQD), has evolved from lexical and syntactic approaches through neural embeddings to Transformer-based methods (see Section 2.2 for detailed background). The progression from keyword matching to semantic representations dramatically improved robustness to paraphrase and linguistic variation.

Scaling to Large Databases. While FAQ systems typically contain hundreds or thousands of curated entries, modern DBQA systems query automatically constructed databases containing millions of QA pairs extracted or generated from text corpora (Lewis et al., 2021). This scale enables broader coverage but introduces challenges in retrieval efficiency, ranking accuracy, and quality control.

A basic DBQA pipeline consists of two stages. The *retrieval* stage uses sparse methods like BM25 or dense bi-encoders to find the top-k most similar questions in the database. Traditional approaches encode only questions, computing similarity between the user query and stored questions without considering answers. The *ranking* stage optionally reranks retrieved candidates using more expensive models, typically cross-encoders that jointly encode query-question pairs.

Early work on DBQA for forums and FAQ systems (Nakov et al., 2016a; Shen et al., 2017; Hoogeveen et al., 2015) pointed out that when answers are available together with questions, the resulting systems can be very accurate. However, most practical applications were confined to domain-specific settings with limited q/a pair availability. Othman et al. (2019) introduced WEKOS for FAQ retrieval using k-means clustering and word embeddings. After the rise of Transformers, Mass et al. (2020) proposed ensemble systems combining BM25 with BERT, also exploring GPT-2 for generating question paraphrases to augment

low-resource FAQ datasets. Sakata et al. (2019) combined BERT with TSUBAKI, an efficient BM25-based retrieval architecture, to retrieve similar questions from FAQ databases. More recently, Lewis et al. (2021) assembled PAQ, a database of 65 million automatically generated QA pairs, demonstrating that DBQA pipelines (RePAQ) can achieve competitive accuracy with substantially lower latency than generative approaches. However, PAQ contains considerable noise (estimated 18% incorrect pairs) and relies on generated questions that can be unnatural. Seonwoo et al. (2022) proposed two-step retrieval combining BM25 and DPR for improved efficiency, though these systems still rely solely on question similarity without leveraging answers.

The Role of Answers in Retrieval. Although early work noted the potential benefit of answers, their systematic utilization remained poorly explored in DBQA applications. A critical insight is that answers provide essential context for determining question equivalence (Wang et al., 2020b). For instance, "*Can a cat and a dog get along?*" and "*Do cats like the company of dogs?*" appear lexically different, but their shared answer confirms they ask the same thing. Chapter 3 addresses this gap by demonstrating that incorporating answers during both retrieval and ranking substantially improves accuracy (Campese et al., 2023). The approach encodes question-answer pairs jointly in the retriever ([CLS] q_i [SEP] a_i [EOS]) rather than questions alone, and includes answers in the reranker input as disambiguating context.

Existing Resources and Limitations. Several datasets have been developed for question similarity and DBQA. QuoraQP contains 404290 question pairs annotated for semantic equivalence, though without answers for most pairs; Wang et al. (2020b) released an extension with answers extracted from Quora threads. CQADupStack (Hoogeveen et al., 2015) provides questions from twelve StackExchange subforums with duplicate annotations, but contains only ≈5% duplicates and limited answer coverage. WikiAnswers clusters over 30 million questions into paraphrase groups with an average of 25 questions per cluster, but associated answers are long paragraphs unsuitable for DBQA. SemEval-2016 Task 3 (Nakov et al., 2016a) introduced community QA annotations for question-comment and question-question similarity, but is limited in scale and domain-specific.

These resources share common limitations: most do not include answers, quality guarantees are lacking, and they are structured as question pairs rather than question ranking scenarios. This prevents studying and training the full retrieval-reranking pipeline essential for DBQA. To address these gaps, Chapter 3 introduces the Question Ranking Corpus (QRC), comprising 15211 queries each paired with 30 candidate QA pairs annotated for semantic equivalence. Crucially, annotators were shown answers during annotation, reducing annotation error by 45% compared to question-only annotation. The resulting 443000 annotated examples enable systematic study of retrieval and ranking for DBQA.

Advantages and Limitations. DBQA offers several advantages as a QA paradigm. For example, compared to generative approaches, it provides guaranteed factual answers from curated databases, eliminating hallucination by construction and enabling transparent source attribution. In similar way, compared to retrieval-augmented generation, it offers lower inference latency since answers are returned directly rather than generated, with retrieval typically completing in tens of milliseconds even for databases containing millions of entries. Knowledge can be updated by modifying the database without model retraining. However, DBQA systems face inherent limitations: they cannot answer questions outside database coverage, they require robust semantic matching to handle question phrasing variations, and databases

may become outdated if not regularly maintained. The coverage limitation is fundamental, as even databases containing millions of pairs cannot cover the long tail of possible questions.

2.1.5 Retrieval-Augmented Generation

Large language models (LLMs) can answer questions directly from their parametric memory, but this closed-book approach faces significant limitations: knowledge becomes outdated as the world changes after training, models may hallucinate plausible but incorrect information with high confidence, and generated answers lack verifiable source attribution (Schimanski et al., 2024). These limitations are particularly problematic for applications requiring factual accuracy and accountability.

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020b) addresses these limitations by augmenting language models with external knowledge retrieved at inference time. Rather than relying solely on parametric memory, RAG systems first retrieve relevant documents from a knowledge corpus, then condition generation on both the question and retrieved context. However, the original RAG system had notable limitations: it relied on a Wikipedia passage index with DPR retrieval, used no reranking stage, and was primarily evaluated on factoid QA where generation reduces to entity prediction, making the system less sensitive to retrieval noise.

The basic RAG pipeline, visible in Figure 2.3, operates in two stages. Given a question, the retriever identifies relevant passages from a document corpus using dense or sparse retrieval methods. These passages are then concatenated with the question and provided as context to a language model, which generates an answer grounded in the retrieved evidence. The retriever and generator can be trained jointly or separately, with various architectural choices affecting the trade-off between efficiency and accuracy.

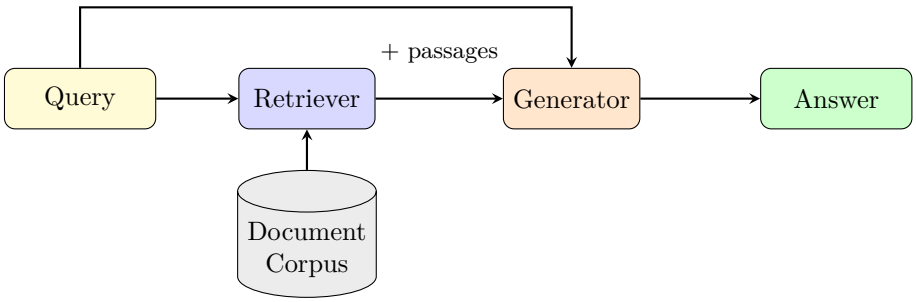

Figure 2.3: Retrieval-Augmented Generation: a query retrieves relevant passages from a document corpus, then both query and passages are provided to a generator that produces the answer.

Several influential RAG variants have emerged to address different aspects of the retrieve-generate pipeline. Fusion-in-Decoder (FiD) (Izacard and Grave, 2021) independently encodes each retrieved passage with the question, then fuses representations in the decoder, enabling efficient scaling to many documents; however, it still lacked reranking and focused on factoid QA. Re2G (Glass et al., 2022) introduced cross-encoder reranking to reduce noise in retrieved passages, achieving significant improvements, though still primarily in factoid settings.

The first RAG system to address non-factoid QA with a complete industrial pipeline was GenQA (Hsu et al., 2021), which combined a real web-scale search engine (approximately 10 billion documents) with state-of-the-art passage and sentence rerankers, enabling generation of long-form answers rather than simple entity extraction. This demonstrated that high-quality

retrieval and reranking are essential for RAG systems generating complex answers. Subsequent work explored training strategies for RAG: Gabburo et al. (Gabburo et al., 2022) showed that reranker scores can supervise generator training through knowledge distillation, conceptually similar to RLHF approaches, while follow-up work (Gabburo et al., 2023) introduced reward functions based on automatic QA evaluators for more sophisticated training signals.

REALM (Guu et al., 2020) integrates retrieval directly into pre-training, learning to retrieve documents that improve masked language modeling performance. RETRO (Borgeaud et al., 2022) augments language model pre-training with retrieval at the chunk level, retrieving relevant text chunks for each segment of the input. Atlas (Izacard et al., 2023) fine-tunes both retriever and generator end-to-end with careful attention to training stability.

Modern RAG systems have evolved beyond simple document retrieval to address increasingly sophisticated scenarios. Graph-enhanced RAG leverages structured knowledge graphs alongside unstructured text. Temporal-aware retrieval handles time-sensitive queries by weighting document recency (Chen et al., 2025). Adaptive query augmentation reformulates or expands queries to improve retrieval quality. Multi-modal RAG extends to images, tables, and other non-textual content. These advances reflect the growing importance of RAG as a paradigm for building reliable, knowledge-grounded AI systems.

2.2 Question Understanding and Semantic Similarity

Question understanding is the foundation of effective QA systems. A system that cannot recognize when two questions seek the same information will fail to retrieve relevant answers, will exhibit inconsistent behavior across phrasings, and will be vulnerable to minor surface variations. This section explores how questions are represented computationally, how semantic equivalence is defined, and the tasks designed to evaluate question similarity.

2.2.1 Question Representation

The representation of questions has evolved in parallel with broader developments in natural language processing, moving from manually designed features to distributed word embeddings and, more recently, to contextualized neural representations.

Early approaches relied on manually engineered features capturing different aspects of question structure and content. Lexical features included word overlap counts, n-gram matches, and TF-IDF weighted similarity. Syntactic features captured structural properties through part-of-speech tags, dependency parse trees, and constituency structures. Semantic features incorporated external knowledge through WordNet synonym expansion, named entity types, and question classification taxonomies (Zhang and Lee, 2010; Heilman and Smith, 2010; Moschitti, 2006). While interpretable, these handcrafted features required substantial engineering effort and often failed to capture subtle semantic relationships.

The introduction of distributed word representations marked a significant advance. Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) learned dense vector embeddings where semantically similar words occupy nearby regions of the embedding space. Questions could be represented by aggregating word embeddings through averaging, weighted pooling, or recurrent encoding. However, these static embeddings assign the same vector to a word

regardless of context, conflating distinct senses ("bank" as financial institution versus river-bank). Contextualized embeddings from ELMo (Peters et al., 2018) addressed this limitation by computing word representations as a function of the entire input sequence, capturing context-dependent meaning.

The introduction of Transformer architectures (Vaswani et al., 2017) and large-scale pre-training fundamentally reshaped question encoding. Models such as BERT (Devlin et al., 2019b) provide deep contextualized representations that capture rich interactions between all tokens in a sequence. These representations enable nuanced modeling of lexical, syntactic, and semantic cues that are difficult to capture with shallower approaches.

Two main architectural patterns emerged for computing question similarity from Transformer representations. Bi-encoders independently encode each question into a fixed-dimensional embedding, then compute similarity through cosine distance or dot product. Sentence-BERT (Reimers and Gurevych, 2019a) demonstrated that Transformer-based bi-encoders trained on natural language inference and semantic textual similarity data produce high-quality sentence embeddings. The key advantage of bi-encoders is efficiency: question embeddings can be pre-computed and indexed, enabling sub-linear retrieval through approximate nearest neighbor search. However, independent encoding prevents modeling of fine-grained token-level interactions between questions.

Cross-encoders process two questions jointly by concatenating them into a single sequence separated by a special token. The Transformer's self-attention mechanism then models token-level interactions across both questions (Peinelt et al., 2020). This joint encoding captures fine-grained lexical relationships that bi-encoders miss. For example, recognizing that "author" in one question aligns with "wrote" in another. In this case, Cross-encoders typically achieve higher accuracy than bi-encoders but are computationally expensive: similarity must be computed at inference time for each candidate pair, precluding pre-computation.

Modern systems adopt a hybrid approach that combines the efficiency of bi-encoders with the accuracy of cross-encoders (Nogueira and Cho, 2019). Bi-encoders perform fast first-stage retrieval over millions of candidates, returning a manageable set of top-k results. Cross-encoders then rerank these candidates with full attention over query-candidate pairs, achieving high accuracy where it matters most.

2.2.2 Semantic Equivalence

Defining when two questions are semantically equivalent is non-trivial. The definition adopted in this thesis requires two jointly necessary conditions: two questions are equivalent iff they (i) express the same information-seeking intent, and (ii) accept the same set of correct answers. Neither condition alone is sufficient. The first captures the underlying semantic goal of the question; the second provides an operationally verifiable grounding criterion that helps annotators resolve ambiguous cases. For instance, "*How tall is Mount Everest?*" and "*What is the height of the world's highest mountain?*" satisfy both: they express the same intent and accept identical answers.

This dual requirement addresses edge cases that either condition alone would miss. Questions may coincidentally share answers without expressing the same information need. For example, "*When was Shakespeare born?*" and "*When was Galileo born?*" both have "1564" as a correct answer, but they are clearly not equivalent: they ask about different entities and different events. This case is especially common when answers have low cardinality (dates, numbers, common named entities). More subtly, "*Who is the current president?*" and "*Who*

won the last election?" might share an answer at a given point in time but express structurally different information needs, one about a political office and the other about an electoral outcome, failing condition (i). In the opposite direction, questions with seemingly identical intent may require different answers depending on context: "*How old is he?*" could refer to any male entity. Accounting for context is essential during annotation and evaluation, particularly for conversational QA where questions depend on dialogue history (Choi et al., 2018; Reddy et al., 2019b).

The intent condition (i) thus ensures that equivalence reflects genuine semantic relatedness rather than accidental answer coincidence, while the answer condition (ii) grounds the definition in observable, task-relevant behavior. In annotation practice, both conditions are enforced jointly: annotators in the Question Ranking Corpus (Chapter 3) were shown both questions and answers, enabling them to verify that candidates express the same intent *and* accept the same answers. Chapter 3 introduces a formal operationalization (Definition 3.1.1) that captures both conditions for the specific requirements of database-based question answering.

While the use of answers as a signal for question similarity has been explored in prior work, Match^2 (Wang et al., 2020b) compares matching patterns of two questions over the same answer, and WikiAnswers (Fader et al., 2013) groups questions by shared answers into paraphrase clusters, these approaches treat answer overlap as a feature for similarity models rather than as a component of a formal equivalence definition. The dual formulation adopted here, requiring both intent equivalence and answer set identity, makes explicit the conditions under which answer-based similarity judgments are valid, and identifies the failure cases where answer overlap alone is insufficient.

2.2.3 Duplicate Question Detection

Duplicate Question Detection (DQD) is the task of identifying whether two questions are semantically equivalent. As discussed in Section 2.1.4, this capability is fundamental for DBQA, FAQ systems, and Community Question Answering platforms: recognizing that a user's query matches a stored question enables retrieving pre-computed answers despite lexical variation.

Traditional approaches to DQD relied on lexical matching methods including TF-IDF similarity and BM25, topic modeling approaches that cluster questions by latent themes (Cai et al., 2011; Ji et al., 2012), and syntactic features captured through tree kernels operating on parse structures (Moschitti, 2006). Shared evaluation benchmarks, particularly SemEval Task 3 on Community Question Answering (Nakov et al., 2016a), catalyzed systematic comparison and drove methodological advances.

Neural approaches brought substantial improvements to DQD. Convolutional and recurrent neural networks were among the first architectures applied (Lei et al., 2016; Hochreiter and Schmidhuber, 1997). Siamese networks with shared weights proved particularly effective, encoding both questions with identical networks and comparing their representations (Mueller and Thyagarajan, 2016). The Match-LSTM architecture (Wang and Jiang, 2016a,b) incorporated attention mechanisms to model fine-grained interactions between question tokens, identifying which parts of one question align with which parts of another.

Transformer-based methods significantly improved DQD performance. Sentence-BERT (SBERT) (Reimers and Gurevych, 2019a) introduced a Siamese architecture using BERT encoders trained on natural language inference and semantic textual similarity data, producing

embeddings well-suited for efficient similarity computation. tBERT (Peinelt et al., 2020) augmented BERT with topic model information to capture domain-specific patterns beyond what pre-training provides. Match^2 (Wang et al., 2020b) proposed a matching-over-matching approach that leverages answers as a bridge between questions: if two questions share similar answers, they are likely equivalent even if their surface forms differ substantially.

2.3 Transformer Models and Pre-training

The Transformer architecture (Vaswani et al., 2017) marked a paradigm shift in NLP, enabling unprecedented advances in language understanding and generation. Combined with large-scale pre-training on massive text corpora, Transformer-based models have become the foundation for virtually all state-of-the-art NLP systems.

2.3.1 The Transformer Architecture

The Transformer departs fundamentally from previous sequence models by replacing recurrent and convolutional operations with self-attention mechanisms. This design choice enables fully parallel computation across sequence positions and effective modeling of long-range dependencies without the vanishing gradient problems that plague recurrent networks.

The core innovation is scaled dot-product attention, which computes relevance-weighted combinations of value vectors:

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

where Q (queries), K (keys), and V (values) are linear projections of input embeddings. Each position attends to all other positions, with attention weights determined by query-key compatibility. The scaling factor $\sqrt{d_k}$ prevents dot products from growing large in magnitude, which would push softmax into regions with small gradients.

Multi-head attention extends this mechanism by running h parallel attention operations with different learned projections:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

Different heads can learn to attend to different types of relationships, including, syntactic dependencies, semantic associations, and positional patterns, providing thus a richer representational capacity than single-head attention. Figure 2.4, illustrates both single and multihead attention.

Three main architectural patterns emerged from the original Transformer design. Encoder-only models like BERT use bidirectional self-attention, where each position attends to all positions in the sequence. This is natural for classification and embedding tasks where the full input is available. Decoder-only models like GPT (Radford et al., 2018) use causal (left-to-right) attention, where each position attends only to previous positions. This autoregressive structure enables generation by predicting one token at a time. Encoder-decoder models like T5 (Raffel et al., 2020b) and BART (Lewis et al., 2020a) combine bidirectional encoding of the input with autoregressive decoding of the output, suitable for sequence-to-sequence tasks like translation and summarization.

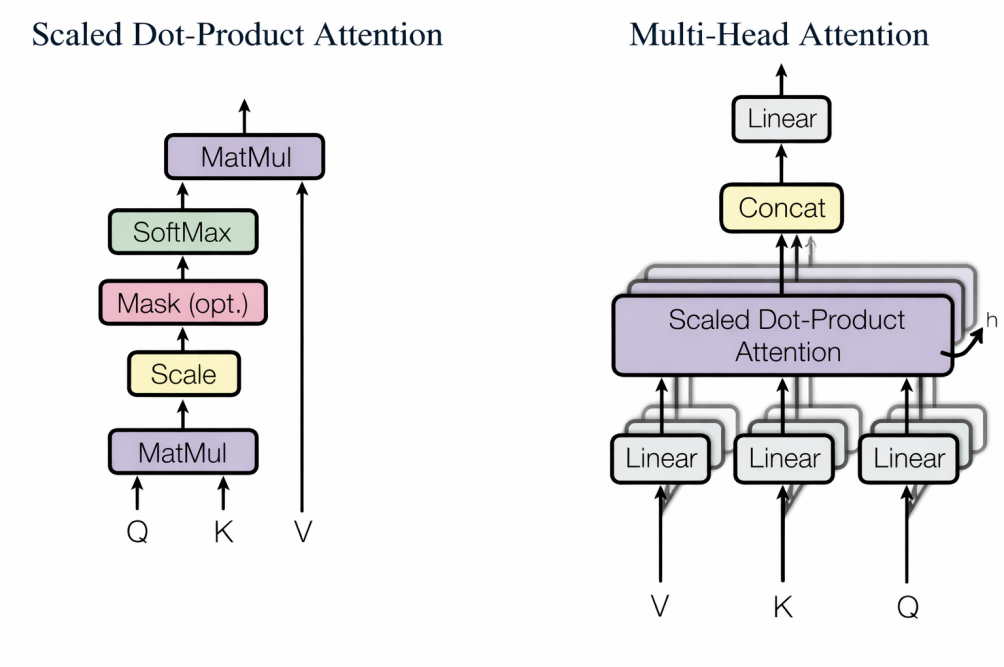

Figure 2.4: The figure illustrates the Scaled Dot-Product Attention mechanism (left) and its Multi-Head Attention extension (right), as introduced in the Transformer architecture.

2.3.2 Pre-training Objectives

Pre-training on large unlabeled corpora enables models to acquire broad linguistic knowledge before task-specific fine-tuning. The choice of pre-training objective significantly impacts what knowledge models acquire and how well they transfer to downstream tasks. Pre-training objectives have evolved from token-level prediction tasks through sentence-level objectives to task-specific approaches designed for particular applications.

Masked Language Modeling (MLM). Introduced by BERT (Devlin et al., 2019b), MLM randomly masks approximately 15% of input tokens and trains the model to reconstruct them using bidirectional context. The model must leverage both left and right context to predict masked tokens, encouraging rich contextual representations. Variants include whole word masking (masking all tokens of a word together), entity masking (preferentially masking named entities), and span masking (masking contiguous spans rather than individual tokens) (Joshi et al., 2020). MLM produces strong general-purpose representations but provides learning signal only for masked positions.

Causal Language Modeling (CLM). Used by the GPT family (Radford et al., 2018), CLM trains models to predict the next token autoregressively given all previous tokens. This objective aligns naturally with text generation and enables zero-shot and few-shot learning by conditioning on task descriptions and examples. Scaling CLM to massive model sizes has proven remarkably effective, with emergent capabilities arising at scale that are absent in smaller models (Wei et al., 2022b). The autoregressive nature means models see only left context when making predictions, potentially limiting bidirectional understanding.

Replaced Token Detection (RTD). ELECTRA (Clark et al., 2020) introduced a generator-discriminator setup for more efficient pre-training. A small generator network proposes re-

placement tokens for masked positions, and a discriminator network learns to identify which tokens have been replaced. Unlike MLM where only masked positions provide learning signal, RTD provides signal for all positions, substantially improving compute efficiency. Related approaches include Random Token Swap (RTS) (Di Liello et al., 2022b), which perturbs input sequences to create self-supervised signals without requiring a separate generator.

Contrastive Learning. Methods like SimCSE (Gao et al., 2021) and DeCLUTR (Giorgi et al., 2021) learn sentence representations by contrasting semantically similar pairs with dissimilar ones. SimCSE uses dropout as minimal augmentation: passing the same sentence through the encoder twice with different dropout masks produces two similar representations that serve as positive pairs, while other sentences in the batch serve as negatives. DeCLUTR samples spans from the same document as positive pairs, leveraging document coherence as a supervision signal. These approaches produce representations particularly well-suited for semantic similarity and retrieval tasks.

Sentence-level Objectives. Beyond token-level tasks, several objectives target sentence-level understanding. Next Sentence Prediction (NSP), introduced with BERT, trains models to predict whether two sentences are consecutive in the original document. However, subsequent work showed that NSP provides limited benefit and may even hurt performance, leading to its removal in models like RoBERTa (Liu et al., 2019). More effective sentence-level objectives focus on semantic similarity or in emulating the final task, rather than discourse coherence.

2.3.3 Task-oriented Pre-training

While general pre-training objectives capture broad linguistic knowledge, they may not optimally prepare models for specific downstream tasks. Task-oriented pre-training designs objectives that more closely mimic the structure of target tasks, potentially improving transfer efficiency and final performance.

For answer sentence selection, Di Liello et al. (2022a,c) proposed pre-training objectives based on predicting whether sentences belong to the same paragraph. This simulates the structure of AS2 tasks where models must identify relevant answer sentences among candidates from the same document. The paragraph structure provides natural positive pairs, that are sentences from the same paragraph, and negatives that are sentences from different paragraphs, without manual annotation.

Despite advances in task-oriented pre-training, a gap remained for question ranking specifically. Token-level objectives like MLM and RTD do not explicitly model relationships between complete questions. Sentence-level objectives like NSP and contrastive learning focus on discourse coherence or general semantic similarity rather than the specific notion of question equivalence. Answer-oriented methods address question-answer matching but not question-question similarity.

Chapter 4 addresses this gap by introducing Question Ranking Pre-training (QRP), a novel objective specifically targeting question ranking through self-supervised corruption detection over retrieved candidate sets.

2.3.4 Notable Pre-trained Models

The landscape of pre-trained models has expanded rapidly, with different architectures suited to different applications.

Among encoder-only models, BERT (Devlin et al., 2019b) pioneered bidirectional pre-training via MLM and remains widely used. RoBERTa (Liu et al., 2019) improved upon BERT through longer training, larger batches, and removal of NSP. ELECTRA (Clark et al., 2020) achieved strong performance with greater compute efficiency through RTD task. DeBERTa (He et al., 2021) introduced disentangled attention mechanisms that separately model content and position, achieving state-of-the-art results on many benchmarks. MPNet (Song et al., 2020) combined masked and permuted language modeling to capture both token dependencies and positional information. For efficient deployment, MiniLM (Wang et al., 2020a) applied deep self-attention distillation to compress large models while preserving quality, making it popular for sentence embeddings. More recently, ModernBERT (Warner et al., 2024) revisited the encoder architecture incorporating advances from decoder-only models, such as the flash attention (Dao et al., 2022), achieving improved efficiency and performance.

Decoder-only models have scaled dramatically in recent years. The GPT family (Radford et al., 2018; Brown et al., 2020) demonstrated the power of scaling causal language modeling, with GPT-3's 175 billion parameters enabling impressive few-shot learning. LLaMA (Touvron et al., 2023) provided high-quality open-weight models that democratized access to large-scale language modeling. Mistral (Jiang et al., 2023a) introduced efficient architectural innovations including grouped-query attention that reduce memory requirements while maintaining quality. The Qwen family (Bai et al., 2023) from Alibaba achieved competitive performance across diverse tasks with models ranging from 0.5B to 72B parameters. Microsoft's Phi series (Abdin et al., 2024a) demonstrated that smaller models trained on high-quality data can match larger models on many benchmarks, with Phi-3 and Phi-4 offering strong reasoning capabilities at reduced computational cost.

Mixture-of-Experts (MoE) architectures offer an alternative scaling paradigm that increases model capacity without proportionally increasing computational cost (Shazeer et al., 2017). Rather than activating all parameters for every input, MoE models route each token to a subset of specialized "expert" networks. Mixtral-8x7B (Jiang et al., 2024) exemplifies this approach: despite having 56 billion total parameters across eight experts, it activates only 13 billion parameters per forward pass by routing each token to two experts. This sparse activation enables the model to maintain the quality of much larger dense models while achieving inference efficiency comparable to smaller ones. The routing mechanism learns which experts specialize in different types of inputs, effectively creating an ensemble that dynamically adapts to each example.

Multilingual and cross-lingual models extend language understanding across linguistic boundaries. mBERT (Devlin et al., 2019b) demonstrated that multilingual pre-training on 104 languages produces representations that transfer across languages, enabling zero-shot cross-lingual transfer. XLM-RoBERTa (Conneau et al., 2020) scaled this approach with larger data and improved training. For sentence embeddings specifically, LaBSE (Language-agnostic BERT Sentence Embedding) (Feng et al., 2022) produces embeddings that are directly comparable across 109 languages, trained with a translation ranking task that aligns representations of parallel sentences. This cross-lingual alignment makes LaBSE particularly suitable for multilingual semantic similarity and coherence evaluation where embeddings from different languages must be compared directly.

Encoder-decoder models excel at sequence-to-sequence tasks. T5 (Raffel et al., 2020a) unified diverse NLP tasks under a text-to-text framework, treating everything from classification to translation as text generation. BART (Lewis et al., 2020a) combined bidirectional encoding with autoregressive decoding, pre-trained as a denoising autoencoder. Flan-T5 (Chung

et al., 2022) applied instruction tuning to T5, substantially improving zero-shot and few-shot performance through training on diverse prompted tasks.

2.3.5 LLM Alignment and Preference Learning

Pre-training produces models with broad linguistic capabilities, but these models may generate harmful, unhelpful, or inconsistent outputs. *Alignment* refers to the process of adjusting model behavior to better match human intentions and preferences (Ouyang et al., 2022). This post-training phase has become essential for deploying LLMs in real-world applications, and the techniques developed for alignment can be repurposed for other objectives such as improving model coherence.

Supervised Fine-Tuning (SFT) is the simplest alignment approach fine-tunes a pre-trained model on demonstrations of desired behavior. Given a dataset of (instruction, response) pairs $\mathcal{D} = \{(x_i, y_i)\}$, SFT maximizes the likelihood of target responses:

$$\mathcal{L}_{\text{SFT}} = -\sum_{(x,y)\in\mathcal{D}} \log p_\theta(y|x) \tag{2.1}$$

Instruction tuning (Wei et al., 2022a; Chung et al., 2022) applies SFT to diverse prompted tasks, substantially improving zero-shot and few-shot generalization. The quality and diversity of demonstration data critically affects outcomes: models trained on high-quality demonstrations exhibit better instruction-following capabilities. While SFT is effective and straightforward, it requires curated examples of ideal behavior which can be expensive to collect, and it optimizes for imitation rather than explicitly for human preferences.

Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Stiennon et al., 2020) optimizes models directly for human preferences through a multi-stage process. First, human annotators compare model outputs and indicate which they prefer. These comparisons train a *reward model* $r_\phi(x, y)$ that predicts human preference scores. The language model is then fine-tuned using reinforcement learning (typically Proximal Policy Optimization, PPO) to maximize expected reward while staying close to the original model through a KL penalty:

$$\mathcal{L}_{\text{RLHF}} = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[r_\phi(x, y) - \beta\,\text{KL}(\pi_\theta(y|x)\|\pi_{\text{ref}}(y|x))\right] \tag{2.2}$$

RLHF has proven effective for improving helpfulness and reducing harmful outputs, but the multi-stage pipeline is complex: training a separate reward model, then using RL optimization which can be unstable and sample-inefficient.

Direct Preference Optimization (DPO) (Rafailov et al., 2023) offers a simpler alternative that directly optimizes the policy to satisfy preferences without learning an explicit reward model. Given preference pairs (x, y_w, y_l) where y_w is preferred over y_l, DPO derives a closed-form loss from the RLHF objective:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right] \tag{2.3}$$

where σ is the sigmoid function, π_θ is the policy being optimized, π_{ref} is a reference policy (typically the SFT model), and β controls the strength of the KL constraint. Intuitively, DPO increases the relative probability of preferred responses while decreasing that of dispreferred ones, with the reference model preventing excessive deviation. DPO achieves comparable results to RLHF with substantially simpler implementation: it requires only supervised learning

on preference pairs, avoiding the complexity of reward model training and RL optimization. This simplicity makes DPO particularly attractive for applications beyond traditional alignment, such as optimizing for coherence across equivalent inputs (Chapter 5).

2.4 From Sparse to Dense Retrieval

Information retrieval has undergone fundamental transformation with neural methods. Traditional approaches represented documents and queries as sparse, high-dimensional vectors based on term frequencies. Dense retrieval instead represents texts as continuous, low-dimensional vectors learned by neural networks, enabling semantic similarity matching that transcends exact term overlap.

2.4.1 Sparse Retrieval Methods

Traditional retrieval systems relied on sparse lexical representations where each dimension corresponds to a vocabulary term. BM25 (Robertson and Zaragoza, 2009b), the most successful sparse method, computes relevance through a probabilistic ranking function incorporating term frequency saturation and document length normalization:

$$\text{BM25}(q,d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t,d) \cdot (k_1 + 1)}{\text{TF}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)}$$

Despite its simplicity, BM25 remains remarkably competitive on many retrieval benchmarks and powers production search systems worldwide. Its efficiency stems from inverted index structures that enable sub-linear retrieval complexity.

However, sparse methods suffer from the *vocabulary mismatch problem*: queries and relevant documents may express the same concepts using entirely different words. A query asking about "*car prices*" may fail to retrieve a relevant document discussing "*automobile costs*" because there is no lexical overlap despite clear semantic relevance. Query expansion techniques partially address this by adding related terms, but they introduce noise and cannot fully bridge semantic gaps.

Dense retrieval addresses vocabulary mismatch by learning continuous representations where semantically similar texts occupy nearby regions of embedding space. A query about "*car prices*" and a document about "*automobile costs*" can have high similarity if the encoder learns that these phrases express related concepts. This semantic matching capability is the primary advantage of dense methods.

2.4.2 Dense Retrieval Methods

Dense retrieval methods encode queries and documents independently into fixed-dimensional embeddings, then compute relevance through vector similarity (typically dot product or cosine similarity).

Sentence-BERT (SBERT) (Reimers and Gurevych, 2019a) adapted pre-trained BERT for efficient sentence embeddings using Siamese architectures. Two identical encoders process query and document separately, producing embeddings that can be compared directly. SBERT introduced several training objectives that became standard for dense retrieval. Multiple Negatives Ranking (MNR) loss treats other examples in the batch as negatives, providing

efficient training signal without explicit negative mining. Triplet loss separates positive documents from negatives by a fixed margin. Contrastive loss pulls similar pairs together while pushing dissimilar pairs apart.

Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) demonstrated that dense methods could substantially outperform BM25 for open-domain QA. DPR uses separate encoders for questions and passages, allowing asymmetric architectures optimized for each input type. Training employs contrastive loss with carefully mined hard negatives.

ColBERT (Khattab and Zaharia, 2020) introduced multi-vector representations that preserve token-level information. Rather than compressing each text into a single vector, ColBERT represents documents as bags of contextualized token embeddings. Relevance is computed via MaxSim: each query token attends to its best-matching document token, and these maximum similarities are summed. This late interaction mechanism captures fine-grained matching while still enabling efficient retrieval through approximate search over token embeddings.

Recent advances have further improved dense retrieval. Contriever (Izacard et al., 2022) demonstrated effective unsupervised training using data augmentation techniques like cropping and word deletion to create positive pairs. Models like BGE (Xiao et al., 2023a) and E5 (Wang et al., 2024b) demonstrated to achieve strong zero-shot retrieval performance. Instruction-following retrievers (Su et al., 2023; Wang et al., 2024a) accept natural language task descriptions that condition retrieval behavior, enabling a single model to handle diverse retrieval needs. Some recent work explores adapting large language models directly for retrieval, leveraging their extensive pre-training for semantic understanding. SPLADE (Formal et al., 2021) bridges dense and sparse approaches by learning sparse representations where weights are determined by neural networks rather than term frequency statistics. In this case, the model naturally performs query expansion while leveraging efficient inverted index infrastructure.

For our experiments we relied on the usage of SentenceTransformers as they provide (i) inference and training efficiency, and (ii) optimal performances.

2.4.3 Dense Retrieval for Question Matching

Question retrieval differs from document retrieval in a key aspect: the goal is semantic equivalence rather than topical relevance. Two questions about "weather in Paris" and "Paris climate" may be topically related but are not equivalent since they seek different information. Question retrieval must make finer distinctions than document retrieval, requiring representations that capture precise semantic intent.

Bi-encoder architectures with shared weights are standard for question matching. Sentence Transformers (Reimers and Gurevych, 2019a) encode query and candidate questions separately using the same encoder, then compute similarity via cosine distance. The shared encoder ensures that semantically equivalent questions, regardless of which is query or candidate, map to nearby embeddings.

A distinctive aspect of question matching in DBQA is the availability of associated answers as additional context. The choice of what to encode significantly impacts performance: question-only encoding (Q) uses just the question text, while question-answer encoding (QA) concatenates the question with its answer. As demonstrated in Chapter 3, incorporating answers substantially improves retrieval accuracy, as answers disambiguate intent when questions are underspecified or ambiguous.

2.4.4 Cross-Encoder Reranking

While bi-encoders enable efficient retrieval through pre-computed embeddings and approximate nearest neighbor search, cross-encoders achieve higher accuracy by jointly encoding query-candidate pairs. The cross-encoder processes query and candidate as a single concatenated sequence, allowing full self-attention across both texts. This enables modeling of fine-grained token-level interactions, recognizing that "wrote" in the query aligns with "author" in the candidate, that independent encoding cannot capture.

Production systems typically combine both approaches in a retrieve-and-rerank pipeline. The bi-encoder retrieves the top-k candidates (typically k = 100-1000) from the full collection in milliseconds using a full similarity search. The cross-encoder then reranks these candidates, applying expensive but accurate joint encoding only to the small candidate set. This pipeline achieves the accuracy of cross-encoders at a fraction of the computational cost of exhaustively encoding all pairs.

2.4.5 Sensitivity in Dense Retrieval

Dense retrieval models exhibit sensitivity to query formulation similar to what is observed in LLMs. This sensitivity can be understood as a problem of coherence: models often fail in producing consistent retrieval outcomes for semantically equivalent or near-equivalent queries. The sensitivity or coherence, has practical consequences: behavioral studies have shown that users frequently start multiple searches with rewritten queries when initial results are unsatisfactory (Bernard et al., 2007; Jansen et al., 2005), with estimates suggesting that up to 50% of traffic in early retrieval engines consisted of query reformulations (Wang et al., 2021).

Recent work indicates this problem persists in modern retrieval systems. Liu et al. (2023a) studied sensitivity in generative retrieval settings through simple query variations including misspelling, token order modification, and rule-based paraphrasing, quantifying the impact of these perturbations on retrieval results. Campos et al. (2023) proposed CAPOT (Contrastive Alignment POst Training), focusing on making query encoders robust to noise (lemma variations, stemming, character swaps, and paraphrasing) while keeping document encoders frozen. Chen et al. (2024a) proposed an unsupervised technique to make model scores robust toward irrelevant paragraphs in documents. Other work has examined sensitivity from an adversarial viewpoint (Liu et al., 2023b; Wu et al., 2022).

Several approaches have been explored. Synthetic query data augmentation has shown promise for improving generalization (Chaudhary et al., 2024; Liang et al., 2020; Meng et al., 2024), with Guo et al. (2025) specifically targeting queries with negations. Sunkara (2024) combined query augmentation via back-translation with multi-task learning that forces embeddings of equivalent queries to be similar, though results did not consistently improve over standard training.

Query rewriting offers an alternative approach, aligning input distribution to retriever-preferred query forms (He et al., 2016a). Shi et al. (2024) demonstrated benefits of using multiple query rewrites with subsequent combination of retrieved documents, while Ma et al. (2023) introduced a trainable rewrite-and-retrieve approach for RAG settings. However, query rewriting requires additional components in the retrieval pipeline, typically LLMs, increasing latency and cost in industrial applications.

Chapter 5 addresses retrieval coherence through a different approach: directly training the dense retrieval model to produce consistent rankings across equivalent queries through a Coherence Ranking (CR) loss, without external rewriting components.

2.5 LLM Coherence

Large Language Models have become central to modern QA systems, serving both as closed-book QA models that answer from parametric memory and as generation components in RAG systems. However, despite remarkable capabilities, LLMs exhibit a critical weakness: instability with respect to input variation. Slight changes in question phrasing can produce inconsistent or contradictory outputs (Voronov et al., 2024b; Mizrahi et al., 2024b). Understanding and addressing this limitation is essential for building reliable QA systems.

2.5.1 Prompt Sensitivity

LLMs are surprisingly sensitive to seemingly minor variations in prompt wording, formatting, and structure (Hu et al., 2024a). Different phrasings of semantically equivalent questions may bring to different responses. For example, a model might correctly answer "*What is the capital of France?*" but fail on "*France's capital is?*" even though both queries seek identical information. Formatting changes such as adding or removing punctuation, adjusting whitespace, or restructuring sentences can affect outputs despite preserving meaning.

Beyond surface-level formatting, the position and order of information affects model behavior. For multiple-choice questions, reordering answer options can change model predictions, a phenomenon known as position bias (Zheng et al., 2023a). Similarly, the order of in-context examples in few-shot settings influences outputs (Liu et al., 2022; Zhao et al., 2021). Chatterjee et al. (2024) introduced POSIX, a metric specifically designed to quantify prompt sensitivity, enabling systematic measurement of how much outputs vary under input perturbations.

These sensitivities undermine reliability, as users cannot predict which phrasings will succeed or fail.

2.5.2 Measuring Coherence

Coherence in QA refers to consistency of outputs for semantically equivalent inputs. A coherent model produces the same answer whether asked "*What is the capital of France?*" or "*France's capital is?*". Both queries should reliably be answered with "*Paris.*"

Rabinovich et al. (2023) introduced PopQA-TP, a dataset of 118K paraphrased questions organized into 14K clusters, specifically designed to benchmark LLM coherence. Coherence can be measured through embedding-based similarity between answers to equivalent questions, or through discrete metrics based on answer correctness. These metrics are formally defined in Section 2.9.

A useful conceptual distinction emerges from coherence analysis: clusters can be classified as *coherent-correct* when all answers in the cluster are correct, *coherent-incorrect* when all answers are incorrect, or *incoherent* when exist a mixed correctness. Incoherent clusters are particularly informative for understanding model behavior as they indicate that the model have relevant knowledge to correctly answer to some questions, but fails to access it reliably across all phrasings of the same question. Chapter 5 provides systematic analysis of LLM coherence across multiple models and introduces methods to address understanding failures through question prompting.

2.5.3 Approaches to Improving Coherence

Several approaches have been explored to improve LLM coherence.

The first approach is based on Prompt Engineering, a techniques that encourage more systematic reasoning can improve consistency. Chain-of-thought prompting (Kojima et al., 2022; Wei et al., 2022c) asks models to show reasoning steps, potentially improving consistency by making the inference process explicit. Tree-of-thoughts (Yao et al., 2023) extends this to explore multiple reasoning paths. Few-shot examples demonstrate desired behavior, though they consume context window capacity. Le Scao and Rush (2021) showed that well-designed prompts can be as effective as hundreds of training examples.

A second possible approach is based on Fine-tuning. In this case instruction tuning trains models on diverse prompted tasks, improving ability to follow specifications (Wei et al., 2022a). Preference optimization methods such as RLHF and DPO (Section 2.3.5) can optimize for human preferences, potentially including consistency preferences. Chapter 5 demonstrates that DPO can be applied to distill coherence improvements into model parameters by training on preference pairs where coherent answers are preferred over incoherent ones.

A final approach is based on Retrieval Augmentation. Standard RAG (Gao et al., 2024; Chen et al., 2024b) provides relevant documents that constrain generation, grounding outputs in external evidence rather than relying on potentially inconsistent parametric memory. Chapter 5 introduces an alternative approach: retrieving semantically equivalent questions (a cluster) and presenting them alongside the original query provides redundant semantic signal about user intent. This *question prompting* approach improves both accuracy and coherence by providing multiple perspectives on the same information need, addressing understanding failures rather than knowledge gaps.

2.6 Question Clustering

Question clustering groups semantically similar or equivalent questions into coherent sets. This capability has fundamental applications across QA systems: organizing FAQs, deduplicating community question archives, improving model coherence through redundant signal, constructing evaluation datasets, and aggregating user feedback. The ability to identify clusters of equivalent questions is central to several contributions in this thesis.

2.6.1 Foundations

Building on pairwise equivalence from Section 2.2, we extend to sets of questions. A question cluster $C = \{q_1, q_2, \ldots, q_n\}$ is a set of questions where all pairs are semantically equivalent:

$$\forall (q_i, q_j) \in C^2 : q_i \leftrightarrow q_j$$

where $\leftrightarrow$ denotes semantic equivalence. By transitivity, if questions share the same information-seeking intent and accept interchangeable answers, they belong to the same cluster.

Question clusters exhibit important properties that enable their applications. All questions in a cluster should accept the same set of correct answers, enabling answer sharing across phrasings.

2.6.2 Constructing Question Clusters

Question clusters can be constructed through different approaches. Classical clustering algorithms such as k-means, hierarchical clustering, and DBSCAN, can be applied to question embeddings from neural encoders, partitioning questions based on similarity in embedding space.

However, for the applications in this thesis, a retrieval-based approach is more natural. Given a seed question, a dense retrieval model identifies the top-k most similar questions from a large database. These retrieved questions implicitly form a cluster around the seed, sharing semantic equivalence without requiring explicit global clustering. Chapter 5 adopts this retrieval-based formulation for both LLM coherence improvement (question prompting) and retrieval model training (coherence constraints).

2.7 Privacy-Preserving NLP and Dataset Release

The ability to share datasets drives research progress, but privacy concerns, proprietary restrictions, and benchmark contamination increasingly constrain what can be released. This section reviews approaches to transforming or protecting QA data while preserving its utility, providing context for the declassification framework introduced in Chapter 6.

2.7.1 Data Sanitization and Anonymization

The most direct approach to privacy-preserving data release is identifying and removing sensitive content. Traditional anonymization redacts personally identifiable information (PII) such as names, locations, and identifying numbers. Entity recognition systems identify sensitive spans, which are then masked, replaced with placeholders, or substituted with synthetic alternatives (Yermilov et al., 2023).

However, simple redaction often degrades data utility substantially. Pal et al. (2024a) observed that PII redaction can cause over 25% accuracy drops in QA tasks, as removed entities often carry semantic content essential for understanding. Questions like "*What year did [PERSON] win the Nobel Prize?*" lose critical information when the person's name is masked. More sophisticated approaches replace entities with semantically similar alternatives rather than generic placeholders, preserving some utility while protecting specific identities (Pilán et al., 2022; Sánchez and Batet, 2016).

Differential privacy (DP) provides formal mathematical guarantees by adding calibrated noise to data or computations (Dwork, 2006). DP has become a de facto standard for privacy-preserving machine learning, with guarantees that bound information leakage about any individual record. However, achieving meaningful privacy guarantees typically requires substantial noise that degrades utility. The privacy-utility trade-off is particularly acute for text data, where discrete tokens and semantic coherence constraints limit noise tolerance.

Even with anonymization, residual privacy risks remain. Membership inference attacks can determine whether specific examples appeared in training data. Attribute inference can recover sensitive properties from seemingly innocuous features. Dataset statistics may enable re-identification when combined with auxiliary information. These attacks have been demonstrated against NLP models and datasets (Hu et al., 2024b), motivating stronger protection approaches.

2.7.2 Synthetic Data Generation

Rather than sanitizing real data, an alternative paradigm generates synthetic data that captures the statistical properties needed for training and evaluation without corresponding to real individuals or events. Recent advances in large language models have made synthetic data generation increasingly viable.

For QA applications, LLMs can generate question-answer pairs that mimic real data distributions. Bai et al. (2024) used GPT-4 to generate clinical QA pairs from electronic health records, carefully prompting the model to produce challenging questions while avoiding patient-specific information. The synthetic data improved downstream QA performance while alleviating privacy constraints that limit access to real medical data. Kotschenreuther (2024) constructed a large synthetic medical QA dataset by prompting LLaMA 2 to generate questions from clinical discharge summaries, producing 156000 question-answer pairs with a physician-verified subset.

Synthetic data can also address benchmark contamination by generating fresh evaluation examples. If original benchmark questions have leaked into training data, synthetic equivalents can provide uncontaminated evaluation. Xia et al. (2024) proposed automatically generating alternative test prompts for coding tasks by rephrasing, combining, or adjusting difficulty of existing questions. The resulting evaluation sets preserve task semantics while removing specific phrasings that might have been memorized.

However, synthetic generation has limitations. Generated data may not faithfully capture real data distributions, introducing biases or missing important phenomena. Quality control requires substantial human verification effort. For sensitive domains, even the process of generating synthetic data may require access to protected content. And synthetic data cannot address scenarios where specific real questions must be preserved, such as evaluating on established benchmarks.

2.7.3 Translation-based Transformation

Neural Machine Translation (NMT) offers another approach to dataset transformation: translating content to another language and back, know as backtranslation, produces paraphrased versions that preserve semantic content while changing surface form. Modern NMT systems employ Transformer-based encoder-decoder architectures trained on parallel corpora spanning hundreds of language pairs. The encoder maps source sentences to contextualized representations, and the decoder generates target translations autoregressively while attending to encoder outputs.

Massively multilingual models have extended NMT to hundreds of languages simultaneously. mBART (Liu et al., 2020) pre-trains a denoising autoencoder on monolingual corpora across 25 languages, then fine-tunes on parallel data for translation. NLLB-200 (No Language Left Behind) (Team et al., 2022) pushed multilingual coverage further, supporting direct translation between 200 languages including many low-resource languages previously underserved by NMT. The NLLB-3.3B model achieves strong translation quality across diverse language pairs through distillation from larger teacher models and careful data curation.

For dataset transformation, translation provides a principled paraphrasing mechanism. Translating a question to an intermediate language and back to the source language produces a semantically equivalent reformulation with different lexical choices and syntactic structure. This round-trip translation approach can generate dataset variants that test the same capabilities with different surface forms, addressing both benchmark contamination (translated

versions are unlikely to appear in training data) and evaluation robustness (consistent performance across phrasings indicates genuine understanding). Chapter 6 employs NLLB-200 as a baseline for creating equivalent translation-based versions dataset, comparing backtranslation against retrieval-based approaches.

2.7.4 Knowledge Distillation for Data Release

Knowledge distillation offers another approach: train a "teacher" model on private data, then distill its knowledge into a "student" model using public data. The student model captures capabilities learned from private data without directly exposing that data. This approach has been proposed for releasing models trained on proprietary corpora.

However, distillation provides weaker privacy guarantees than often assumed. Zhang et al. (2025) demonstrated that distilled students can memorize many of the same examples as teachers, and membership inference attacks remain effective against students. None of their knowledge distillation variants achieved membership inference AUC below ≈ 0.60, indicating substantial information leakage. Additional protections such as differential privacy during distillation may be needed for meaningful privacy guarantees.

2.7.5 Retrieval-Based Data Transformation

Retrieval-based methods offer an alternative to generation: rather than creating new content, find existing public content that serves the same purpose. For QA applications, this means finding public questions semantically equivalent to private questions, then using the public questions as substitutes.

Parvez et al. (2023) applied this idea in privacy policy QA, using dense retrievers to find relevant sentences in public corpora for each private query. Retrieved public sentences augmented training data without exposing private policy text. The retrieval-based augmentation substantially expanded training data while avoiding direct exposure of proprietary content.

More ambitiously, entire retrieval databases can be replaced with synthetic content. Zeng et al. (2025) proposed SAGE, a pipeline that rewrites private documents into synthetic equivalents by extracting key attributes, generating synthetic text, and refining with additional prompts. An LLM using the synthetic corpus for retrieval achieved comparable QA performance to using original data while substantially reducing privacy risk.

These retrieval-based approaches connect directly to the declassification framework developed in Chapter 6. By leveraging large-scale question retrieval to find public equivalents for private questions, datasets can be transformed while preserving semantic content and utility. The key insight is that question equivalence, properly operationalized through retrieval, enables systematic content replacement.

2.7.6 Benchmark Contamination

A distinct but related concern is benchmark contamination: evaluation datasets inadvertently appearing in LLM training corpora, compromising evaluation validity. As training data scales to trillions of tokens scraped from the web, the probability of including benchmark content increases. Even without direct inclusion, discussions, solutions, and paraphrases of benchmark questions may appear in training data (Sainz et al., 2023; Balloccu et al., 2024). Contamination undermines evaluation in several ways. Models may achieve high scores through mem-

Table 2.1: Dataset statistics for primary benchmarks used in this thesis. For WikiQA and TrecQA, numbers indicate questions; each question has multiple answer candidates (on average 9 for WikiQA, 38 for TrecQA), yielding larger total example counts. Statistics refer to the "clean" splits standard for evaluation; TrecQA train-all contains 1,229 questions but includes noisy annotations.

Dataset	Task	Train	Dev	Test
MS MARCO	Passage Ranking	502939	6980	—
WikiQA	Answer Selection	2118	296	237
TrecQA	Answer Selection	94	65	68
OpenBookQA	Scientific Reasoning	4957	500	500
SimpleQA	Factual QA	—	—	4326
QuoraQP	Duplicate Detection	404000 pairs		
PopQA-TP	Question Paraphrase	118000 questions in 14000 clusters		

orization rather than generalization. Performance on contaminated benchmarks does not predict performance on fresh questions. Comparisons between models trained on different corpora become invalid if contamination levels differ.

Detecting contamination is challenging. Direct string matching may miss paraphrases. Statistical tests for memorization have limited power. Models may exhibit partial contamination where some but not all benchmark examples were seen during training. Several approaches address contamination. Temporal splits evaluate only on data created after training cutoffs, though this limits evaluation to recent content. Dynamic benchmarks regenerate evaluation questions periodically, though this prevents longitudinal comparison. Contamination-resistant evaluation designs tests that are difficult to memorize, such as questions requiring compositional reasoning over multiple facts.

2.8 Benchmark Datasets

Progress in QA research has been driven by benchmark datasets that enable systematic evaluation and comparison of approaches. This section surveys the key resources used throughout the thesis. Table 2.1 summarizes statistics for the primary datasets.

Answer Sentence Selection. These datasets require ranking or selecting whole sentences that answer a question. **WikiQA** (Yang et al., 2015) provides questions from Bing query logs paired with candidate answer sentences from Wikipedia. Each question has multiple candidate sentences, some correct and some incorrect, making it suitable for evaluating ranking models. A peculiarity of WikiQA is that many questions have either no correct answers or only correct answers; the "clean" splits used in evaluation retain only questions with at least one correct and one incorrect candidate. **TrecQA** (Wang et al., 2007) derives from TREC QA tracks 8–13 and represents a classic benchmark for answer selection. Unlike WikiQA's encyclopedic focus, TrecQA consists of factoid questions with candidate sentences from newswire corpora, often concerning news events and temporal facts from the late 1990s and early 2000s. Processed "clean" versions are standard, retaining only questions with unambiguous annotations. Together, WikiQA and TrecQA provide complementary evaluation: encyclopedic versus news domains, Bing queries versus TREC competition questions. Both are used extensively in Chapter 6.

Passage Ranking and Retrieval. **MS MARCO** (Nguyen and et al., 2016) comprises over one million queries from Bing search logs with human-generated answers. The dataset provides both passage ranking and answer generation tasks, and its scale makes it valuable for training and evaluating dense retrieval models. **Natural Questions (NQ)** (Kwiatkowski et al., 2019) contains real queries from Google search paired with Wikipedia articles, requiring systems to identify both the relevant passage and extract the answer span. With 132K training queries over 2.68 million passages, NQ provides a large-scale benchmark for dense retrieval that complements MS MARCO's web search focus with encyclopedic content. **BEIR** (Thakur et al., 2021) provides a heterogeneous benchmark comprising 18 diverse retrieval datasets for zero-shot evaluation, testing generalization across domains (biomedical, financial, scientific) and task types (fact verification, argument retrieval, duplicate detection). All these datasets are used in Chapter 5 for evaluating retrieval coherence.

Generative QA and LLM Evaluation. **OpenBookQA** (Mihaylov et al., 2018) tests scientific reasoning by requiring models to combine elementary science facts with broad common knowledge to answer multiple-choice questions. The "open book" refers to a provided set of 1326 science facts, but answering requires additional commonsense reasoning not explicitly stated. **SimpleQA** (Wei et al., 2024) evaluates factual accuracy in large language models, containing 4326 short fact-seeking questions with unambiguous, verifiable answers. Importantly, SimpleQA was constructed to be "uncontaminated". Questions target obscure facts unlikely to appear in public training data, making it challenging for retrieval-based approaches. Both datasets are used in Chapter 6 to evaluate test set declassification. **PopQA** (Mallen et al., 2022) evaluates factual knowledge across entities of varying popularity, revealing how model accuracy correlates with entity frequency in training data.

Question Similarity and Paraphrase. **QuoraQP** contains 404,000 question pairs annotated for semantic equivalence, widely used for duplicate question detection, though most pairs lack associated answers. **QuoraQP-a** (Wang et al., 2020b) extends the original dataset by pairing questions with answers extracted from Quora threads, enabling research on answer-aware question matching. **PopQA-TP** (Rabinovich et al., 2023) is an extension of PopQA which contains 118,000 paraphrased questions organized into 14,000 clusters, enabling coherence evaluation across equivalent phrasings. This dataset is built upon the PopQA to test LLM coherence. The **Question Ranking Corpus (QRC)** (Campese et al., 2023), introduced in Chapter 3, provides 15,211 queries each paired with 30 candidate QA pairs annotated for semantic equivalence, the first large-scale resource structured for training and evaluating complete DBQA pipelines.

Text Corpora. Beyond QA datasets, several text corpora serve as knowledge sources for retrieval and answer reconstruction. **Wikipedia** provides encyclopedic coverage across virtually all domains and is the primary knowledge source for many QA systems. The retrieval index based on Wikipedia, consist of 30M of passages. **CCNews** (Wenzek et al., 2019) ($\approx$ 461M) provides a large corpus of news articles, useful for domains where encyclopedic sources lack coverage. These corpora are used in Chapter 6 for domain-appropriate answer reconstruction.

2.9 Evaluation Metrics

The evaluation of question answering and retrieval systems presents unique challenges that standard classification metrics do not fully address. A QA system may correctly identify that a question is about French geography, yet fail to surface the specific answer about Paris being the capital. Conversely, a retrieval model might return highly relevant documents on average while producing wildly inconsistent results for semantically equivalent queries. These characteristics motivate the usage of diverse set of metrics, each capturing different aspects of system behavior.

This section introduces the evaluation metrics used throughout the experimental chapters, organized by the aspect of performance they measure: ranking quality, answer correctness, and coherence.

2.9.1 Ranking Metrics

Ranking metrics evaluate how effectively systems order candidates by relevance. In retrieval and answer selection tasks, finding *some* relevant item is often insufficient as the users expect the best results to appear first.

Precision at k (P@k) measures the proportion of relevant items among the top-k results:

$$\text{P@}k = \frac{|\{\text{relevant items in top-}k\}|}{k}$$

P@1 is particularly important for QA systems, where users typically expect the single best answer to appear first. A system achieving high P@1 reliably places the correct answer at rank one.

Hit Rate at k (Hit@k) measures the proportion of queries for which at least one relevant item appears among the top-k retrieved results:

$$\text{Hit@}k = \frac{1}{|Q|} \sum_{q \in Q} \mathbb{1}(R_q \cap \text{TopK}(q) \neq \varnothing)$$

where Q is the set of queries, R_q is the set of relevant items for query q, and $\text{TopK}(q)$ denotes the set of the top-k retrieved items for q. Unlike P@k, which penalizes having fewer relevant items in the top-k, Hit@k only asks whether *any* relevant item was retrieved. This metric is useful for evaluating recall at different ranking depths and for systems where finding one good answer suffices.

Mean Reciprocal Rank (MRR) captures how quickly a system surfaces the first relevant result:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q}$$

where rank_q is the position of the first relevant item for query q. MRR rewards systems that place relevant items early: a relevant item at rank 1 contributes 1.0, at rank 2 contributes 0.5, at rank 10 contributes 0.1. This metric is appropriate when users scan results sequentially and stop upon finding a satisfactory answer.

Mean Average Precision (MAP) summarizes precision across all recall levels. For a single query, Average Precision computes precision at each position where a relevant item appears, then averages these values:

$$\text{AP} = \frac{1}{|\text{relevant}|} \sum_{k=1}^{n} \text{P@}k \cdot \mathbb{1}[\text{item at rank } k \text{ is relevant}]$$

MAP averages AP across all queries. Unlike MRR, which considers only the first relevant item, MAP rewards systems that rank *all* relevant items highly, making it suitable when multiple correct answers exist.

Normalized Discounted Cumulative Gain (NDCG) extends ranking evaluation to graded relevance judgments. Discounted Cumulative Gain applies a logarithmic discount to relevance scores based on position:

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

where rel_i is the relevance grade of the item at rank i. NDCG normalizes DCG by the ideal DCG achievable with perfect ranking, producing values in $[0, 1]$. Unlike binary relevance metrics, NDCG credits partially relevant items proportionally to their relevance grade.

Rank-Biased Overlap (RBO) (Webber et al., 2010) measures similarity between two ranked lists with configurable top-weighting:

$$\text{RBO}(\tau_1, \tau_2, p) = (1-p) \sum_{d=1}^{\infty} p^{d-1} \cdot A_d$$

where A_d is the overlap proportion at depth d and $p \in (0, 1)$ controls how much weight is given to top positions. RBO produces values in $[0, 1]$, where 1 indicates identical rankings. Unlike Spearman correlation, RBO naturally handles lists of different lengths and provides intuitive top-weighted comparison, making it particularly suitable for comparing retrieval results where early positions matter most.

Spearman Correlation (ρ) measures the monotonic relationship between two rankings by computing the Pearson correlation on rank values:

$$\rho = 1 - \frac{6 \sum_i d_i^2}{n(n^2 - 1)}$$

where d_i is the difference between ranks of item i in the two lists and n is the number of items. Values range from -1 (perfect inverse ranking) through 0 (no correlation) to $+1$ (identical ranking). While Spearman treats all positions equally, it provides a complementary view to RBO by measuring overall ranking agreement.

2.9.2 Answer Correctness Metrics

Beyond ranking, QA systems must produce correct answers. Different evaluation scenarios require different notions of correctness.

Exact Match (EM) checks whether the predicted answer exactly matches any reference answer after normalization (lowercasing, punctuation removal, article stripping):

$$\text{EM} = \begin{cases} 1 & \text{if normalize(prediction)} = \text{normalize(reference)} \\ 0 & \text{otherwise} \end{cases}$$

This strict binary metric is particularly appropriate for factoid questions with short, canonical answers.

F1 Score measures token-level overlap between predicted and reference answers, balancing precision and recall:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where Precision is the fraction of predicted tokens appearing in the reference, and Recall is the fraction of reference tokens appearing in the prediction. F1 provides partial credit for answers that overlap with the reference without matching exactly.

ROC-AUC (Area Under the Receiver Operating Characteristic Curve) evaluates binary classification performance across all decision thresholds. ROC-AUC measures the probability that a randomly chosen positive example is scored higher than a randomly chosen negative example:

$$\text{ROC-AUC} = P(\text{score}(x^+) > \text{score}(x^-))$$

Values range from 0.5 (random classifier) to 1.0 (perfect ranking of positives above negatives). ROC-AUC is particularly useful for imbalanced datasets like duplicate question detection, where it provides a threshold-independent measure of discriminative ability.

Accuracy measures the proportion of correctly answered questions:

$$\text{Accuracy} = \frac{|\{\text{correctly answered questions}\}|}{|\{\text{total questions}\}|}$$

For generative QA, determining correctness is non-trivial since answers may be correct but phrased differently from references. Human evaluation remains the gold standard but is expensive and slow. *LLM-as-judge* approaches (Zheng et al., 2023b) offer a scalable alternative: a capable LLM (e.g., GPT-4, Claude) is prompted to assess whether a generated answer is correct given the question and reference answer. The judge model receives instructions specifying evaluation criteria and outputs a binary judgment or graded score. While not perfect, LLM-as-judge correlates well with human judgments for factual QA and enables evaluation at scale. Care must be taken to avoid biases: LLM judges may favor verbose answers, prefer their own outputs, or exhibit position bias when comparing multiple responses (Wang et al., 2024c).

2.9.3 Coherence Metrics

Coherence metrics measure the consistency of system behavior across semantically equivalent inputs. A coherent system should produce similar outputs for questions that express the same information need, regardless of surface phrasing.

Semantic Coherence for generative models measures the similarity between answers produced for equivalent questions. Given a model $\mathcal{M}$, a cluster of equivalent questions $C = \{q_1, \ldots, q_n\}$, and a sentence embedding function $e(\cdot)$:

$$\text{Coherence}_{\text{sem}}(C, \mathcal{M}) = \frac{2}{n(n-1)} \sum_{i<j} \cos(e(\mathcal{M}(q_i)), e(\mathcal{M}(q_j))) \tag{2.4}$$

This metric computes the average pairwise cosine similarity of answer embeddings within a cluster. Higher values indicate more consistent answers across phrasings, regardless of whether those answers are correct.

Binary Coherence for factual QA with verifiable answers measures whether a model's correctness is consistent within a cluster of similar questions:

$$\text{Coherence}_{\text{bin}}(C, \mathcal{M}) = \frac{2}{n(n-1)} \sum_{i<j} \mathbb{1}[\text{correct}(\mathcal{M}(q_i)) = \text{correct}(\mathcal{M}(q_j))] \tag{2.5}$$

A value of 1.0 means the cluster is either entirely correct (*coherent-correct*) or entirely incorrect (*coherent-incorrect*). Values below 1.0 indicate *incoherent* behavior: the model answers some phrasings correctly while failing on semantically equivalent alternatives, suggesting it possesses the required knowledge but cannot reliably access it.

Ranking Coherence for retrieval models measures whether equivalent queries produce consistent document rankings. Given a retrieval function δ that returns top-k documents $\tau_\delta(q, k)$ for query q:

$$\text{Coherence}_{\text{rank}}(C, \delta) = \frac{2}{n(n-1)} \sum_{i<j} \text{RBO}(\tau_\delta(q_i, k), \tau_\delta(q_j, k)) \tag{2.6}$$

This metric computes the average RBO between rankings produced for all pairs of equivalent queries. Low ranking coherence indicates that small changes in query phrasing cause large changes in retrieved documents, suggesting fragile retrieval behavior.

2.10 Summary

This chapter provided comprehensive background on concepts and methods underlying this thesis.

We reviewed Question Answering systems from early rule-based approaches through modern neural architectures, with emphasis on Database-based QA (DBQA) systems that retrieve answers from pre-computed question-answer databases. DBQA offers advantages in factual accuracy, inference speed, and knowledge updateability that motivate their study. We examined question understanding and semantic similarity, exploring how questions are represented and compared, and the importance of recognizing equivalence despite lexical variation.

We discussed Transformer models and pre-training objectives, tracing evolution from token-level tasks through sentence-level contrastive learning to task-oriented pre-training. We identified gaps in existing methods for question ranking and reviewed specialized approaches that address these gaps. We examined dense retrieval methods from sparse lexical approaches through neural embeddings, emphasizing bi-encoder efficiency and cross-encoder accuracy

in retrieve-and-rerank pipelines. We discussed retrieval coherence and methods to improve consistency across equivalent queries.

We reviewed LLM coherence challenges, showing how models struggle with consistent answers to equivalent questions, and how question retrieval can improve coherence through redundant semantic signal. We examined question clustering methods and their applications including coherence improvement, feedback aggregation, and retrieval training.

We introduced dataset equivalence as a framework for understanding when different evaluation sets measure the same capabilities, enabling robust evaluation and dataset transformation. We reviewed privacy-preserving approaches to NLP data release including anonymization, synthetic generation, and retrieval-based transformation, providing context for declassification.

These foundations directly enable the contributions in subsequent chapters:

- **Chapter 3**: Building on DBQA architectures and dense retrieval methods, we develop a large-scale question-answer database retrieval system and introduce the Question Ranking Corpus for training and evaluation.
- **Chapter 4**: Extending task-oriented pre-training methods, we introduce specialized pre-training objectives leveraging question retrieval knowledge distillation.
- **Chapter 5**: Building on LLM coherence challenges and question clustering applications, we demonstrate how retrieved equivalent questions improve understanding and consistency, and how clusters improve document retrieval coherence.
- **Chapter 6**: Building on dataset equivalence concepts and privacy-preserving approaches, we develop a framework for transforming datasets through semantic question mapping.

Chapter 3

Question Retrieval from Large-Scale Question-Answer Databases

Understanding the semantics of questions is fundamental to building effective Question Answering systems. When a user poses a question, the system must recognize not just the surface form but also the underlying information need—and crucially, it must identify when two differently-worded questions express the same need. This chapter addresses the first core challenge of this thesis: how to efficiently retrieve semantically equivalent questions from massive databases containing millions of question-answer pairs.

The approach we develop here, which we call QUADRo (QUestion-Answer Database Retrieval), serves as the foundation for all subsequent contributions in this thesis. The question retrieval models trained in this chapter will be enhanced through specialized pre-training (Chapter 4), and the concept of pairwise question equivalence introduced here will be extended to question clusters that enable LLM grounding and coherent document retrieval (Chapter 5).

The core insight motivating this work is that many user questions have been asked before, possibly in different formulations. Rather than generating answers from scratch or extracting them from unstructured documents, Database-based Question Answering (DBQA) systems can efficiently retrieve pre-computed answers by finding semantically equivalent questions in a curated database. To illustrate, consider a user asking "*What are the side effects of ibuprofen?*". Our system retrieves the stored question "*Ibuprofen side effects?*" paired with a pre-verified answer listing common adverse reactions, bypassing the need for answer generation. The challenge is recognizing this equivalence across millions of candidates with sub-second latency. This paradigm offers several advantages: guaranteed factual accuracy (since answers are pre-verified), low inference latency (retrieval is faster than generation), easy knowledge updates (adding new pairs without retraining), and transparent source attribution.

However, the development of effective DBQA systems has been hindered by a critical gap: the lack of large-scale annotated resources specifically designed for training and evaluating question retrieval and ranking models. Existing resources for question similarity, such as Quora Question Pairs (Wang et al., 2020b) or CQADupStack (Hoogeveen et al., 2015), focus on pairwise classification rather than ranking, and crucially, they do not incorporate answer quality into the annotation process.

This chapter makes three main contributions:

1. **A Large-Scale Question-Answer Database**: We construct a database of 6.3 million

question-answer pairs from diverse high-quality sources, providing broad coverage for open-domain question answering and enabling retrieval at scale.

2. **The Question Ranking Corpus (QRC)**: We introduce a novel annotated dataset of 15,211 queries, each associated with 30 candidate question-answer pairs annotated for semantic equivalence. This results in approximately 443,000 annotated examples specifically designed for training and evaluating question retrieval models.

3. **Comprehensive Experimental Analysis**: We conduct extensive experiments demonstrating that (i) incorporating answers during retrieval and ranking substantially improves accuracy, (ii) neural retrieval methods are essential for semantic matching at scale, achieving up to +19% over BM25, and (iii) our system achieves competitive performance with web-based QA and Large Language Models while offering superior efficiency.

The remainder of this chapter is organized as follows. Section 3.1 formalizes the question retrieval task and defines semantic equivalence. Section 3.2 describes the construction of our question-answer database. Section 3.3 presents the system architecture. Section 3.4 details the annotation process for the Question Ranking Corpus. Section 3.5 reports our experimental results. Section 3.6 discusses findings and limitations. Section 3.7 concludes the chapter.

3.1 Problem Formulation

3.1.1 Task Definition

Given a user question q (the *query*) and a database $\mathcal{D} = \{(q_1, a_1), (q_2, a_2), \ldots, (q_N, a_N)\}$ containing N question-answer pairs, the DBQA task consists of:

1. **Retrieval**: Identify a subset $\mathcal{R} \subseteq \mathcal{D}$ of k candidate pairs most likely to contain an answer to q

2. **Ranking**: Reorder the candidates in $\mathcal{R}$ such that the pair (q^*, a^*) with highest semantic equivalence to q is ranked first

3. **Answer Selection**: Return a^* as the answer to the user query

The success of this pipeline depends critically on the ability to recognize when two questions are semantically equivalent, that is, when they express the same information-seeking intent and accept the same answers.

3.1.2 Semantic Equivalence

As discussed in Section 2.2.2, semantic equivalence between questions requires two jointly necessary conditions: the questions must (i) express the same information-seeking intent and (ii) accept the same set of correct answers. We now formalize this for the DBQA setting, where answer correctness provides a natural operationalization of condition (ii) and where the retrieval model's learned representations implicitly enforce condition (i).

Definition 3.1.1 (Semantic Equivalence)**.** *Two questions q_i and q_j are **semantically equivalent**, denoted $q_i \leftrightarrow q_j$, if and only if they express the same information-seeking intent and accept the same set of correct answers:*

$$q_i \leftrightarrow q_j \iff \mathit{intent}(q_i) = \mathit{intent}(q_j) \ \wedge \ \forall a : \ell(q_i, a) \Leftrightarrow \ell(q_j, a)$$

where $\ell(q, a) = 1$ if answer a is correct for question q, and $\ell(q, a) = 0$ otherwise.

The intent condition prevents coincidental answer overlaps from producing false equivalences. For instance, "*When was Shakespeare born?*" and "*When was Galileo born?*" both have "1564" as a correct answer, but they are not equivalent since they seek information about different entities and different events. This case arises especially with low-cardinality answers (dates, common named entities, yes/no answers) and must be ruled out explicitly. The answer condition, in turn, grounds the definition in observable, task-relevant behavior and prevents purely intent-based judgments from conflating questions that happen to sound similar but require different answers.

In practice, the intent condition is enforced at two levels. During *annotation*, human annotators in our Question Ranking Corpus (Section 3.4) were shown both questions and answers, enabling them to verify intent equivalence jointly with answer compatibility, reducing annotation error by 45% compared to question-only annotation. During *inference*, the neural retrieval and reranking models (Section 3.3) learn representations that encode semantic intent: questions with coincidentally overlapping answers but different intents receive low similarity scores because their learned embeddings reflect different information needs.

The combined definition has several important properties. It is *symmetric*: if $q_i \leftrightarrow q_j$, then $q_j \leftrightarrow q_i$, unlike document retrieval where the query-document relationship is inherently asymmetric. It is *transitive*: if $q_i \leftrightarrow q_j$ and $q_j \leftrightarrow q_k$, then $q_i \leftrightarrow q_k$, enabling the formation of equivalence classes (clusters) of questions. It is *task-oriented*: by grounding equivalence in both intent and answer correctness, the definition captures what matters for QA system performance. Surface similarity between questions is neither necessary nor sufficient for equivalence: "*Who wrote Hamlet?*" and "*Who played Hamlet?*" share most words but require different answers (Shakespeare vs. various actors), while "*Who wrote Hamlet?*" and "*The author of Hamlet is?*" look different but are equivalent. Finally, equivalence is *context-sensitive*: both the intent function and the correctness function ℓ may depend on context, so "*Who is the president?*" may have different correct answers depending on the country and time period implied.

3.1.3 Challenges

DBQA systems face several challenges that motivate our design choices:

- **Lexical Variation:** users express the same information need using vastly different vocabulary and syntactic structures. For example, *"What is the capital of France?"* and *"Name the city that serves as France's seat of government"* are equivalent but share minimal lexical overlap.

- **Scale:** practical DBQA systems must search databases that contain millions of question-answer pairs. This requires efficient approximate search methods rather than exhaustive comparison.

- **Answer Disambiguation:** questions may be ambiguous in isolation but disambiguated by their answers. For instance, *"How long does it take?"* is underspecified, but when paired with an answer about cooking times versus travel times, the intended meaning becomes clear.
- **Quality Variation:** database entries may vary in answer quality. Some pairs may contain incorrect, outdated, or incomplete answers that should be ranked lower even if the questions appear similar.

3.2 Question-Answer Database Construction

A key contribution of this work is the construction of a large-scale, high-quality database of question-answer pairs that enables open-domain question answering through retrieval.

3.2.1 Design Principles

Our database construction follows three guiding principles:

- **Quality over Quantity:** while scale is important for coverage, we prioritize answer correctness. Each source is evaluated for quality, and filtering mechanisms are applied to remove likely incorrect pairs.
- **Diversity:** we combine multiple sources spanning different domains, question types, and answer styles to ensure broad applicability.
- **Verifiability:** where possible, we use sources with human annotations or verifiable answers, reducing reliance on automatically generated content.

3.2.2 Data Sources

Our database aggregates question-answer pairs from multiple high-quality sources, summarized in Table 3.1.

Labeled Sources

These sources contain question-answer pairs with human annotations or verified answers:

GooAQ: provides questions derived from Google autocomplete suggestions paired with answers extracted from featured snippets. The automated extraction process yields high-quality pairs due to Google's answer selection mechanisms (Khashabi et al., 2021).

WQA: contains questions from web search queries with extracted answer passages from authoritative sources (Zhang et al., 2021).

WikiAnswer: aggregates questions and answers from the WikiAnswers community platform, where users provide answers to submitted questions (Fader et al., 2013).

CovidQA: is a domain-specific dataset containing questions about COVID-19 with expert-verified answers from scientific sources (Möller et al., 2020).

HotpotQA: provides multi-hop reasoning questions with short factual answers derived from Wikipedia (Yang et al., 2018).

Augmented Sources

For some sources, answers were not directly available and required extraction or filtering:

Quora Match: Quora Match contains a small fraction of the original Quora dataset for which the answers are available directly from the Quora threads. The answers were selected through a heuristic approach based on the rank of users' content in Quora threads (Wang et al., 2020b).

Quora Question Pairs: The original QuoraQP (Iyer et al., 2017) dataset contains question pairs from Quora without answers. First we remove the queries for which the answers were available in Quora Match, then we extracted answers using a two-stage process:

1. For each question, we queried a 2020 CommonCrawl snapshot using BM25 and retrieved the 200 most relevant documents.
2. We split documents into sentences and applied a state-of-the-art answer sentence selector (Lauriola and Moschitti, 2021a) to identify the best answer passage.
3. We retained only pairs where the selector confidence exceeded the 90th percentile, ensuring high answer quality.

Manual evaluation of 200 randomly sampled pairs from this augmented set showed 93.0% answer correctness.

ELI5 (Explain Like I'm 5): (Fan et al., 2019) contains questions from Reddit's explainlikeimfive subreddit with community-voted answers. We retained only the top 50% of pairs ranked by our answer selector to filter low-quality responses, achieving an estimated 84.3% accuracy based on manual annotation.

Table 3.1: Statistics of the QUADRo database. QA = number of question-answer pairs, Q = unique questions, Q length and A length report mean ± standard deviation in tokens.

Source	QA Pairs	Unique Q	Q Length	A Length
Labeled Sources				
GooAQ (Khashabi et al., 2021)	3.1M	2.9M	9.1 ± 2.3	45.9 ± 18.9
WQA (Zhang et al., 2021)	391K	80.5K	7.5 ± 3.2	24.8 ± 11.3
WikiAnswer (Fader et al., 2013)	2.3M	2.3M	9.1 ± 2.5	60.3 ± 117.3
CovidQA (Möller et al., 2020)	2K	1.9K	10.6 ± 4.1	15.8 ± 17.1
HotpotQA (Yang et al., 2018)	64K	64K	20.4 ± 10.6	4.1 ± 2.4
Augmented Sources				
Quora Match (Wang et al., 2020b)	230K	170K	12.5 ± 6.7	38.8 ± 20.5
QuoraQP (Iyer et al., 2017)	219K	134K	9.6 ± 2.9	25.1 ± 10.8
ELI5 (Fan et al., 2019)	58.9K	58.7K	17.6 ± 9.1	60.7 ± 28.1
Total	**6.3M**	**5.7M**	$\mathbf{12.1 \pm 4.3}$	$\mathbf{34.4 \pm 19.2}$

3.2.3 Quality Considerations

We deliberately excluded certain large-scale resources due to quality concerns:

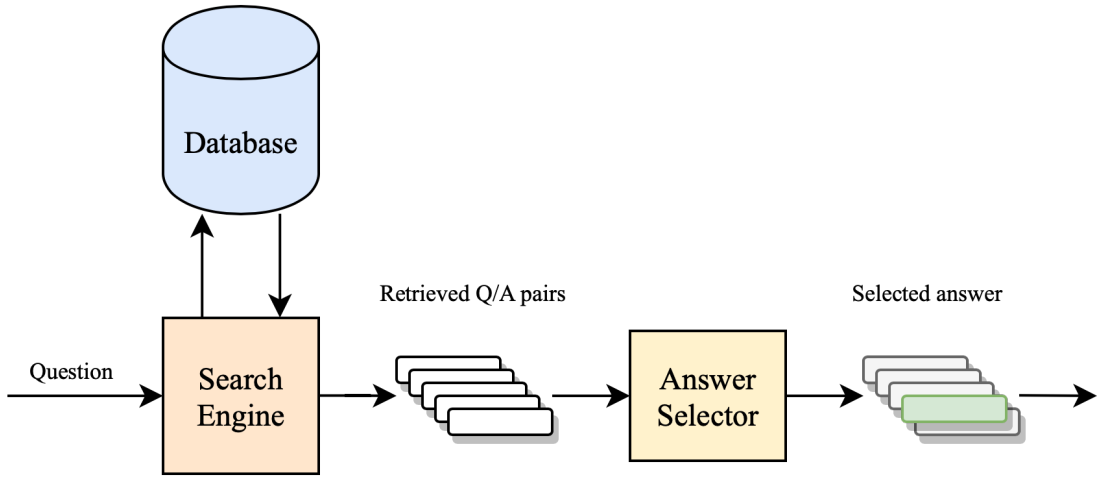

Figure 3.1: QUADRo system architecture. Given a user query, the neural search engine retrieves top-k similar question-answer pairs from the database. The answer selector reranks candidates and returns the best answer.

PAQ (Probably Asked Questions): contains 65 million automatically generated question-answer pairs. While valuable for some applications, the authors report only 82% correctness, and the questions often exhibit unnatural phrasing due to the generation process (Lewis et al., 2021). We excluded PAQ from our database to prioritize quality but note that it could be integrated for applications where coverage is prioritized over precision.

3.2.4 Database Statistics

The final database contains 6.3 million question-answer pairs covering 5.7 million unique questions. Questions average 12 tokens in length, while answers range from short factual responses (4 tokens in HotpotQA) to detailed explanations (60+ tokens in WikiAnswer and ELI5). This diversity ensures the database can support both factoid and explanatory question answering.

3.3 System Architecture

As described in Section 3.1.1, QUADRo implements a two-stage retrieve-and-rerank architecture, illustrated in Figure 3.1. This design balances efficiency (fast retrieval over millions of pairs) with accuracy (precise reranking of top candidates).

3.3.1 Neural Search Engine

The retrieval component employs a bi-encoder architecture based on Sentence Transformers (Reimers and Gurevych, 2019a), enabling efficient similarity search over millions of database entries.

Architecture

The bi-encoder consists of two Transformer encoders with shared weights that independently map texts to fixed-dimensional embeddings:

Query Encoder: Given a user query q, the encoder produces an embedding:

$$\mathbf{q} = \text{Enc}(q)$$

Database Encoder: For each database entry (q_i, a_i), we encode the concatenated question-answer pair:

$$\mathbf{d}_i = \text{Enc}(q_i \oplus a_i)$$

where $\oplus$ denotes concatenation with a separator token.

The similarity between the query and a database entry is computed as cosine similarity:

$$\text{sim}(q, (q_i, a_i)) = \frac{\mathbf{q}^\top \mathbf{d}_i}{\|\mathbf{q}\|\|\mathbf{d}_i\|}$$

In our implementation, $\text{Enc}(\cdot)$ uses mean pooling over token representations from a RoBERTa encoder.[1]

Input Configurations

As the key research question is what information to encode for optimal question retrieval. We investigate three configurations:

Question-Question (QQ): The query encoder encodes only the query q, and the database encoder encodes only questions q_i (ignoring answers). This tests whether question similarity alone suffices for retrieval.

Question-Question+Answer (QQA): The query encoder encodes only the query q, while the database encoder encodes concatenated question-answer pairs (q_i, a_i). This tests whether answer context improves retrieval without modifying the query representation.

Question+Answer-Question+Answer (QAQA): Both encoders receive question-answer pairs. This is not applicable at inference time as the user provides only a question, but it serves as an oracle to measure the information gain from answer availability.

Efficient Search

At inference time, all database embeddings $\{\mathbf{d}_i\}_{i=1}^N$ are pre-computed and indexed. Given a new query, we compute $\mathbf{q}$ and retrieve the top-k most similar entries using a full search. We employ FAISS (Johnson et al., 2019a) with Flat indexing on multiple GPUs, enabling for fast retrieval over millions of entries.

[1] Specifically, we use the Sentence-RoBERTa architecture with mean pooling, though other pooling strategies (e.g., [CLS] token) yield similar results.

Continuous Pre-training

The bi-encoder is initialized from a public Sentence-RoBERTa-base checkpoint and we continuously pre-trained on a large collection of datasets for unsupervised STS tasks, including paraphrasing, sentence similarity, question answering, and summarization. This pre-training develops robust semantic matching capabilities across diverse domains before specialization on question retrieval. The pre-training corpus includes:

- **Question Answering**: MS MARCO (Nguyen et al., 2016), Natural Questions (Kwiatkowski et al., 2019), PAQ (Lewis et al., 2021)
- **Scientific Text**: S2ORC (Semantic Scholar Open Research Corpus) (Lo et al., 2020), Specter (Cohan et al., 2020)
- **Natural Language Inference**: SNLI/MultiNLI (Bowman et al., 2015)
- **Summarization**: CNN/DailyMail (See et al., 2017), XSum (Narayan et al., 2018), WikiHow (Koupaee and Wang, 2018)
- **Other**: AmazonQA (Gupta et al., 2019), ELI5 (Fan et al., 2019), FEVER (Thorne and et al., 2018), Flickr30K (Young et al., 2014), GooAQ (Khashabi et al., 2021), CodeSearchNet (Husain et al., 2020), SimpleWiki (Coster and Kauchak, 2011), StackExchange (Narayan et al., 2018), COCO Captions (Lin et al., 2014), SQuAD (Rajpurkar et al., 2016), and Altex (Hidey and McKeown, 2016)

These datasets consist of pairs of semantically equivalent texts (e.g., question-answer, title-abstract, paraphrases). Overall, the pre-training data includes approximately 180 million positive text pairs and 17.5 million hard negatives where available.

We train with the Multiple Negatives Ranking (MNR) loss (Henderson et al., 2017). Given a batch of B query-positive pairs $\{(q_j, d_j^+)\}_{j=1}^{B}$, the loss treats other batch elements as in-batch negatives:

$$\mathcal{L}_{\text{MNR}} = -\frac{1}{B}\sum_{j=1}^{B} \log \frac{\exp\Big(\text{sim}(q_j, d_j^+)/\tau\Big)}{\sum_{k=1}^{B} \exp\big(\text{sim}(q_j, d_k^+)/\tau\big)} \tag{3.1}$$

where τ is a temperature parameter and $\text{sim}(\cdot,\cdot)$ denotes cosine similarity. Pre-training uses batch size 384, maximum sequence length 256 tokens, and AdamW optimizer with learning rate 2×10^{-5}.

Fine-tuning

After continuous pre-training, we fine-tune the model for question retrieval using a two-stage strategy:

Stage 1: QuoraQP. We first fine-tune on QuoraQP, a dataset of duplicate question detection. To adapt this resource for our setting, we augment each question pair with artificially generated answers, creating question-answer entries that mirror our database structure. This intermediate step exposes the model to question equivalence patterns before training on our more challenging corpus.

Stage 2: Question Ranking Corpus. We then fine-tune on the Question Ranking Corpus (QRC), a large-scale dataset we constructed specifically for this task (described in detail in Section 3.4). QRC contains approximately 443,000 annotated triplets across 15,211 queries, where each query is paired with 30 candidate question-answer pairs annotated for semantic equivalence. A critical aspect of this stage is training with *hard negatives*: database entries that are semantically similar but not equivalent to the query, such as questions sharing topical overlap but with different information-seeking intents. Differently from random negatives, which are trivially distinguishable, hard negatives force the model to learn fine-grained semantic distinctions essential for accurate retrieval.

Both stages use MNR loss. Hyperparameters are selected via grid search on the development split, with early stopping after 2 consecutive epochs of validation loss degradation. We evaluate learning rates in $\{5 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$, batch sizes in $\{64, 128, 256, 384\}$, and sequence lengths in $\{128, 256\}$.

3.3.2 Answer Selector (Reranker)

While the bi-encoder enables efficient retrieval over millions of candidates, its dual-encoder architecture has an inherent limitation: *query and candidate are encoded independently*, preventing the model from capturing fine-grained interactions between them. The query encoder cannot "see" the candidate text during encoding, and vice versa. This independence assumption, while enabling pre-computation of candidate embeddings, limits the model's ability to recognize subtle semantic relationships that depend on the specific pairing of query and candidate.

To address this limitation, we adopt the *retrieve-and-rerank* paradigm that has proven effective across information retrieval tasks. The bi-encoder first retrieves a manageable set of candidates (e.g., top-500), then a more expressive *cross-encoder* reranks these candidates by jointly modeling query-candidate interactions. This two-stage approach combines the efficiency of bi-encoders for initial retrieval with the effectiveness of cross-encoders for precise ranking.

In the context of question retrieval, the reranking stage faces a unique challenge: determining semantic equivalence requires understanding not just the surface similarity between questions, but also whether they would be satisfied by the same answer. Two questions might use very different vocabulary yet seek identical information (e.g., *"What's the boiling point of water?"* vs. *"At what temperature does H_2O transition to gas?"*). Conversely, superficially similar questions might have distinct intents (e.g., *"How tall is the Eiffel Tower?"* vs. *"How tall was Eiffel?"*).

Our answer selector leverages on a key insight: *the pre-computed answers in the database provide valuable context for assessing question equivalence.* If a candidate's answer would appropriately address the user's query, this is strong evidence that the questions are equivalent. We therefore design our cross-encoder to incorporate the answer as an additional input signal, enabling three-way interactions between query, database question, and answer.

Architecture

The cross-encoder jointly encodes the query q and each candidate (q_i, a_i) in a single forward pass. The input is formatted as a single sequence and processed by a Transformer encoder ψ:

$$\mathbf{h} = \psi([\texttt{CLS}]\ q\ [\texttt{SEP}]\ a_i\ [\texttt{SEP}]\ q_i\ [\texttt{EOS}])$$

The [CLS] representation is projected to two classes (equivalent / not equivalent) through a linear layer:

$$\text{logits}(q, q_i, a_i) = \mathbf{W}\mathbf{h}_{[\text{CLS}]} + \mathbf{b}$$

where $\mathbf{W} \in \mathbb{R}^{2\times d}$ and $\mathbf{b} \in \mathbb{R}^2$ are learned parameters. The input ordering places the answer between the query and database question, following findings from answer sentence selection that this arrangement optimizes attention patterns (Lauriola and Moschitti, 2021a).

At inference time, we apply softmax and use the probability of the positive class (equivalent) as the ranking score:

$$\text{score}(q, q_i, a_i) = \frac{\exp(\text{logits}_1)}{\exp(\text{logits}_0) + \exp(\text{logits}_1)}$$

Input Configurations

We investigate multiple input configurations for the reranker:

- **QQ (Query-Question):** encode only (q, q_i), testing pure question similarity.
- **QA (Query-Answer):** encode only (q, a_i), testing direct query-answer relevance without the database question.
- **QQA (Query-Question-Answer):** encode (q, q_i, a_i) with the answer as context for the question pair.
- **QAQ (Query-Answer-Question):** encode (q, a_i, q_i) with the database question as context for query-answer matching.

Training

The reranker is trained with cross-entropy loss on annotated triplets (q, q_i, a_i, y) where $y \in \{0, 1\}$ indicates semantic equivalence:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|\mathcal{T}|} \sum_{(q,q_i,a_i,y)\in\mathcal{T}} \log P(y \mid q, q_i, a_i)$$

where $P(y \mid q, q_i, a_i) = \text{softmax}(\text{logits}(q, q_i, a_i))_y$ is the predicted probability of the true class.

We initialize from an ELECTRA-base checkpoint pre-trained on ASNQ (Garg et al., 2020) for answer sentence selection (AS2) (Lauriola and Moschitti, 2021a). Following the two-stage fine-tuning strategy described in Section 3.3.1, we first fine-tune on QuoraQP, then on the Question Ranking Corpus. Hyperparameters are selected via grid search: learning rates in $\{5 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$, batch sizes in $\{64, 128, 1024\}$, with sequence length fixed at 256 tokens.

3.3.3 End-to-End Pipeline

As illustrated in Figure 3.1, QUADRo combines efficient dense retrieval with precise cross-encoder reranking. At inference time, the system operates through four stages:

Stage 1: Query Encoding. The user query q is encoded by the bi-encoder into a dense vector $\mathbf{q} \in \mathbb{R}^d$. This encoding is performed online for each incoming query.

Stage 2: Candidate Retrieval. The query embedding is compared against pre-computed embeddings of all database entries via exhaustive cosine similarity search. Despite the database size (6.3 million entries), modern GPU hardware enables efficient brute-force search over dense embeddings. This stage retrieves the top-k most similar candidates (we use $k = 500$ in our experiments).

Stage 3: Candidate Reranking. Each retrieved candidate (q_i, a_i) is scored by the cross-encoder, which jointly encodes the query and candidate to capture fine-grained semantic interactions. While more expensive than bi-encoder scoring, running the cross-encoder on only k candidates (rather than millions) keeps latency manageable.

Stage 4: Answer Selection. The answer a^* associated with the highest-ranked candidate is returned to the user:

$$a^* = a_{\arg\max_i \text{score}(q,q_i,a_i)}$$

3.4 Question Ranking Corpus

A major contribution of this work is the Question Ranking Corpus (QRC), a large-scale annotated dataset specifically designed for training and evaluating DBQA models.

3.4.1 Motivation

Existing resources for question similarity have several limitations for DBQA:

Pairwise vs. Ranking: Datasets like QuoraQP provide binary labels for question pairs but do not support ranking evaluation. DBQA requires ranking multiple candidates, not just classifying pairs.

Answer Exclusion: Most question similarity datasets do not include answers, preventing models from learning answer-aware representations and preventing annotators from using answers for disambiguation.

Easy Negatives: Random negative sampling produces trivially distinguishable pairs and/or false samples. Effective DBQA training requires hard negatives that challenge the model's semantic understanding.

3.4.2 Dataset Construction

The construction of the Question Ranking Corpus followed a four-step pipeline designed to produce challenging, high-quality annotations with hard negatives that require fine-grained semantic understanding.

Step 1: Query Sampling. We randomly sampled 15,211 questions from our database to serve as input queries. These questions were removed from the database to prevent trivial exact matches during retrieval and evaluation. The queries span diverse topics and formulations, reflecting the open-domain nature of the target application.

Step 2: Candidate Retrieval. For each query, we retrieved the top-$k = 30$ most similar question-answer pairs using a preliminary version of our retrieval model: Sentence-RoBERTa trained on QuoraQP with QQA configuration. This ensures candidates include challenging hard negatives that require fine-grained semantic understanding to be distinguished.

Step 3: Crowdsourced Annotation. We employed Amazon Mechanical Turk to annotate each query-candidate pair. Annotators judged whether the query q and retrieved question q_i were semantically equivalent according to Definition 3.1.1. Each triplet (q, q_i, a_i) was labeled by two independent annotators; in case of disagreement, a third annotator provided the tie-breaking vote.

Step 4: Quality Control. We implemented multiple mechanisms to ensure annotation quality: (i) restricting to Master Turkers with $\geq$95% approval rate and 100+ approved tasks; (ii) including control questions with known labels in each batch; (iii) excluding annotators who failed more than 10% of controls. Annotators were compensated \$0.15 per task (7 triplets including 2 controls). The resulting annotations show strong agreement: the two primary annotators agreed on 78% of triplets (random labeling would yield 50%), with Cohen's Kappa of 0.875 indicating high inter-annotator reliability.

Answer-Guided Annotation

A distinctive feature of our annotation protocol is that annotators were shown the answer a_i alongside the questions. We provided the following guidance:

> *Two questions are equivalent if they (i) have the same meaning AND (ii) share the same answers. Use the provided answer to help determine if both questions would accept this answer as correct.*

This answer-guided approach reduces ambiguity and improves consistency, as the answer provides context that disambiguates underspecified questions. We provided detailed examples to train annotators on in Table 3.2.

We also analyzed representative annotation cases to illustrate the task's inherent complexity, including easy positives, answer-dependent decisions, and annotation errors in both directions. These examples are provided in Appendix A.1.

3.4.3 Dataset Statistics

The annotation process yielded approximately 443000 labeled triplets distributed across 15,211 queries (Table 3.4). We allocated 11,711 queries for training, 1500 for development, and 2000 for testing. For development and test splits, we annotated exactly 30 candidates per query; training queries have a variable number of candidates (28.9 ± 10.3 on average) due to filtering of low-confidence annotations.

The dataset exhibits several properties that make it particularly challenging and realistic:

Answerability: Not all queries can be answered from the database. We find that 75.4% of queries have at least one equivalent question among their top-30 candidates. This reflects realistic deployment conditions where some user questions may not have pre-existing answers.

Table 3.2: Explained examples used during annotators training.

Positive Examples	Negative Examples
Query: Can a cat and a dog get along? **Question**: Do cats like the company of dogs and in the other way around? **Answer**: If you are lucky, your cat and dog can become friends within a couple of hours. But that won't usually happen. It takes time for cats to adapt to the dogs and similarly for the dogs to learn how to behave around cats. **Explanation**: *These questions are both asking if Cats and dogs can be friends. The Answer for the Question is also correct for the Query*	**Query**: Who did kill Brutus? **Question**: Who did Brutus kill? **Answer**: Brutus was one of the leaders of the conspiracy that assassinated Julius Caesar **Explanation**: *These questions are not asking for the same thing. Moreover, the Answer for the Question is not correct for the Query*
Query: Can a person fall in love with another person while he/she is already in love? **Question**: Is it possible for people to love 2 person at the same time? **Answer**: It is possible to love and be intimate with more than one person at a time. **Explanation**: *These questions are both asking if loving 2 people at the same time is possible. The Answer for the Question is correct for both Question and Query.*	**Query**: What is the best restaurant in LA ? **Question**: What is the best dish of the best restaurant in LA? **Answer**:The best dish of the best restaurant in LA is Lobster Rolls **Explanation**: *Those questions are not asking for the same thing. The query asks for a restaurant while the Question asks for a dish. Moreover, the Answer is not correct for the Query*

Table 3.3: Explained examples used during annotators training.

Hard Negatives: On average, each query has 5.8 positive (equivalent) and 24.2 negative candidates. Crucially, these negatives are *hard negatives*: questions retrieved based on semantic similarity that nonetheless have different meanings. This is far more challenging than random negative sampling and essential for training discriminative models.

Class Imbalance: The positive rate ranges from 16% (test) to 21% (train), reflecting the inherent difficulty of finding truly equivalent questions among semantically similar candidates. This imbalance mirrors real-world retrieval scenarios.

3.4.4 Comparison with Existing Resources

We compare the Question Ranking Corpus with existing question similarity datasets in Table 3.5. While several resources exist for question similarity tasks, they each lack one or more properties essential for training effective DBQA models. QuoraQP (404K pairs) is widely used for duplicate question detection but provides only pairwise labels without ranking setup or answers. CQADupStack (48K pairs) includes answers for some questions but is domain-specific (StackExchange forums) and uses random negatives. SemEval-2016 Task 3 supports ranking and includes answers but is limited in size (4K pairs) and domain coverage. WikiAnswers (30M clusters) offers scale but lacks answer quality control and ranking annotations. Finally, PopQA-TP (14K clusters) provides paraphrased questions but is limited to a closed

Table 3.4: Question Ranking Corpus statistics across data splits.

Split	Queries	Candidates/Query	Positive Rate
Train	11,711	28.9 ± 10.3	21.1%
Dev	1,500	30	16.0%
Test	2,000	30	15.7%
Total	15,211		

Table 3.5: Comparison of question similarity datasets. QRC uniquely combines ranking setup, answer availability, hard negatives, and open-domain coverage.

Dataset	Size	Ranking	Answers	Hard Neg.	Open Domain
QuoraQP	404K	✗	✗	✗	✓
CQADupStack	48K	✗	Partial	✗	✗
SemEval-2016	4K	✓	✓	✗	✗
WikiAnswers	30M	✗	✓	✗	✓
PopQA-TP	14K	✗	✓	✗	✗
QRC (Ours)	443K	✓	✓	✓	✓

domain.

In contrast, the QRC is the first large-scale resource that combines all desirable properties for DBQA research: (i) a ranking setup with multiple candidates per query, (ii) high-quality answers shown during annotation, (iii) challenging hard negatives from neural retrieval, and (iv) open-domain coverage across diverse topics.

3.5 Experiments

We conduct extensive experiments to validate the QUADRo system and the Question Ranking Corpus. Our evaluation addresses several key research questions reported in Table 3.6.

Table 3.6: Research questions for question retrieval from large-scale databases.

RQ	Research Question	Section
RQ1	Does training on QRC improve retrieval and reranking performance compared to existing resources?	§3.5.2, §3.5.3
RQ2	Do answers provide useful signal for question retrieval, and how should they be incorporated?	§3.5.2
RQ3	What is the optimal input configuration for cross-encoder reranking?	§3.5.3
RQ4	How does the end-to-end QUADRo pipeline compare against alternative QA approaches?	§3.5.5

We first describe our experimental setup, then present results for retrieval (Section 3.5.2) and reranking (Section 3.5.3) components individually, followed by end-to-end evaluation (Section 3.5.4) and comparison with alternative systems (Section 3.5.5).

3.5.1 Experimental Setup

Models and Training

We evaluate the QUADRo components described in Section 3.3: the bi-encoder retriever (Sentence-RoBERTa) and cross-encoder reranker (ELECTRA). Both models follow the training procedure outlined in Sections 3.3.1–3.3.2, with hyperparameters selected via grid search on the development set.

The choice of RoBERTa for retrieval and ELECTRA for reranking was driven by preliminary experiments. For reranking, ELECTRA outperforms RoBERTa by +3.3 P@1, +1.9 MAP, and +2.5 MRR in the QAQ configuration. For retrieval, Sentence-RoBERTa achieves 85.4 Pearson and 85.1 Spearman correlation on STS Benchmark, compared to 74.8/75.1 for Sentence-ELECTRA and 76.4/77.3 for Sentence-DeBERTa-V3.

3.5.2 Retrieval Results

We evaluate using standard ranking metrics as defined in Chapter 2: P@1 (Precision at 1), MAP (Mean Average Precision), and MRR (Mean Reciprocal Rank), with P@1 as the primary accuracy measure. For end-to-end evaluation, we additionally report Hit@k and accuracy based on manual annotation.

We first evaluate the bi-encoder retrieval component in isolation. The key question is whether incorporating answers into the database representation improves retrieval quality. We compare two configurations: QQ, which encodes only the database question, and QQA, which concatenates the question with its answer. Both configurations encode the user query identically (question only).

Table 3.7 presents the results on the QRC test set. The QQA configuration substantially outperforms QQ across all metrics, with +5.0 P@1 absolute improvement. This confirms that answers provide valuable semantic signal for question matching. In this scenario, the answers act as a "bridge" that helps identify equivalent questions even when the surface forms differ significantly. For completeness in the table is also reported the performance of the model used to build the dataset.

Table 3.7: Retrieval model performance on QRC test set. (*) This model is the one used to build the dataset.

Configuration	P@1	MAP	MRR
*S-RoBERTa$_{\text{QQA}}$	39.1	39.1	50.4
S-RoBERTa$_{\text{QQ}}$	43.4	41.6	52.9
S-RoBERTa$_{\text{QQA}}$	48.4 ± 0.4	45.6 ± 0.4	58.3 ± 0.4

3.5.3 Reranking Results

The cross-encoder reranker operates on the top-k candidates returned by retrieval. Unlike the bi-encoder, the cross-encoder can jointly model interactions between query, database question, and answer. We evaluate four input configurations to understand how best to leverage this capacity:

- **QA**: Query and answer only (ignoring the database question)

- **QQ**: Query and database question only (ignoring the answer)
- **QQA**: Query, database question, then answer
- **QAQ**: Query, answer, then database question

We also compare against QP-RoBERTa, the state-of-the-art cross-encoder for question pair similarity trained on Quora Question Pairs. Table 3.8 presents results on the QRC test set.

Table 3.8: Reranking model performance on QRC test set. †State-of-the-art cross-encoders for question pairs (Reimers and Gurevych, 2019a).

Configuration	P@1	MAP	MRR
$\text{ELECTRA}_{\text{QA}}$	37.1 ± 0.6	40.4 ± 0.2	49.5 ± 0.3
$\text{ELECTRA}_{\text{QQ}}$	50.0 ± 0.2	47.7 ± 0.3	59.5 ± 0.2
$\text{ELECTRA}_{\text{QQA}}$	49.3 ± 0.2	47.6 ± 0.1	59.2 ± 0.1
$\text{ELECTRA}_{\text{QAQ}}$	**50.8** ± 0.2	**48.4** ± 0.1	**60.2** ± 0.1
$\text{QP-RoBERTa}_{\text{base}}$†	43.5	41.8	54.4
$\text{QP-RoBERTa}_{\text{large}}$†	45.6	43.5	56.0

Results highlight the following key observations:

Answer Position Matters. The QAQ configuration (query-answer-question) outperforms QQA (query-question-answer) by +1.5 P@1. This suggests that placing the answer between the query and database question allows the model to first assess query-answer relevance, then verify question-question equivalence.

Question-Only Underperforms. The QA configuration (ignoring the database question entirely) performs worst, confirming that both the database question and answer contribute to accurate ranking.

Superior to Prior Art. Our models substantially outperform prior state-of-the-art cross-encoders for question pairs (QP-RoBERTa), demonstrating the value of training on QRC with answer-aware inputs.

3.5.4 End-to-End Pipeline Performance

Having evaluated retrieval and reranking components individually, we now assess the complete QUADRo pipeline. The end-to-end system combines $\text{S-RoBERTa}_{\text{QQA}}$ for retrieval with $\text{ELECTRA}_{\text{QAQ}}$ for reranking: the best configurations identified in the previous experiments.

We measure performance using Hit@k: the proportion of queries for which at least one equivalent question (and thus a correct answer) appears in the top-k results. This metric captures the system's ability to surface correct answers at different ranking depths. Figure 3.2 compares retrieval alone against the full pipeline with reranking.

The reranker improves Hit@1 from 42.6% to 46.6% (+4.0% absolute), demonstrating consistent benefit from two-stage ranking. However, a significant gap remains between system accuracy (46.6%) and the upper bound of answerable queries (75.4% at $k = 30$), indicating opportunities for future improvement.

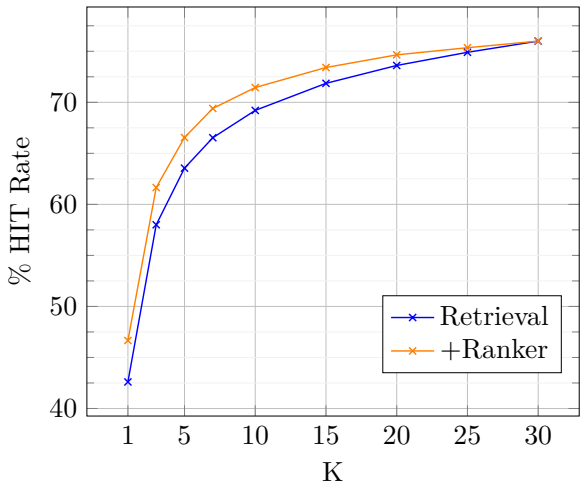

Figure 3.2: Hit rate at different cutoffs for retrieval alone (blue) and the full pipeline with reranking (orange).

Latency Analysis

We analyze QUADRo's efficiency on an NVIDIA A100 GPU with the 6.3M pair database. The system achieves sub-second latency across configurations. Total latency scales linearly with the number of reranked candidates. Retrieving and reranking 500 candidates takes approximately 530ms, while 50 candidates requires only 140ms. The majority of time is spent in cross-encoder reranking; retrieval via exhaustive cosine similarity search remains constant at approximately 15ms regardless of the number of returned candidates. These latency characteristics confirm that DBQA offers substantial efficiency advantages over generative approaches. While state-of-the-art LLMs typically require multiple seconds per query, QUADRo delivers answers in under 150ms when reranking 50 candidates which fast enough for interactive applications. Moreover, production deployments could achieve even lower latencies: retrieval can be accelerated using approximate nearest neighbor indices (e.g., FAISS (Douze et al., 2025) with IVF or HNSW, OpenSearch), while the cross-encoder reranker can be compiled and optimized using frameworks like TensorRT (Jeong et al., 2022). Figure 3.3 show the latency at various database size cutoff.

3.5.5 Comparison with Web-based QA and LLMs

The experiments so far have evaluated QUADRo on the Question Ranking Corpus, demonstrating the value of our dataset and architectural choices. However, a natural question arises: how does database-based QA compare against alternative paradigms in realistic open-domain settings? To answer this question, We compare QUADRo against two fundamentally different approaches to question answering:

- **Web-based QA**: Systems that retrieve documents from the web and extract answers from text. These systems have access to virtually unlimited knowledge but must identify relevant passages and extract precise answers.
- **Large Language Models**: Models that generate answers from parametric knowledge acquired during pre-training, without accessing external sources. These systems offer

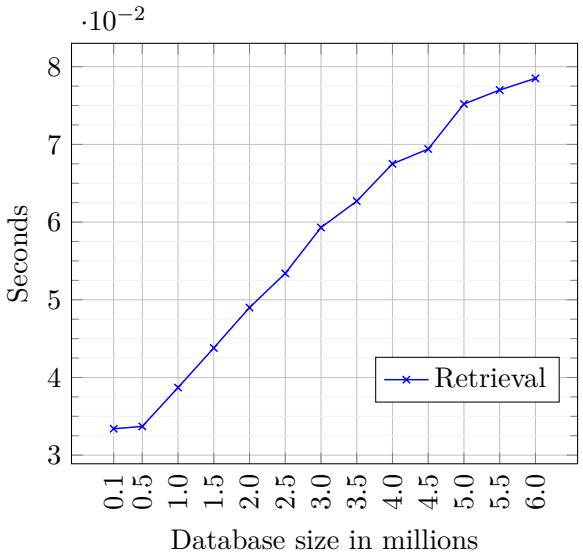

Figure 3.3: Latency of the end-to-end QUADRo system with different size cutoff of the DB. Values are averaged over 200 executions.

flexibility but may hallucinate or lack specific factual knowledge.

This comparison helps characterize the strengths and limitations of the DBQA paradigm: when does retrieving from a curated database outperform searching the web or querying an LLM?

Baseline Systems

WebQA. A web-based QA pipeline using Bing search engine for document retrieval and a state-of-the-art answer sentence selector (Lauriola and Moschitti, 2021a) for answer extraction. This represents a strong retrieve-and-rank baseline with access to web-scale knowledge.

LLMs. We evaluate Falcon-7B (ZXhang et al., 2023) and Vicuna-7B-v1.5 (Chiang et al., 2023) in zero-shot settings, where models generate answers from parametric knowledge without external retrieval. We use a structured prompt with two examples to elicit well-formed, concise answers.

```
Answer the question below. The answer must be [well-formed and concise].
The only accepted format is the following:
Question: [here the question]
Answer: [here the answer]
Here you have some examples:
Example 1:
Question: what is an apple?
Answer: An apple, (Malus domestica), is a domesticated tree and fruit
of the rose family (Rosaceae), one of the most widely cultivated tree
fruits. Apples are predominantly grown for sale as fresh fruit, though
apples are also used commercially for vinegar, juice, jelly, applesauce,
and apple butter and are canned as pie stock
Example 2:
```

```
Question: What is the largest airport in the world by travelers?
Answer: Atlanta Hartsfield-Jackson International Airport (ATL) is the
larger airport in the world with 75,704,760 total passengers. Dubai International
Airport (DXB) is the second busiest airport, followed by Tokyo International
Airport (HND) which is the third.
Ok, let's begin!
Question: {input-question}
```

Evaluation Setup

We sample questions from three open-domain sources not included in our database or training set. The samples aim to test the system in different scenarios:

- **Quora** (200 questions): Open-domain questions from QuoraQP that were held out from training. These questions tend to be underspecified and conversational.
- **Natural Questions** (200 questions): Questions sampled from Google search traffic (Kwiatkowski et al., 2019). Each question is designed to have a Wikipedia article containing the answer, making this favorable for web-based systems.
- **TriviaQA** (150 questions): Challenging trivia questions authored by enthusiasts (Joshi et al., 2017). These are complex, compositional questions that test deep knowledge.

Answers are manually evaluated for correctness by expert annotators.

Results

Table 3.9 presents end-to-end QA accuracy.

Table 3.9: End-to-end QA accuracy (%) on open-domain benchmarks.

System	Quora	NQ	TriviaQA
WebQA (Bing + Reranker)	35.0	**56.0**	27.0
Falcon-7B	**58.0**	40.0	**40.6**
Vicuna-7B v1.5	25.0	40.0	21.0
QUADRo (Full)	**58.0**	50.5	29.3
– w/o QRC training	53.0	47.0	28.0
– w/o neural retrieval	39.0	37.5	29.3
– w/o reranker	51.5	40.0	19.0
– w/o answer in input	50.0	45.0	25.3

QUADRo vs. WebQA. QUADRo substantially outperforms WebQA on Quora (58% vs. 35%) where questions are often underspecified and hard to answer via document retrieval. On Natural Questions, WebQA performs better (56% vs. 50.5%) since NQ questions are specifically designed to have Wikipedia answers, matching Bing's strengths. Performance is comparable on TriviaQA.

QUADRo vs. LLMs. QUADRo matches Falcon-7B on Quora (58%) and outperforms both LLMs on Natural Questions (50.5% vs. 40%). LLMs struggle with factual questions requiring specific knowledge not reliably encoded in their parameters. However, Falcon performs best on TriviaQA, possibly due to trivia content in its training data.

Ablation Analysis. The ablation rows demonstrate the contribution of each QUADRo component:

- Training on QRC provides +5% on Quora, +3.5% on NQ
- Neural retrieval (vs. BM25) provides +19% on Quora, +13% on NQ
- The reranker provides +6.5% on Quora, +10.5% on NQ
- Answer-aware inputs provide +8% on Quora, +5.5% on NQ

Each component contributes meaningfully, with neural retrieval being the most impactful.

Key Findings

QRC Training Improves Performance (RQ1). Training on the Question Ranking Corpus yields consistent improvements over existing resources: +5% accuracy on Quora and +3.5% on Natural Questions compared to models trained only on QuoraQP. Our reranker substantially outperforms prior state-of-the-art cross-encoders for question pairs (QP-RoBERTa), achieving +5.2 P@1 improvement on the QRC test set. The gains stem from two key properties of QRC: answer-aware annotations that reduce ambiguity, and hard negatives from neural retrieval that force models to learn fine-grained semantic distinctions. ask-specific training data with challenging negatives is essential for effective question ranking.

Answers Are Essential for Question Retrieval (RQ2). Incorporating answers provides substantial benefits at every stage of the pipeline. In retrieval, the QQA configuration (encoding question-answer pairs) outperforms the answer-agnostic QQ configuration by +5 P@1, indicating that answers produce more discriminative embeddings. During annotation, showing answers to crowdworkers reduced disagreement and error rates by 45%, as answers disambiguate underspecified questions. Even the ablation removing answers from end-to-end evaluation shows -8% accuracy on Quora. Answers serve as a semantic bridge that helps identify equivalent questions even when their surface forms differ significantly.

Answer-First Ordering Is Optimal for Reranking (RQ3). Among the four input configurations tested (QQ, QA, QQA, QAQ), the QAQ configuration (query-answer-question) achieves the best reranking performance, outperforming QQA by +1.5 P@1. The QA configuration, which ignores the database question entirely, performs worst (37.1% P@1 vs. 50.8% for QAQ), confirming that both question and answer information are necessary. Placing the answer between query and database question creates an optimal information flow: the model first assesses query-answer compatibility, then verifies question-question equivalence.

DBQA Is Competitive with Web-based QA and LLMs (RQ4). QUADRo matches or outperforms alternative paradigms across open-domain benchmarks. On Quora, QUADRo achieves 58% accuracy, matching Falcon-7B and substantially outperforming WebQA (+23%). On Natural Questions, QUADRo reaches 50.5%, outperforming both LLMs tested (+10.5% over Falcon-7B). Crucially, QUADRo delivers answers in 140ms when reranking 50 candidates, compared to multiple seconds for LLM generation. The ablation analysis confirms each component's contribution: neural retrieval provides +19% over BM25, the reranker adds +6.5-10.5%, and answer-aware inputs contribute +5.5-8%. DBQA offers a practical trade-off between accuracy and efficiency, with transparent provenance that generative approaches cannot provide.

3.6 Discussion

3.6.1 The Role of Answer Context

The consistent benefit of incorporating answers across all pipeline stages: retrieval, ranking, and annotation, suggests a deeper principle: in DBQA, answers are not merely the output but an integral part of the matching process. This has practical implications for system design. First, answer quality directly impacts retrieval accuracy; a database with incorrect or outdated answers will produce poor results regardless of model sophistication. Second, investing in answer verification and quality control can yield improvements beyond what model enhancements alone can achieve. Future DBQA systems should therefore prioritize answer curation as much as retrieval model development.

3.6.2 DBQA in the QA Landscape

Our comparison with WebQA and LLMs reveals complementary strengths rather than clear winners. DBQA offers several advantages: guaranteed factual answers when the database contains relevant pairs, fast inference without expensive generation, transparent provenance that lets users see which question matched, and easy knowledge updates without retraining.

However, DBQA also has inherent limitations: it cannot answer questions outside database coverage, is sensitive to database quality, and requires significant upfront investment in database construction.

These trade-offs suggest that DBQA is best viewed not as a replacement for other paradigms but as a complementary component. Hybrid systems could leverage DBQA for covered questions, exploiting its speed and accuracy, while falling back to generative approaches for novel questions. The reranking confidence score provides a natural mechanism for triggering such fallback.

3.6.3 Error Analysis

While QUADRo achieves competitive performance, understanding its failure modes is essential for guiding future improvements. We manually examined 100 incorrectly answered queries from the test set, categorizing errors by their underlying cause.

The most common error type (42%) is *semantic similarity without equivalence*: the system ranks highly questions that are topically related but not truly equivalent. For example, "*What is the capital of France?*" may retrieve "*What are major cities in France?*" which is semantically similar but requiring a different answer. This suggests that the model sometimes captures topical relatedness rather than strict equivalence.

A second pattern (28%) involves *lexical overlap bias*. Despite neural modeling, the system may rank "*What year was X born?*" above more semantically similar paraphrases that use different vocabulary (e.g., "*When did X come into the world?*"). This indicates residual reliance on surface-level matching that specialized pre-training (Chapter 4) may help address.

The remaining errors (30%) stem from *database coverage gaps*: the database simply lacks any equivalent question. In these cases, even perfect ranking would not help. Expanding coverage through automated question-answer pair generation, or falling back to generative approaches when confidence is low, could address this limitation.

3.6.4 Relationship to Document Retrieval

While this chapter focuses on question-to-question retrieval, the techniques developed here relate to broader information retrieval research. Three key differences distinguish question retrieval from document retrieval. First, question retrieval is inherently symmetric, if q_i is equivalent to q_j, then q_j is equivalent to q_i, whereas document retrieval is asymmetric since a query relates to a document but not vice versa. This symmetry enables different training strategies and evaluation metrics. Second, questions are typically short, 12 tokens on average in our database, while documents span paragraphs or pages, affecting both encoding strategies and the role of lexical overlap. Third, our QQA configuration uses answers as a bridge between query and database question, which has no direct analog in document retrieval, though passage-level retrieval shares some characteristics.

These relationships inform the document retrieval coherence work in Chapter 5, where we apply question clustering concepts to improve retrieval consistency.

3.6.5 Limitations

This work has several limitations. In terms of coverage, our database and evaluation focus exclusively on English questions; extending to multilingual settings requires additional resources and cross-lingual modeling techniques, which we address partially in Chapter 5. Additionally, the current system targets factoid questions with verifiable answers, while subjective, opinion-based, or multi-part questions require different handling.

Regarding the database itself, our collection represents knowledge at a specific point in time; production systems would require mechanisms for updating answers, removing outdated pairs, and handling temporal queries (e.g., "*Who is the current president?*"). Furthermore, system accuracy is bounded by answer quality in the database as incorrect or outdated answers cannot be corrected through improved retrieval alone.

Finally, we return only the answer from the top-ranked pair. Some questions may benefit from aggregating information across multiple similar questions or presenting alternative answers with confidence scores.

3.7 Conclusion

This chapter addressed the first core challenge of this thesis: retrieving semantically equivalent questions from large-scale databases. We presented QUADRo, a comprehensive framework for Database-based Question Answering, and made the following contributions:

1. A large-scale database of 6.3 million question-answer pairs from diverse high-quality sources, enabling question retrieval at scale

2. The Question Ranking Corpus (QRC), the first large-scale dataset specifically designed for question retrieval with answer-aware annotations and hard negatives (443,000 annotated examples)

3. Extensive experiments demonstrating that incorporating answers substantially improves both retrieval (+5 P@1) and ranking accuracy

4. Evidence that neural retrieval is essential for semantic matching at scale (+19% over BM25), and that our system achieves competitive performance with web-based QA and LLMs while offering sub-second latency

Our findings establish question retrieval as a viable paradigm for factual question answering, offering advantages in accuracy, efficiency, and transparency. The resources introduced: (i) the database, (ii) QRC dataset, and (iii) trained models are released to support future research.

Crucially, this chapter provides the foundation for subsequent contributions. The question retrieval models developed here, while effective, follow a standard paradigm: pre-training on general semantic similarity data followed by fine-tuning on task-specific annotations (QRC). This raises a natural question: can we design pre-training objectives that are specifically aligned with question ranking, reducing dependence on expensive annotations?

Chapter 4 addresses this question by introducing Question Ranking Pre-training (QRP), which leverages the question-answer database itself to create self-supervised training signal. We show that pre-training on ranking perturbation detection yields specialized representations that further improve retrieval accuracy while reducing annotation requirements.

Beyond pairwise equivalence, the concept of semantic equivalence naturally extends to *clusters* of questions sharing the same information need. Chapter 5 makes this extension explicit, demonstrating that question clusters can ground LLM responses and ensure coherent document retrieval. Finally, the question retrieval infrastructure developed here enables applications beyond direct question answering. Chapter 6 demonstrates that retrieving semantically equivalent questions can transform proprietary datasets into public alternatives, addressing both privacy concerns in data sharing and contamination issues in benchmark evaluation.

Chapter 4

Specialized Pre-Training for Question Ranking

The previous chapter established the foundations for question retrieval from large-scale databases, demonstrating that neural models trained on the Question Ranking Corpus (QRC) significantly outperform lexical baselines. However, a fundamental question remains: can we further improve these models by designing pre-training objectives that are specifically aligned with the question ranking task?

Standard pre-training objectives like Masked Language Modeling (MLM) are designed for general language understanding and do not explicitly capture the semantics of question equivalence or the structure of ranking tasks. While supervised fine-tuning on task-specific annotated data substantially improves performance, acquiring such data is expensive and time-consuming. For example, constructing the QRC dataset described in Chapter 3 required extensive annotation effort through Amazon Mechanical Turk, with costs estimated at $2-3 per query for expert-quality labels on 30 ranked candidates.

This motivates the development of effective pre-training techniques that can reduce the amount of task-specific supervision required to achieve strong performance. In this chapter, we address this gap by introducing a novel unsupervised pre-training method specialized for question retrieval and ranking. Our approach, which we call *Question Ranking Pre-training (QRP)*, leverages two key innovations:

1. **Knowledge Distillation from Retrieval**: We use the trained bi-encoder retrieval model from Chapter 3 to generate rankings of question-answer pairs for millions of queries, effectively distilling the retrieval model's learned similarity function into training data.

2. **Ranking Corruption Detection**: We create a self-supervised task by randomly swapping the top-ranked question-answer pair with another candidate, then training models to detect whether rankings have been corrupted. This forces models to learn internal representations of query semantics from the relationships between candidates.

A crucial insight underlying our method is that question retrieval implicitly creates *clusters* of semantically equivalent questions. When we query a database with a question, the retrieval model returns a ranked list of similar questions that effectively form a cluster around the same information need. This implicit clustering structure contains rich information about question semantics that can be distilled into a pre-training objective. Importantly, while this concept

of question clusters is implicit in this chapter, it will be made explicit in Chapter 5, where we demonstrate that clusters of equivalent questions can ground LLM responses, and ensure coherent document retrieval.

The pre-training task is designed with a crucial constraint: the original query is *not* provided to the model during pre-training. Instead, the model must learn to internally reconstruct the query's semantic properties by analyzing relationships between the ranked question-answer candidates. This constraint ensures that the model learns robust representations of question semantics that transfer effectively to downstream ranking tasks.

To illustrate, imagine a retrieval system returns the following candidates for an unknown query: (1) "*How many bones in the human body?*" [answer: 206], (2) "*What bones make up the skeleton?*" [answer: 206 bones…], (3) "*How many muscles in the body?*" [answer: over 600]. A human can tell that candidate (3) does not belong with the others, even without seeing the original query. QRP trains models to make exactly this judgment: detecting which candidate has been swapped into the top position, forcing the model to learn question equivalence patterns from candidate relationships alone.

This chapter makes the following contributions:

1. **Novel Pre-Training Objective**: We introduce Question Ranking Pre-training (QRP), the first unsupervised pre-training method specifically designed for question ranking tasks. Unlike existing pre-training objectives that target general language understanding or answer selection, QRP directly models the ranking task structure.

2. **Large-Scale Training Data Generation**: We develop a pipeline for generating 18 million pre-training examples using a basic DBQA system with 38 million question-answer pairs. The data generation process leverages knowledge distillation from a retrieval model while introducing controlled perturbations to create the self-supervised task.

3. **Comprehensive Evaluation**: We conduct extensive experiments on three benchmarks: QRC (question ranking), Quora-match (question similarity classification), and SemEval-2016 (community QA). Results demonstrate consistent improvements over both general pre-training methods (MLM, RTS, STS) and standard knowledge distillation approaches.

4. **State-of-the-Art Performance**: Our approach achieves state-of-the-art results on QRC (+1.05% P@1 over the public checkpoint baseline, statistically significant with p-value=0.0005) and competitive performance on Quora-match. Combining QRP with modern distillation techniques yields further gains (+1.19% P@1 on QRC).

5. **Analysis and Insights**: We provide detailed analysis showing that: (i) including the query during pre-training degrades performance by making the task too easy, (ii) QRP substantially reduces model variance across runs, improving stability, and (iii) the benefits transfer across different question ranking scenarios.

The remainder of this chapter is organized as follows. Section 4.1 describes the Question Ranking Pre-training method, including the DBQA system for data generation, the data creation pipeline, and the pre-training objective. Section 4.2 presents our experimental setup. Section 4.3 reports comprehensive results across multiple benchmarks. Section 4.4 discusses implications, limitations, and connections to subsequent chapters. Finally, Section 4.5 concludes the chapter.

4.1 Question Ranking Pre-Training Method

Existing pre-training objectives, including token-level methods such as MLM and ELECTRA and sentence-level methods such as NSP and contrastive learning, are not specifically designed for question ranking (see Section 2.3.3 for a comprehensive review of pre-training methods). In this scenario, our QRP approach fills this gap by combining: (i) explicit modeling of ranking structure over multiple candidates, (ii) knowledge distillation from retrieval models, (iii) self-supervised corruption detection without access to the original query, and (iv) specific targeting of question-question similarity.

The method consists of three main stages:

1. **Expanding the DBQA System**: We scale the question-answer database from Chapter 3 from 6.3M to 38M pairs, reusing the trained bi-encoder retrieval model from that chapter.

2. **Generating Ranking Data**: We use the system to retrieve ranked question-answer pairs for millions of queries, then apply controlled perturbations to create pre-training examples.

3. **Pre-Training with Ranking Detection**: We train Transformer models to detect whether rankings have been corrupted, forcing them to learn question semantics from candidate relationships without access to the original query.

The following subsections detail each stage.

4.1.1 Question-Answer Database System

Our pre-training data generation relies on a DBQA built upon the same architecture as the one introduced in Chapter 3. The system consists of two components: a large-scale database and a trained retrieval model.

Database

Generating millions of diverse pre-training examples requires a database substantially larger than the 6.3 million pairs used in Chapter 3. A larger and more varied database ensures that pre-training data covers a broad range of question types, topics, and linguistic patterns, thus reducing the risk of overfitting to specific domains. We therefore expand the database to approximately *38 million pairs* by combining two sources:

Original QUADRo Database (6M pairs): We include all 6.3 million question-answer pairs from the QUADRo database described in Chapter 3, Section 3.2. This provides high-quality pairs from diverse sources including GooAQ, WikiAnswers, WQA, CovidQA, HotpotQA, and filtered Quora/ELI5 pairs.

PAQ Database (32M pairs): We augment with pairs from the Probably Asked Questions (PAQ) dataset (Lewis et al., 2021), which originally contains 65 million automatically generated questions from Wikipedia passages. Since PAQ was constructed using question generation models, it contains noise from incorrect or low-quality question-answer associations. To ensure quality, we filter PAQ pairs using the Answer Sentence Selection (AS2) model from Lauriola and Moschitti (2021a), and retain only the top 50% with

the highest confidence scores. This filtering yields approximately 32 million high-quality pairs that complement the manually curated QUADRo data.

This combination balances quality (from QUADRo) with scale (from PAQ), ensuring that pre-training data reflects both well-formed question patterns and the noisy, varied questions that models encounter in real-world applications.

Retrieval Model

We reuse the bi-encoder retrieval model trained in Chapter 3. This model is based on MiniLM-L12-v2 (Wang et al., 2020a), a distilled version of BERT with 12 layers, 384 embedding dimensions, and approximately 33 million parameters, providing a good balance between accuracy and efficiency for large-scale retrieval. As described in Section 3.3.1, the model was fine-tuned on QRC using the QQA configuration: the query branch encodes questions only, while the database branch encodes question-answer pairs concatenated with `[SEP]` tokens.

The retrieval model achieves strong performance on question similarity while remaining computationally efficient (33M parameters vs. 110M+ for typical reranking models). Importantly, this model is *not* intended to be state-of-the-art: it serves as a knowledge source for generating pre-training data. The model's limitations and imperfections actually benefit pre-training by introducing realistic noise and uncertainty that the student model must learn to handle.

We pre-compute embeddings for all 38 million question-answer pairs offline, storing them in a FAISS Flat index (Johnson et al., 2019b) with a full search. Given a query, retrieval takes less than 100ms on CPU, enabling generation of millions of pre-training examples in reasonable time.

4.1.2 Pre-Training Data Generation

The core idea is to leverage the DBQA retrieval model as an imperfect "teacher" for generating pre-training data without manual annotation. For any query, the retrieval model returns a ranked list of similar questions that form an implicit cluster around the same information need. By introducing controlled perturbations such as swapping the top-ranked item with a lower-ranked one, we create examples where the model must distinguish correct from corrupted orderings. More precisely, we omit the original query, requiring the model to infer semantic relationships solely from the candidates and their arrangement. With this in mind, we generated 18 million examples through four stages: query collection, candidate retrieval, perturbation, and formatting. Figure 4.1 illustrates the complete data generation pipeline, which proceeds through four stages: query collection, candidate retrieval, perturbation, and formatting.

Query Collection

We collect 18 million query questions from three diverse sources:

WQA: Approximately 15K questions from the WebQuestions dataset (Zhang et al., 2022), consisting of real user queries to a web search engine. These are typically short factoid queries (e.g., *"What is the capital of France?"*).

GooAQ: Approximately 3 million questions derived from Google auto-complete suggestions and People Also Ask boxes (Khashabi et al., 2021). These represent common information-seeking queries with natural phrasing patterns.

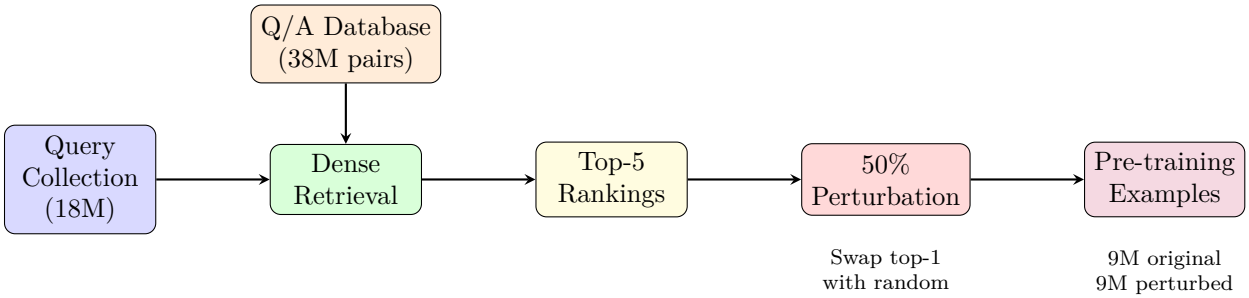

Figure 4.1: QRP data generation pipeline. Queries are processed through dense retrieval to obtain top-5 rankings, then 50% are perturbed by swapping the top-ranked item with a random position. The resulting 18M examples (balanced between original and perturbed) form the pre-training dataset.

PAQ: The remaining approximately 15 million questions are randomly sampled from the PAQ dataset (Lewis et al., 2021). To avoid overlap with the database, we sample only from questions that are not present in the 38M QA pairs used for retrieval. While these automatically generated questions are less natural than real user queries, they provide broad topical coverage across encyclopedic knowledge.

The diversity of sources ensures that pre-training data covers a wide range of question types, topics, and linguistic patterns, preventing overfitting to specific question distributions.

Retrieval and Ranking

Given the collection of queries and the database with pre-computed embeddings, we generate ranked candidate lists that will serve as the basis for pre-training examples.

For each of the 18 million queries, we:

1. Compute the query embedding using the fine-tuned MiniLM retrieval model
2. Compute cosine similarity between the query embedding and all 38M pre-computed database embeddings
3. Sort the entire database by similarity and retrieve the top $k = 5$ question-answer pairs
4. Remove the query from the retrieved rank if it appears among the candidates

The last step is important: since the queries originate from datasets which are also part of the database, the query itself might appear in the retrieved results. Removing it ensures that the model cannot trivially identify the ”correct” ranking by finding an exact match.

We use exhaustive search rather than approximate nearest neighbor methods to ensure maximum retrieval quality, as the ranked candidates directly determine the quality of pre-training signal. We set $k = 5$ as a trade-off: more candidates would provide richer context, but encoding longer sequences is expensive during pre-training and risks exceeding the 512-token limit of standard Transformer models. Each retrieved ranking consists of five triplets:

$$R = \{(q_1, a_1, s_1), (q_2, a_2, s_2), \ldots, (q_5, a_5, s_5)\}$$

where q_i is the i-th question, a_i is its answer, and s_i is the cosine similarity score with the query.

Importantly, the retrieved questions form an *implicit cluster* of semantically equivalent questions around the query. While the questions may have different surface forms, they share the same underlying information need. Table 4.1 illustrates this clustering phenomenon with representative examples. This cluster structure plays a central role in our pre-training objective and anticipates the explicit question clusters introduced in Chapter 5. The underlying idea is that, by learning to identify a cluster's internal structure without access to its center (the query), the model develops strong representations of semantic equivalence.

Table 4.1: Examples of retrieved question rankings for different queries. The retrieved questions form implicit clusters around the same information need. The model sees only the ranked questions (not the query) and must learn to detect perturbations.

Rank	Question
Query: How many calories in a pineapple?	
1	How many calories are in an pineapple?
2	How many calories in a whole pineapple?
3	How many calories does a pineapple have?
4	How many calories are in a serving of pineapple?
5	How many calories are in a piece of a pineapple?
Query: How old is the sun?	
1	How old is the Sun?
2	How old is sun?
3	How old can the Sun be?
4	What is the approximate age of the sun?
5	How long has the sun existed?
Query: What is a cucumber?	
1	What are cucumbers?
2	What is cucumber mean?
3	Tell me what is cucumbers?
4	What does cucumber mean?
5	What is the definition of cucumber?

Pre-training Example Construction

The key innovation of QRP is the ranking perturbation strategy. We randomly select 50% of the retrieved rankings and apply the following perturbation:

1. Randomly select a position $i \in \{2, 3, 4, 5\}$ (not position 1)
2. Swap the question-answer pair at position 1 (highest ranked) with the pair at position i
3. Keep all other pairs in their original positions

This creates two types of pre-training examples:

1. **Original Rankings:** Rankings where the retrieval model's ordering is preserved. These examples are labeled as `correct=1`.
2. **Perturbed Rankings:** Rankings where the top position has been corrupted by swapping with a lower-ranked candidate. These examples are labeled as `correct=0`.

The perturbation is designed to be both challenging and realistic. By swapping the top-ranked pair with another retrieved candidate rather than a random question from the database, we ensure that both pairs are semantically related to the query, yet are unlikely to be equally good matches due to their different positions in the original ranking. The model must therefore learn to detect subtle semantic distinctions in order to identify the corruption.

Each pre-training example is encoded as a concatenated sequence of question-answer pairs:

$$\texttt{[CLS]}\ q_1\ \texttt{/}\ a_1\ \texttt{[SEP]}\ q_2\ \texttt{/}\ a_2\ \texttt{[SEP]}\ \ldots\ \texttt{[SEP]}\ q_5\ \texttt{/}\ a_5\ \texttt{[EOS]} \tag{4.1}$$

Crucially, the original query is *not included* in the sequence. This is a fundamental design decision: by omitting the query, we force the model to internally reconstruct what query would have produced this ranking. As consequence, the model must learn to identify semantic patterns shared by highly-ranked candidates and detect when a candidate is inconsistent with the others.

The resulting Question Ranking Pre-training (QRP) dataset contains 18 million examples, evenly split between original and perturbed rankings. The dataset covers diverse question types across factoid queries, definitional questions, how-to questions, and more complex information needs.

4.1.3 Pre-Training Task and Objective

Task Definition and Training Objective

The pre-training task is formulated as binary classification: given a ranking of five question-answer pairs, without the original query, predict whether the ranking is correct (label 1) or has been corrupted by swapping (label 0).

Formally, let $R = [(q_1, a_1), \ldots, (q_5, a_5)]$ denote a ranking sequence and $y \in \{0, 1\}$ indicate whether the ranking has been perturbed ($y = 0$) or remains original ($y = 1$). The model outputs two logits from the `[CLS]` token representation:

$$[z_0, z_1] = f_\theta(R) \tag{4.2}$$

where f_θ is the Transformer encoder followed by a linear classification head. The predicted class is selected as $\hat{y} = \text{argmax}_i\, z_i$.

We train using binary cross-entropy loss:

$$\mathcal{L}_{\text{QRP}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(z_{y_i})}{\exp(z_0) + \exp(z_1)} \tag{4.3}$$

where N is the batch size and z_{y_i} is the logit corresponding to the true label $y_i \in \{0, 1\}$.

The model architecture is a standard Transformer encoder. Based on preliminary experiments comparing multiple architectures (Table 4.2), we select DeBERTa-v3-base (He et al., 2023) (184M parameters) as it achieves the best baseline performance on QRC.

Design Rationale

The key insight underlying our task design is that to detect whether a ranking has been corrupted, the model must *implicitly reconstruct the query* that generated the ranking. This requires the model to:

Table 4.2: Architecture selection based on QRC baseline performance (no additional pre-training).

Architecture	Parameters	P@1
BERT-base	110M	47.81
MiniLM-L12	33M	49.25
ELECTRA-base	110M	49.65
DeBERTa-v3-base	184M	50.82

1. **Identify Semantic Patterns**: Recognize properties shared by highly-ranked candidates that suggest what the original query was asking.

2. **Detect Inconsistencies**: Identify when a candidate doesn't match the inferred query semantics.

3. **Compare Relative Quality**: Assess whether the top position contains the most query-relevant pair compared to alternatives.

Consider the *"How old is the sun?"* example in Table 4.1. Questions 1–3 are nearly identical paraphrases, while questions 4–5 use different phrasings such as *"approximate age"* and *"how long has existed"*. To recognize that question 1 should be ranked highest, the model must understand that direct paraphrases are more similar to the latent query than semantic reformulations. This requires learning the fine-grained structure of question equivalence.

These capabilities directly transfer to the downstream ranking task, where the model must identify which candidates are most semantically similar to a given query.

Why not include the query? We experimented with including the query during pre-training (variant called QQRP - Query Question Ranking Pre-training), encoding examples as:

$$\texttt{[CLS]}\ q_{\text{query}}\ \texttt{[SEP]}\ q_1\ \texttt{/}\ a_1\ \texttt{[SEP]}\ \ldots\ \texttt{[EOS]} \tag{4.4}$$

However, this variant performed *worse* than QRP (see Section 4.3). Analysis of training dynamics revealed that QQRP achieves near-perfect training accuracy within the first few thousand steps, while QRP requires the full training period. This suggests that including the query makes the task trivially easy. In this scenario, the model can solve it through simple query-candidate matching without learning deep semantic patterns.

During fine-tuning, QRP-pretrained models converge faster and achieve better final performance, indicating that the harder pre-training task forces learning of more transferable representations.

Connection to Knowledge Distillation. The QRP data generation process inherently performs knowledge distillation from the retrieval model. The rankings encode the retrieval model's learned similarity function, as the candidates ranked higher are more similar to the query according to the retrieval model. By training on these rankings, the student model, which is larger and more powerful than the retrieval model, learns to internalize this knowledge while gaining additional capacity through the corruption detection task.

This can be viewed as a form of *structured* knowledge distillation:

1. **Traditional distillation**: Student matches teacher's probability outputs on individual examples

2. **QRP**: Student learns to recognize patterns in teacher's ranking behavior across multiple candidates

The key difference is that QRP distills relational knowledge which is the ranking structure, rather than pointwise knowledge which are the similarity scores, capturing richer semantic information about question equivalence.

4.1.4 Training Procedure

The training process consists of two stages: unsupervised pre-training on the QRP dataset, followed by supervised fine-tuning on the target task:

Pre-Training : We pre-train DeBERTa-v3-base on the QRP dataset using:

- Learning rate: 5×10^{-6}
- Batch size: 1024 (accumulated over multiple gradient steps)
- Optimizer: AdamW with linear warmup (10% of steps) and decay
- Loss: Cross-entropy
- Epochs: 2 epochs over 18M examples
- Hardware: Training on 8× NVIDIA V100 GPUs takes approximately 36 hours

Fine-Tuning : After pre-training, models are fine-tuned on each target dataset separately. Based on findings from Chapter 3, we encode triplets using the QAQ configuration:

$$\texttt{[CLS]}\ q_{\text{query}}\ \texttt{[SEP]}\ a_i\ \texttt{[SEP]}\ q_i\ \texttt{[EOS]} \tag{4.5}$$

This places the answer between query and database question, which was shown to be most effective for question ranking with cross-encoders.

Fine-tuning hyperparameters are selected via grid search:

- Learning rate: $\{1, 2\} \times 10^{-\{5,6\}}$
- Batch size: $\{32, 64, 128, 256\}$
- Early stopping based on validation performance

Model selection uses development set performance: P@1 for QRC, ROC-AUC for Quora-match.

4.2 Experimental Setup

We conduct extensive experiments to validate the Question Ranking Pre-training (QRP) method. Our evaluation addresses several key research questions summarized in Table 4.3.

Table 4.3: Research questions for specialized pre-training.

RQ	Research Question	Section
RQ1	Does QRP improve question ranking performance compared to general-purpose pre-training objectives (MLM, RTS, STS)?	§4.3.1
RQ2	Why does excluding the query during pre-training lead to better downstream performance?	§4.3.1
RQ3	How does QRP compare to knowledge distillation approaches, and are they complementary?	§4.3.1
RQ4	Do the benefits of QRP transfer to different datasets and task formulations (ranking vs. classification)?	§4.3.2, §4.3.3
RQ5	Does QRP reduce variance across training runs, leading to more stable models?	§4.3.4

4.2.1 Datasets

In order to answer our research questions, we evaluate pre-trained models on three benchmarks representing different question ranking scenarios:

1. **Question Ranking Corpus (QRC):** the dataset introduced in Chapter 3, consisting of 15,211 queries with 30 annotated question-answer candidates each, split into training (11711 queries), development (1500), and test (2000) sets. Performance is measured using standard ranking metrics as defined in Chapter 2: P@1, MAP, and MRR. QRC is the primary evaluation benchmark as it most closely matches the pre-training task: open-domain question ranking in DBQA contexts.

2. **Quora-match:** a large dataset of approximately 200,000 question-question-answer triplets for binary classification of question equivalence (Wang et al., 2020b). Each example contains two questions and an answer, and the task is to determine whether the questions are semantically equivalent. The dataset is imbalanced (35% positive, 65% negative), so we focus on: ROC-AUC (primary metric), Accuracy, and F1 Score. Quora-match tests whether pre-training benefits transfer to pairwise classification settings, which are common in duplicate question detection applications. Details on how the dataset has been built are available in Section 2.8.

3. **SemEval-2016 Task 3:** the SemEval-2016 Community Question Answering shared task (Nakov et al., 2016b) evaluates question-question similarity in forum contexts. Each query is associated with 10 related questions from Qatar Living forums. The dataset contains 387 queries total.

   We use this as a *transfer learning* benchmark: models are trained on QRC and directly evaluated on SemEval without additional fine-tuning. This tests whether the learned representations generalize to community QA settings, which differ from open-domain QA in several ways:

   - Conversational style and informal language (e.g., *"Hi Guys; I need to open a new bank account. Which is the best bank in Qatar?"*)
   - Domain-specific context (Doha, Qatar)
   - Smaller ranking depth (10 candidates vs. 30 in QRC)

4.2.2 Baselines

We compare our QRP method against several baselines to isolate the contribution of different components. The baselines are organized into three groups: (i) standard pre-training objectives applied to the same data, to test whether the QRP task itself is essential; (ii) a variant that includes the query, to validate our query-exclusion hypothesis; and (iii) knowledge distillation approaches, to compare implicit distillation through ranking data with explicit distillation during fine-tuning. In light of this the used baselines are:

Public Checkpoint. DeBERTa-v3-base without additional pre-training, fine-tuned directly on target datasets. This represents the standard transfer learning baseline.

Standard Pre-Training Objectives. To ensure fair comparison, we implement several general pre-training objective, details about these approaches can be found in Section 2.3.2, and train them on the *same 18M QRP data*:

- **MLM**: Masked Language Modeling (Devlin et al., 2019b) . In short, the method masks 15% of tokens and predict them
- **RTS**: Random Token Swap (Di Liello et al., 2022b). The method swap tokens and detect which were swapped.
- **STS**: Semantic Textual Similarity (Reimers and Gurevych, 2019b). This approach predicts similarity between questions

Using the same data for all baselines ensures that performance differences reflect the pre-training objective rather than data scale or domain coverage.

ALL. A structural self-supervised pre-training method from Di Liello et al. (2022a) that aims to simulate the AS2 task. Unlike the other baselines, this model was provided by the original authors and was pre-trained on 600M examples (42× our data), so it is not directly comparable but included for reference. Details on this technique are available in Section 2.3.3.

QQRP (Query-Question Ranking Pre-training). A variant of our method that includes the original query alongside the top-5 q/a pairs. This tests our hypothesis that query exclusion is essential.

Knowledge Distillation. Since QRP implicitly distills knowledge from the retrieval model, we compare against explicit distillation approaches applied during fine-tuning:

1. **Hinton et al.** (Hinton et al., 2015): Standard distillation where the student model is trained to match both ground-truth labels and teacher predictions:

$$\mathcal{L}_{\text{distill}} = (1-\lambda)\mathcal{L}_{\text{CE}}(y, s_s) + \lambda\mathcal{L}_{\text{MSE}}(s_s, s_t) \tag{4.6}$$

   where the loss (Equation 4.6) is defined as linear combination of (i) the CrossEntropy loss between the student model prediction (s_s) and label (y), and (ii) MSE between the teacher (s_t) and the student (s_s) probability scores $[0, 1]$, and $\lambda \in \{0, 0.1, \ldots, 1\}$ is selected via validation.

2. **Gabburo et al.** (Gabburo et al., 2023): Uncertainty-weighted distillation that increases loss weight for examples where the teacher is uncertain:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{CE}}(y, s_s) \times (1 - s_t) \tag{4.7}$$

   This helps the student model learn to correct the teacher's mistakes on difficult examples.

Table 4.4: Results on QRC test set. Best results in bold. All experiments averaged over 5 runs with standard deviation. All models use DeBERTa-v3-base architecture.

Setting	P@1	MAP	MRR
Public checkpoint	$50.82_{\pm 0.38}$	$48.44_{\pm 0.07}$	$60.23_{\pm 0.23}$
Pre-Training Techniques			
QRP (ours)	$\mathbf{51.87}_{\pm 0.17}$	$\mathbf{48.87}_{\pm 0.06}$	$\mathbf{60.98}_{\pm 0.10}$
QQRP (with query)	$51.04_{\pm 0.44}$	$48.87_{\pm 0.18}$	$60.63_{\pm 0.20}$
MLM	$50.23_{\pm 0.42}$	$48.25_{\pm 0.18}$	$59.90_{\pm 0.23}$
RTS	$50.95_{\pm 0.42}$	$48.63_{\pm 0.08}$	$60.38_{\pm 0.24}$
STS	$50.97_{\pm 0.49}$	$48.60_{\pm 0.25}$	$60.36_{\pm 0.41}$
ALL	$50.85_{\pm 0.45}$	$48.68_{\pm 0.23}$	$60.23_{\pm 0.33}$
Distillation Approaches			
Hinton et al. (2015)	$51.57_{\pm 0.51}$	$48.95_{\pm 0.15}$	$60.86_{\pm 0.24}$
+ QRP	$51.28_{\pm 0.44}$	$48.97_{\pm 0.13}$	$60.63_{\pm 0.30}$
Gabburo et al. (2023)	$50.96_{\pm 0.41}$	$48.84_{\pm 0.24}$	$60.48_{\pm 0.32}$
+ QRP	$\mathbf{52.01}_{\pm 0.34}$	$\mathbf{49.14}_{\pm 0.11}$	$\mathbf{61.02}_{\pm 0.30}$

We also test combinations of distillation with our QRP pre-training (denoted +QRP) to determine whether they provide complementary benefits.

For distillation experiments, the teacher model is the MiniLM-L12-v2 retrieval model (33M parameters) fine-tuned on QRC, and the student is DeBERTa-v3-base (184M parameters). Interestingly, the student is 3× larger than the teacher, which is unusual but reflects practical constraints: *retrieval models must be smaller for efficiency while reranking models can be larger for accuracy.* All experiments are repeated 5 times with different random seeds to measure variance. We report mean ± standard deviation across runs and perform statistical significance testing (paired t-test) to compare methods. A result is considered statistically significant if p-value < 0.05.

4.3 Results and Analysis

We evaluate QRP on three benchmarks: QRC for question ranking, Quora-match for binary classification, and SemEval-2016 for transfer learning. Each benchmark tests different aspects of the learned representations.

4.3.1 Results on Question Ranking Corpus (QRC)

QRC is our primary benchmark, as it directly evaluates the question ranking task for which QRP was designed. Table 4.4 reports performance on the QRC test set, comparing our QRP approach with baselines.

Key Findings

QRP Outperforms General Pre-Training (RQ1). Our QRP method achieves 51.87% P@1, improving over the public checkpoint baseline by +1.05 percentage points (p-value=0.0005,

statistically significant). In contrast, general pre-training objectives (MLM, RTS, STS, ALL) provide minimal or no improvement. Additionally, MLM slightly hurts performance (−0.59 P@1). This demonstrates that task-specific pre-training is crucial: *simply pre-training on the same data with generic objectives does not transfer effectively to question ranking.*

Query Exclusion is Essential (RQ2). QQRP, which includes the query during pre-training, underperforms QRP by −0.83 P@1 despite using identical data and task structure. This validates our hypothesis: *when the query is available, the task becomes too easy and the model does not learn the deep semantic properties needed for ranking.*

Reduced Variance (RQ5). An important but often overlooked benefit: QRP reduces standard deviation by over 50% compared to the baseline (±0.17 vs. ±0.38 for P@1). This indicates that QRP not only improves average performance but also produces more stable models, which is valuable for reproducibility and production deployment.

Distillation Provides Complementary Benefits (RQ3). Standard distillation (Hinton et al., 2015) improves P@1 by +0.75 points (p-value= 0.0299), confirming that knowledge from the retrieval model transfers to the reranking task. However, this improvement is smaller than QRP, suggesting that the ranking corruption detection objective captures additional useful structure beyond simple distillation.

The best overall performance (52.01% P@1) comes from combining QRP pre-training with the Gabburo et al. (2023) uncertainty-weighted distillation during fine-tuning. This provides +1.19 points over baseline (p-value=0.0023) and +0.14 points over QRP alone, indicating complementary benefits. Interestingly, this combination works better than Hinton et al. (2015) + QRP, suggesting that uncertainty-weighted distillation is more complementary to our pre-training objective.

4.3.2 Results on Quora-match

Table 4.5 reports performance on the Quora-match binary classification task.

Table 4.5: Results on Quora-match test set. Best results in bold.

Setting	ROC-AUC	Accuracy	F1
Public checkpoint	$96.92_{\pm 0.05}$	$91.56_{\pm 0.28}$	$87.81_{\pm 0.28}$
Pre-Training Techniques			
QRP (ours)	$\mathbf{97.05}_{\pm 0.03}$	$91.37_{\pm 0.11}$	$87.86_{\pm 0.25}$
QQRP (with query)	$96.63_{\pm 0.07}$	$91.55_{\pm 0.16}$	$87.76_{\pm 0.27}$
MLM	$96.78_{\pm 0.06}$	$91.06_{\pm 0.14}$	$87.05_{\pm 0.20}$
RTS	$96.81_{\pm 0.04}$	$91.22_{\pm 0.14}$	$87.42_{\pm 0.16}$
STS	$94.42_{\pm 0.22}$	$87.61_{\pm 0.38}$	$82.43_{\pm 0.32}$
ALL	$97.00_{\pm 0.09}$	$91.35_{\pm 0.60}$	$87.20_{\pm 0.12}$
Distillation Approaches			
Hinton et al. (2015)	$92.14_{\pm 0.65}$	$90.74_{\pm 0.69}$	$86.59_{\pm 1.15}$
+ QRP	$92.94_{\pm 0.65}$	$90.52_{\pm 0.43}$	$86.59_{\pm 0.61}$
Gabburo et al. (2023)	$97.01_{\pm 0.07}$	$91.67_{\pm 0.12}$	$87.95_{\pm 0.05}$
+ QRP	$\mathbf{97.20}_{\pm 0.20}$	$\mathbf{91.77}_{\pm 0.12}$	$\mathbf{88.05}_{\pm 0.05}$

Key Findings

Consistent but Smaller Gains (RQ4). QRP improves ROC-AUC by +0.13 points (p-value=0.0412, marginally significant) and shows similar trends to QRC results. The smaller improvements are expected for two reasons: (i) our pre-training is specifically designed for ranking, while Quora-match is a binary classification task, and (ii) baseline performance is already near ceiling (≈97% ROC-AUC), making further improvements inherently limited.

STS Performs Poorly. Interestingly, the STS pre-training objective substantially hurts performance on Quora-match (94.42% vs. 96.92% baseline). We hypothesize this is because STS focuses on continuous similarity prediction, while Quora-match requires binary classification. The mismatch between pre-training and downstream objectives degrades transfer.

Best Combination (RQ3). Again, combining QRP with (Gabburo et al., 2023) distillation yields the best results: 97.20% ROC-AUC (p-value=0.0161), 91.77% accuracy, and 88.05% F1. This pattern mirrors QRC results, confirming that the approaches provide complementary benefits across different task formats. The consistency across ranking and classification tasks suggests that QRP learns general question semantics rather than task-specific patterns.

4.3.3 Transfer Learning: SemEval-2016

Table 4.6 reports transfer learning results: models are trained on QRC and directly evaluated on SemEval-2016 without additional fine-tuning.

Table 4.6: Transfer learning results on SemEval-2016. Models are trained only on QRC and tested on SemEval without fine-tuning. Best results in bold.

Setting	P@1	MAP	MRR
Public checkpoint	$61.85_{\pm 1.08}$	$62.30_{\pm 0.81}$	$69.89_{\pm 0.89}$
Pre-Training Techniques			
QRP (ours)	$64.84_{\pm 1.03}$	$64.77_{\pm 1.29}$	$72.73_{\pm 1.05}$
QQRP (with query)	$64.34_{\pm 1.02}$	$64.73_{\pm 0.57}$	$71.74_{\pm 0.52}$
MLM	$63.12_{\pm 0.21}$	$61.63_{\pm 0.82}$	$69.00_{\pm 0.81}$
RTS	$63.12_{\pm 1.80}$	$62.57_{\pm 1.10}$	$70.77_{\pm 1.07}$
STS	$64.29_{\pm 1.25}$	$65.02_{\pm 0.45}$	$71.99_{\pm 0.22}$
ALL	$\mathbf{65.91}_{\pm 1.25}$	$\mathbf{66.61}_{\pm 0.34}$	$\mathbf{73.33}_{\pm 0.26}$
Distillation Approaches			
Hinton et al. (2015)	$64.04_{\pm 1.50}$	$64.74_{\pm 0.51}$	$71.48_{\pm 0.49}$
+ QRP	$64.11_{\pm 1.41}$	$65.76_{\pm 0.92}$	$72.05_{\pm 0.82}$
Gabburo et al. (2023)	$63.21_{\pm 1.48}$	$63.67_{\pm 0.56}$	$70.68_{\pm 0.79}$
+ QRP	$65.68_{\pm 0.85}$	$65.83_{\pm 0.41}$	$72.77_{\pm 0.50}$

Key Findings

Strong Transfer Performance (RQ4). Our QRP pre-training achieves +2.99 P@1 over the baseline (statistically significant, p-value=0.002), with corresponding gains in MAP (+2.47)

and MRR (+2.84). This is particularly impressive given that SemEval questions differ substantially from QRC: conversational style (e.g., "*Hi Guys; I need to open a new bank account. Which is the best bank in Qatar ?*"), domain-specific context (Doha, Qatar), and smaller candidate pools (10 vs. 30). The combination with Gabburo et al. (2023) distillation further improves to +3.83 P@1 (p-value=0.00063). These gains without any SemEval-specific fine-tuning demonstrate that QRP learns transferable question semantics rather than dataset-specific patterns.

ALL Pre-Training Excels on SemEval. The ALL method achieves the best absolute results on SemEval (65.91% P@1). However, this model was pre-trained on 600 million examples, which is 42× more data than our 18M examples, and was specifically designed for general transfer across diverse tasks. Given this massive data scale difference, the competitive performance of QRP is noteworthy and demonstrates the efficiency of task-specific pre-training.

Domain Generalization. The strong transfer performance demonstrates that QRP captures general question semantics that transfer across domains. Models trained only on open-domain factoid questions (QRC) successfully generalize to conversational community QA (SemEval), despite differences in style, vocabulary, and domain. Notably, QRP outperforms MLM and RTS on transfer (64.84% vs. 63.12%), suggesting that the ranking corruption detection task learns more transferable representations than token-level objectives. This generalization ability is valuable for deploying DBQA systems in new domains without requiring domain-specific annotations.

4.3.4 Variance Reduction

An underappreciated benefit of QRP is variance reduction across random seeds. Table 4.7 compares standard deviations:

Table 4.7: Standard deviation comparison across 5 random seeds on QRC. QRP substantially reduces model variance.

Method	P@1 StdDev	MAP StdDev	Improvement
Public Checkpoint	0.38	0.07	—
QRP (ours)	0.17	0.06	-55% (P@1)
Gabburo et al. (2023) + QRP	0.34	0.11	-11% (P@1)

QRP reduces P@1 variance by 55%, from ±0.38 to ±0.17. The reduction is more pronounced for P@1 than for MAP (55% vs. 14%), suggesting that QRP particularly stabilizes the model's ability to identify the single best candidate. When combining QRP with distillation, variance increases slightly (±0.34), likely because distillation introduces additional variability through the teacher's soft labels, partially offsetting QRP's stabilizing effect. This stability is valuable for both research reproducibility and production deployment.

4.4 Discussion

4.4.1 Why Does Query Exclusion Work?

The counterintuitive finding that excluding queries during pre-training improves downstream performance deserves deeper analysis. We propose several complementary explanations.

First, query exclusion forces abstraction. Without access to the query, the model cannot rely on shallow pattern matching between query and candidates. Instead, it must learn to identify abstract semantic properties shared by highly-ranked candidates: properties like "*calorie questions,*" "*age/duration questions,*" or "*definitional questions.*" These abstractions transfer more effectively to new queries than query-specific matching patterns.

Second, the query-free formulation emphasizes relational learning. The model learns relationships *between* candidates rather than relationships *to* a query. This is particularly valuable for reranking, where the model must compare candidates to determine relative quality. Training on candidate-candidate comparisons (implicitly, through corruption detection) may be more aligned with the reranking objective than training on query-candidate comparisons.

Third, increased difficulty acts as regularization. Making the pre-training task harder prevents overfitting to surface-level patterns. The model must develop robust internal representations to solve the more difficult query-free task, and these representations generalize better to downstream applications. This is analogous to findings in curriculum learning and hard negative mining, where appropriately difficult training improves generalization.

Finally, query exclusion prevents shortcut learning. When queries are present, the model may learn shortcuts: for example, simply checking lexical overlap between query and candidate questions. These shortcuts work for pre-training but fail in real applications where lexical variation is common. Removing the query eliminates the most obvious surface signal, forcing the model to learn deeper semantic patterns.

4.4.2 Relationship to Question Clusters

The ranking corruption detection task reveals an important insight: to identify whether a ranking has been perturbed, the model must learn to recognize the coherent structure of question clusters without ever seeing their center. This is a stronger learning signal than simply matching queries to candidates, because it requires understanding what properties make a set of questions belong together.

This insight has implications beyond pre-training. If models can learn to recognize cluster structure implicitly, they should also benefit from explicit cluster information at inference time. Chapter 5 builds on this insight by making clustering explicit: we use question clusters to ground LLM responses and to enforce consistency in document retrieval. The representations learned through QRP provide a foundation for recognizing when questions belong to the same equivalence cluster.

4.4.3 Complementarity of Distillation and Pre-Training

The best performance consistently comes from combining QRP with distillation, suggesting complementary benefits:

- **QRP** teaches internal representations of question semantics through self-supervised learning on large-scale data (18M examples). It provides broad coverage of question

types and linguistic patterns.

- **Distillation** transfers specific similarity judgments from the retrieval model during fine-tuning on task-specific data. It fine-tunes representations using retrieval model knowledge on the target distribution.

QRP provides robust initialization through broad coverage, while distillation adapts these representations to the specific task. The combination leverages both large-scale unsupervised pre-training and targeted knowledge transfer.

4.4.4 Limitations

The QRP approach, while effective, has limitations that inform its applicability. In terms of design trade-offs, our pre-training is specifically designed for question ranking. This specialization yields strong performance on the target task but limits transfer to other NLP applications: answer selection requires different input formats, document retrieval operates on longer texts, and general NLU tasks require broader understanding beyond semantic similarity. The question arises: is task-specific pre-training worth the reduced generality? Our results suggest yes for applications where question ranking is central, but practitioners should weigh this trade-off. Additionally, QRP distills knowledge from a retrieval model, inheriting both its strengths and weaknesses. A weak teacher produces noisy rankings that may limit pre-training effectiveness; while we demonstrate that even imperfect rankings provide useful signal, the quality ceiling is set by the teacher. This creates a bootstrapping challenge: better teachers yield better QRP, but training better teachers requires the resources that QRP aims to reduce. Regarding practical constraints, generating 18M pre-training examples requires substantial resources: a 38M-pair question database, large storage, and GPU infrastructure for embedding computation and pre-training. These requirements may limit accessibility for resource-constrained organizations, though we note that this cost is a one-time investment that can benefit multiple downstream applications, and is significantly lower than equivalent human annotation costs. Furthermore, current QRP uses only English data. Extending to multilingual settings would require multilingual retrieval models and question databases, resources that are less mature than their English counterparts. The implicit clustering structure we exploit may also manifest differently across languages with varying morphological complexity. Chapter 5 partially addresses the multilingual dimension through cross-lingual coherence analysis across six languages.

Despite these limitations, QRP offers practical benefits: reduced annotation requirements make DBQA more accessible to resource-constrained organizations, and lower variance simplifies both research reproducibility and production deployment.

4.5 Conclusion

This chapter addressed a key limitation in question ranking: the reliance on expensive annotated data for training effective models. We presented Question Ranking Pre-training (QRP), a novel task-specific pre-training method, and made the following contributions:

1. A self-supervised pre-training objective that distills knowledge from retrieval models through automatically generated ranking data, eliminating the need for manual annotation

2. A ranking corruption detection task that forces models to learn question semantics without access to the original query, leading to more robust representations

3. Extensive experiments demonstrating consistent improvements across three benchmarks: +1.05% P@1 on QRC (statistically significant, p-value=0.0005), +0.13% ROC-AUC on Quora-match, and +2.99% P@1 on SemEval-2016 transfer learning

4. Evidence that task-specific pre-training substantially outperforms general objectives (MLM, RTS, STS) on the same data, and that combining QRP with distillation yields further gains (+1.19% P@1 on QRC dataset)

5. Analysis showing that QRP reduces model variance by over 50%, improving stability and reproducibility across random seeds

Our findings establish that pre-training objectives should be designed to match the structure of downstream tasks: the ranking corruption detection task directly prepares models for question ranking in ways that token-level or sentence-level objectives cannot. Future work could extend this principle through multi-task pre-training, adaptive perturbation strategies, and multilingual settings. The methodology and pre-trained models are released to support such investigations.

Crucially, this chapter revealed that question retrieval implicitly creates *clusters* of semantically equivalent questions. When retrieving candidates for a query, the returned questions form a cluster around the same information need. By learning to recognize this cluster structure without seeing the cluster center (the query), models develop robust representations of question equivalence. This insight motivates Chapter 5, where we make clustering explicit: we use question clusters to ground LLM responses and train document retrieval models to be *cluster-coherent*, returning consistent results for all questions in an equivalence cluster.

Chapter 5

Question Clustering for Model Coherence

The previous chapters established that question retrieval implicitly creates clusters of semantically equivalent questions. When querying a database, the retrieval model returns questions that share the same underlying information need, forming a cluster around that need. Chapter 4 showed that learning to recognize this cluster structure, without access to the cluster center (the query), leads to robust representations of question equivalence. A natural question arises: can we leverage these question clusters to improve the behavior of downstream systems? Consider an LLM that correctly answers "*What is the boiling point of water?*" but fails on "*At what temperature does water boil?*". The model possesses the required knowledge yet cannot access it reliably across phrasings, a coherence failure rather than a knowledge gap. Can equivalent question clusters help the model overcome this fragility?

The works in this chapter explore a fundamental property that question clusters can help enforce: *coherence*. We define coherence as the ability of a system to produce consistent outputs for semantically equivalent inputs. A coherent question answering system should provide the same answer whether asked *"How many calories in a cucumber?"* or *"What is the calorie count of a cucumber?"*. Similarly, a coherent retrieval system should return the same documents, with the same rank, for both queries. A lack of coherence suggests that the system' s understanding is fragile and relies primarily on superficial patterns rather than on deeper semantic representations.

We investigate coherence in two complementary settings:

1. **Large Language Model Coherence**: We analyze whether state-of-the-art LLMs provide consistent answers when presented with different phrasings of the same question. Our analysis reveals significant coherence gaps across multiple models. We then show that augmenting prompts with clusters of equivalent questions, retrieved from a question database, substantially improves both coherence and accuracy. This technique, which we call *question-augmented generation* (q-RAG), leverages the redundant semantic signal from multiple equivalent phrasings to help LLMs better access their parametric knowledge. We further demonstrate that q-RAG's benefits can be distilled into model parameters through coherence-aware training, producing standalone models with improved coherence that surpass the inference-time approach.

2. **Document Retrieval Coherence**: We analyze whether dense retrieval models return consistent document rankings for equivalent queries. Our analysis reveals that even

well-trained models exhibit substantial sensitivity to query phrasing. We introduce a *Coherence Ranking Loss* that explicitly penalizes inconsistencies between rankings produced by equivalent queries, improving both coherence and retrieval accuracy.

The coherence theme emerged naturally from the work presented in previous chapters. While developing the QUADRo system (Chapter 3) and analyzing its outputs, we repeatedly observed a counterintuitive phenomenon: semantically equivalent questions, correctly identified as such by our retrieval system, can produce inconsistent answers when presented to LLMs. Similarly, the retrieval models themselves would sometimes return different documents for queries expressing the same information need. If question equivalence is a well-defined concept that we can model, as discussed in Chapters 3 and 4, why do existing systems, including both retrieval and generative models, still fail to behave consistently when presented with equivalent inputs? This observation motivated a systematic investigation of coherence across the components of modern QA pipelines.

The key insight connecting both applications is that question clusters provide a principled way to define and optimize for coherence. Rather than treating each query independently, we can use clusters of equivalent questions to (i) provide redundant signal that disambiguates user intent for LLMs, and (ii) define consistency constraints that regularize model behavior for retrieval.

This chapter makes the following contributions:

1. **Comprehensive LLM Coherence Analysis**: We provide systematic analysis showing that state-of-the-art LLMs (Mixtral-8x7B, Llama2-70B, Smaug-72B, Phi-3) exhibit significant coherence gaps, where models answer some phrasings correctly but fail on semantically equivalent alternatives.

2. **Question-Augmented Generation (q-RAG)**: We introduce q-RAG, a retrieval-based approach that supplements LLM prompts with clusters of similar questions from a 38-million question-answer database. Human evaluation demonstrates accuracy improvements of up to 9 percentage points and coherence improvements up to 28 points across multiple models and benchmarks.

3. **Multilingual Coherence Analysis**: We extend the coherence analysis to six typologically diverse languages (English, Italian, German, Chinese, Japanese, Vietnamese) and eleven models (3.8B-235B parameters), revealing that coherence correlates with accuracy ($\rho = 0.39$) and model size ($\rho = 0.42$), but varies significantly across languages and model families.

4. **Coherence Ranking Loss**: We introduce a novel loss function for training dense retrieval models that explicitly optimizes for ranking consistency across equivalent queries. The loss combines Query Embedding Alignment ($\mathcal{L}_{\text{QEA}}$) and Similarity Margin Consistency ($\mathcal{L}_{\text{SMC}}$) with standard relevance optimization.

5. **Comprehensive Evaluation**: We demonstrate that coherence-aware training improves both coherence metrics (up to +30% RBO) and accuracy metrics (up to +1.69% NDCG) across MS-MARCO, Natural Questions, 11 BEIR datasets, and TREC-DL 2019/2020, with benefits transferring to downstream applications including retrieve-and-rerank and RAG pipelines.

6. **Coherence-Aware LLM Training**: We present original experiments showing that q-RAG's coherence benefits can be distilled into model parameters through Direct Preference Optimization (DPO) and Supervised Fine-Tuning (SFT), achieving up to +3.93 EM points and +23.96 coherence points on cross-dataset evaluation, surpassing q-RAG's inference-time improvements.

The remainder of this chapter is organized as follows. Section 5.2 presents our analysis of LLM coherence and the question-augmented generation approach. Section 5.4 extends this work by investigating coherence-aware training through DPO and SFT. Section 5.5 presents the multilingual coherence analysis. Section 5.6 introduces the coherence ranking loss for document retrieval and presents experimental results. Section 5.7 provides a unified discussion connecting all applications. Finally, Section 5.8 concludes the chapter.

5.1 Preliminaries

This section recalls the key definitions from Chapter 2 and establishes the coherence metrics used throughout this chapter.

Question Equivalence. Two questions are semantically equivalent if they express the same information-seeking intent and accept the same set of correct answers (Definition 3.1.1). Both conditions are jointly necessary: questions with coincidentally overlapping answers but different intents are not equivalent (Section 2.2.2).

Question Clusters. A question cluster $C = \{q_1, \ldots, q_n\}$ groups all equivalent phrasings of the same information need. Table 5.1 provides examples across different domains.

Table 5.1: Examples of question clusters across different domains. Each cluster contains semantically equivalent questions with the same answer.

Domain	Cluster Members
Geography	What is the capital of France? Can you tell me what the capital of France is? Name the city that serves as the capital of France. Which city is France's capital?
Science	What is the average lifespan of a flea? How long does a flea typically live? Can you explain the typical duration of a flea's life cycle? What is the life expectancy of fleas?
Biology	What mechanism allows some sharks to retain warmth? How do certain sharks maintain body temperature? How does a select group of sharks stay warm internally? What enables some shark species to be warm-blooded?

Coherence Metrics. We measure coherence as the consistency of system outputs across questions within the same cluster. This chapter employs three complementary metrics:

1. **Semantic Coherence** (for LLM answers): The average pairwise cosine similarity of answer embeddings generated for equivalent questions, computed using all-mpnet-base-v2 for English experiments and LaBSE (Feng et al., 2022) for multilingual experiments. Higher values indicate more consistent answers regardless of correctness.

2. **Binary Coherence** (for discrete analysis): For factual questions with known correct answers, we categorize each cluster as:
   - *Coherent-correct*: All questions answered correctly
   - *Coherent-incorrect*: All questions answered incorrectly
   - *Incoherent*: Mixed results: the model possesses the knowledge (succeeds on some phrasings) but fails to access it consistently
3. **Ranking Coherence** (for document retrieval): The average Rank-Biased Overlap (RBO) (Webber et al., 2010) between document rankings produced for equivalent queries. RBO is top-weighted and produces values in $[0, 1]$.

Table 5.2 clarifies which coherence metrics are used in each experimental section. Note that accuracy is additionally evaluated on QRC via human evaluation (§5.2.4) and on Open-Domain QA datasets (§5.2.6), but coherence measurement requires multiple test questions per cluster, which only PopQA-TP and the generated multilingual clusters provide.

Table 5.2: Coherence metrics used in each experimental section.

Section	Coherence Metric	Dataset
§5.2.2 (Incoherence Analysis)	Semantic Coherence	PopQA-TP
§5.2.4 (q-RAG Effect)	Semantic Coherence	PopQA-TP
§5.2.4 (Per-cluster)	Binary Coherence	PopQA-TP
§5.4 (Per-cluster)	Binary/Semantic Coherence	PopQA-TP
§5.5 (Multilingual)	Semantic Coherence (LaBSE)	Generated clusters
§5.6 (Retrieval)	Ranking Coherence (RBO)	MS-MARCO, NQ

5.2 Coherence in Large Language Models

Large language models exhibit surprising sensitivity to input phrasing: small changes in wording can lead to dramatically different answers (Rabinovich et al., 2023; Voronov et al., 2024a; Mizrahi et al., 2024a).

In this section, we analyze the coherence of multiple LLMs on factual question answering tasks and introduce a retrieval-augmented approach to improve coherence through question clustering.

5.2.1 Experimental Setup

This section addresses the research questions summarized in Table 5.3.

Models

We analyze four state-of-the-art LLMs spanning different sizes, architectures, and training approaches:

- **Mixtral-8x7B** (56B parameters): A mixture-of-experts model that routes each token to a subset of expert networks, achieving strong performance with efficient inference (Jiang et al., 2024).

Table 5.3: Research questions for LLM and multilingual coherence analysis.

RQ	Research Question	Section
RQ1	How coherent are current LLMs when answering semantically equivalent questions?	§5.2.2
RQ2	Does augmenting prompts with similar questions (q-RAG) improve accuracy?	§5.2.4, §5.2.6
RQ3	Does q-RAG improve coherence across equivalent question phrasings?	§5.2.5
RQ4	Does q-RAG improve accuracy through better question understanding rather than external knowledge injection?	§5.2.6, §5.3.1
RQ5	Does coherence vary across languages and model sizes?	§5.5
RQ6	Can q-RAG's coherence benefits be distilled into model parameters through training?	§5.4

- **Llama2-70b** (70B parameters): Meta's open-source LLM trained on 2 trillion tokens with extensive safety fine-tuning (Touvron et al., 2023).
- **Smaug-72b** (72B parameters): A model specifically optimized for reasoning tasks through DPO training on challenging benchmarks like ARC and HellaSwag (Pal et al., 2024b).
- **Phi-3-mini** (3.8B parameters): A smaller but capable model trained on high-quality data, designed for deployment on resource-constrained devices (Abdin et al., 2024a).

These models were selected to represent the diversity of open-source LLMs available at the time of our experiments (early 2024): dense architectures (Llama2, Smaug, Phi-3) versus mixture-of-experts (Mixtral), scales ranging from 3.8B to 72B parameters, and different training focuses (general-purpose versus reasoning-optimized). Proprietary models such as GPT-4 and Claude were not included, as they are only accessible via provider APIs. This prevents full inspection or control of their underlying system instructions, which may affect reproducibility and introduce uncontrolled sources of bias in the analysis.

All experiments use temperature 0.001 to minimize randomness in generation. Models are run using 8×V100 32GB GPUs and Amazon Bedrock.

Datasets

Our evaluation requires two types of resources serving different purposes. First, benchmarks with annotated question clusters enable direct measurement of model coherence by comparing outputs across equivalent phrasings. Second, standard QA datasets without cluster annotations allow end-to-end evaluation of our retrieval-augmented approach in realistic settings where equivalent questions must be retrieved rather than provided.

Benchmarks with Annotated Clusters. We use two datasets containing pre-annotated clusters of semantically equivalent questions:

Question Ranking Corpus (QRC): The dataset developed in Chapter 3, containing clusters of semantically equivalent questions annotated through a rigorous human evaluation process. We extract 762 clusters from the test split, each containing at least 6 equivalent questions (1 test query + 5 support questions). This benchmark provides high-quality clusters but smaller scale.

PopQA-TP: A large-scale resource (Rabinovich et al., 2023) consisting of 118K entity-centric QA pairs divided into 14K clusters of paraphrased question variations. Unlike QRC, PopQA-TP contains larger clusters with more variations per information need. We use 5,518 clusters containing at least 10 questions each, using 5 as support questions and 5 as test queries. Evaluation uses exact match against reference entity answers.

Open-Domain QA Datasets. To evaluate the q-RAG approach end-to-end with retrieved (rather than oracle) support questions, we sample 500 queries from each of Natural Questions (Kwiatkowski et al., 2019), QuoraQP (Wang et al., 2020b), PAQ (Lewis et al., 2021), and TriviaQA (Joshi et al., 2017), for a total of 2,000 test queries spanning diverse question types and domains.

Evaluation Protocol

Different datasets require different evaluation approaches based on answer characteristics.

Automatic Evaluation (PopQA-TP). Since PopQA-TP answers are short entity names (e.g., *"Paris"*, *"1969"*), we use automatic exact match against reference answers as in the original work (Rabinovich et al., 2023). This enables large-scale evaluation across all 5,518 clusters.

Human Evaluation (QRC and Open-Domain QA). For QRC and Open-Domain QA, answers are free-form text that cannot be reliably evaluated through string matching. We use Amazon Mechanical Turk (AMT) with the following protocol. Each Human Intelligence Task (HIT) consists of evaluating 6 question-answer pairs: 1 control question and 5 experimental pairs. Control questions filter inattentive annotators using positive controls (simple factual questions) and negative controls (clearly incorrect answers).

Annotators classify each answer into four categories:

1. **Correct and Natural**: The answer correctly addresses the question and is expressed naturally without extraneous information.

2. **Correct but Not Natural**: The answer contains the correct information but includes irrelevant content, repetitions, or awkward phrasing. For example, answering *"How many calories in a cucumber?"* with *"An average pineapple (900g) contains 452 calories, which is higher compared to the 45 calories of a whole cucumber"*. In this case the answer is technically correct but indirect.

3. **Incorrect but Natural (Hallucination)**: The answer appears fluent and confident but contains factually wrong information.

4. **Incorrect**: The answer is clearly wrong or unrelated.

Annotators verify information using search engines when needed. Selection criteria require: (i) HIT approval rate >95%, (ii) minimum 1000 approved HITs, (iii) Master qualification. Each HIT is paid $0.50.

Coherence Measurement. Beyond accuracy, we evaluate the consistency of model outputs across equivalent questions. Semantic coherence measures the average pairwise cosine similarity of answer embeddings for equivalent questions, computed using all-mpnet-base-v2. Low coherence indicates that model success depends on specific phrasing rather than genuine understanding: the model may answer correctly for some phrasings while failing on semantically equivalent reformulations.

Table 5.4: Baseline LLM coherence on PopQA-TP. Coherence is measured as average pairwise cosine similarity of answer embeddings within clusters (Equation 2.4). Accuracy (EM) shown for reference.

Model	Coherence	Accuracy (EM)
Mixtral-8x7B	53.21	16.72
Llama2-70b	81.36	15.69
Phi-3-mini	43.46	5.01
Smaug-72b	54.51	13.89

5.2.2 Analysis of LLM Incoherence

Before introducing our method, we first quantify the coherence gap in current LLMs. We measure baseline coherence by presenting each model with questions from PopQA-TP clusters (5 questions per cluster) and computing answer coherence across the cluster.

Table 5.4 presents baseline coherence alongside accuracy (computed as EM) for reference.

Key Findings

Coherence Varies Across Models and Reveals Understanding Gaps (RQ1). Before running these experiments, we expected coherence to correlate strongly with model capability. The results revealed a more nuanced picture. Among models of comparable size (56–72B parameters), coherence varies dramatically: Llama2-70b achieves 81.36 while Mixtral-8x7B and Smaug-72b score only $\approx$ 53–54, despite similar accuracy. This shows that coherence is a property distinct from raw performance, shaped by training methodology and data composition rather than by scale alone. Models with lower coherence, specifically Phi-3-mini at 43.46 and Mixtral at 53.21, show highly variable behavior across equivalent questions. When a model correctly answers *"What year was Lincoln born?"* but fails on *"When was Abraham Lincoln born?"*, this cannot be a knowledge gap because the required knowledge is identical. Instead, it indicates failure to properly understand one of the phrasings and retrieve the correct information from internal knowledge. Notably, coherence does not guarantee accuracy: Llama2 has the highest coherence but middling accuracy, demonstrating that the two metrics capture complementary aspects of model behavior.

5.2.3 Question-Augmented Generation

The coherence analysis reveals that LLMs often possess the knowledge needed to answer questions correctly but fail to access it consistently across equivalent phrasings. This suggests that incoherence arises not from missing knowledge but from difficulty in inferring the user' s intent from a single query formulation. Different phrasings can activate different patterns within the model' s parameters, leading it to retrieve distinct and sometimes incorrect information. We therefore hypothesize that providing multiple equivalent questions can help the model more reliably identify the underlying information need and access the relevant parametric knowledge.

Method Overview

We propose Question-RAG (q-RAG), an evolution of RAG (Lewis et al., 2020b) that supplements the input question with semantically equivalent questions retrieved from a large

database. Unlike classical RAG, which retrieves documents to provide new factual information, q-RAG retrieves questions to help the model better understand the user's intent. This shift from document retrieval to question retrieval, introduced in Chapter 3, represents a fundamental change in how retrieval augmentation can benefit language models. The pipeline consists of two stages:

- **Stage 1: Question Retrieval.** We leverage the DBQA system developed in Chapters 3–4, which consists of (i) a 38-million question-answer pair database combining curated sources with filtered PAQ data, and (ii) a fine-tuned MiniLM-L12-v2 bi-encoder for dense retrieval (see Section 4.1.1 for details). Given an input query q, the system retrieves the k most similar questions. We call these retrieved questions *Support Questions* (SQs). As described in Chapter 3, the database covers diverse open-domain topics (geography, science, history, entertainment, etc.), ensuring broad coverage for general-purpose question answering without domain-specific limitations.
- **Stage 2: Augmented Generation.** We construct a prompt that includes both the original question and the retrieved SQs (and optionally their answers). The prompt instructs the LLM to use the support questions to disambiguate or clarify the user's original intent before generating an answer. The specific prompt configurations are detailed below.

The key insight is that SQs contain no new factual knowledge: they are only alternative phrasings of the same information need. It follows that any improvement in accuracy must therefore come from better understanding of the question, not from external knowledge injection.

Prompt Design

The effectiveness of q-RAG depends on how support questions are presented. We frame the task as a Frequently Asked Questions (FAQ) system, where the LLM leverages similar questions to understand user intent.

Prompt effectiveness varies across models due to differences in training data and instruction-tuning procedures. We conducted preliminary experiments on a held-out development set to identify optimal prompts for each model, testing variations in framing, instruction detail, SQs placement, and uncertainty handling.

All prompts share a common structure with four key components:

1. **Role specification**: The model acts as an FAQ system leveraging similar questions
2. **Context description**: Explanation that provided questions are semantically similar and should clarify ambiguity
3. **Behavioral rules**: Instructions governing context usage and response format (e.g., no meta-commentary about the FAQ system, use context for inference, provide concise answers, acknowledge uncertainty rather than hallucinate)
4. **Dataset-specific constraints**: For PopQA-TP, we require entity-based answers

Table 5.5 summarizes the configurations; complete prompt templates are provided in Appendix A.1.

Table 5.5: Prompt structure for different configurations. Full templates in Appendix A.1.

Configuration	Key Components
Base	QA system role, uncertainty instruction ("say I don't know")
q-RAG (questions)	FAQ system role, k similar questions as context, 5 behavioral rules
q-RAG (Q+A pairs)	FAQ system role, k question-answer pairs as context, 5 behavioral rules
PopQA-TP variants	Additional constraint: "answer must be an entity or entity list"
Q Generation	Two-step: first generate paraphrases (JSON output), then use as SQs
Chain-of-Thoughts	Single-pass reasoning over implicit question variations

Table 5.6: LLM accuracy with and without question prompting using gold-standard support questions (5 equivalent questions from the same cluster). QRC uses human evaluation; PopQA-TP uses exact match. Bold indicates better result.

	QRC: Correct		QRC: Natural		PopQA-TP: EM	
Model	Base	q-RAG	Base	q-RAG	Base	q-RAG
Mixtral-8x7B	78.48	**81.10**	40.29	**46.95**	16.72	**20.30**
Llama2-70b	77.69	**84.38**	54.20	**62.73**	15.69	**17.37**
Phi-3-mini	68.76	**71.78**	54.46	**58.53**	5.01	**5.14**
Smaug-72b	**83.59**	72.31	**68.77**	57.21	13.89	**18.41**

5.2.4 Accuracy results on Question Equivalence Benchmarks

To isolate the effect of question prompting from potential retrieval errors, we first evaluate using gold-standard support questions: semantically equivalent questions from the same annotated cluster, rather than questions retrieved by our DBQA system. This controlled setting allows us to assess the upper bound of question prompting effectiveness. Table 5.6 compares accuracy in two conditions: *Base*, where models receive only the input question, and *q-RAG*, where models receive the input question augmented with 5 gold-standard equivalent questions from the same cluster.

Key Findings

Question Prompting Generally Improves Accuracy (RQ2). Three of four models show consistent accuracy improvements across both benchmarks. Llama2 shows the largest gains on QRC: +6.69 points on Correct and +8.53 points on Natural, with +1.68 on PopQA-TP. Similarly, Mixtral improves by +2.62/+6.66 on QRC and +3.58 on PopQA-TP. Finally, Phi-3-mini improves by +3.02/+4.07 on QRC and +0.13 on PopQA-TP.

Smaug Exhibits Anomalous Behavior. Smaug-72b uniquely shows decreased accuracy on QRC (−11.28 points for Correct, −11.56 points for Natural) despite improved accuracy on PopQA-TP (+4.52 points). The pattern across datasets is suggestive: Smaug improves on the harder, entity-based PopQA-TP benchmark but degrades on the easier, free-form QRC benchmark.

The most plausible explanation involves the interaction between Smaug's reasoning-oriented

training and answer format requirements. Smaug was optimized through DPO training on reasoning benchmarks (ARC, HellaSwag), which instilled strong uncertainty-handling mechanisms. When presented with multiple question phrasings, the model appears to interpret the redundancy as a signal of ambiguity, triggering conservative "I don't know" responses even for questions it would otherwise answer correctly. Table 5.7 illustrates this pattern.

Table 5.7 shows representative failure cases where Smaug becomes overly conservative with question prompting.

Table 5.7: Smaug-72b failure cases: the model becomes overly conservative with question prompting, refusing to answer questions it handles correctly without SQs.

Q: Abraham was born in what year? **Base:** Abraham Lincoln was born in 1809. **Q-prompt:** I'm sorry, but as an AI text-based model, I don't have real-time information or access to specific personal data like birth dates. Abraham could refer to various people throughout history.
Q: Who best soccer player in the world? **Base:** Lionel Messi and Cristiano Ronaldo are often considered as two of the best soccer players in the world due to their numerous achievements... **Q-prompt:** I don't know.

While we cannot definitively determine which factor dominates, the pattern across datasets is suggestive: Smaug improves on PopQA-TP which is a harder, entity-based dataset but degrades on QRC which is an easier free-form dataset. This suggests that the most plausible hypothesis is the interaction between Smaug' s reasoning-oriented training and the answer-format requirements. The conservative *"I don' t know"* responses in Table 5.7 further support this interpretation: the model' s uncertainty mechanisms, which are helpful in reasoning tasks, become overly sensitive when redundant question context is introduced.

5.2.5 Coherence results on Question Equivalence Benchmarks

Beyond accuracy, we also evaluate how q-RAG affects coherence. We measure coherence only on PopQA-TP, since this dataset provides multiple test questions per cluster (5 test + 5 support). In contrast, QRC clusters contain only a single test question after reserving five for support, which makes pairwise coherence computation impossible. Table 5.8 presents the obtained results.

Table 5.8: Coherence improvement with question prompting on PopQA-TP. Coherence measured using Equation 2.4. All models show substantial improvement.

Model	**Base**	**q-RAG**	**Δ**
Mixtral-8x7B	53.21	**81.21**	+28.00
Llama2-70b	81.36	**84.51**	+3.15
Phi-3-mini	43.46	**61.71**	+18.25
Smaug-72b	54.51	**75.97**	+21.46

Key Findings

Coherence Improves Substantially for All Models (RQ3). Even Smaug, which shows decreased accuracy on QRC, improves coherence by +21.46 points. Mixtral shows the largest improvement (+28.00 points). This demonstrates that question prompting helps models produce more consistent answers regardless of accuracy effects.

All four models show improved coherence with q-RAG, with gains ranging from +3.15 to +28.00 points. The improvement is inversely related to baseline coherence: Mixtral and Phi-3, which had the lowest baseline coherence (53.21 and 43.46), show the largest gains (+28.00 and +18.25 points respectively). Conversely, Llama2, which already exhibited high baseline coherence (81.36), shows modest improvement (+3.15 points), suggesting a ceiling effect. Interestingly, coherence gains do not correlate with accuracy gains. Phi-3 shows minimal accuracy improvement (+0.13 points) but substantial coherence improvement (+18.25 points), while Smaug shows the opposite pattern: the largest accuracy gain (+4.52 points) but moderate coherence improvement. This suggests that q-RAG affects the two properties through partially independent mechanisms.

Per-Cluster Analysis

To understand the nature of coherence improvement, we analyze the distribution of correct answers within each PopQA-TP cluster. Each cluster contains 5 test questions, so we count how many receive correct answers.

Interpretation. A perfectly coherent model produces a bimodal distribution with mass only at 0 and 5:

- **5/5 correct**: The model knows the answer and consistently retrieves it
- **0/5 correct**: The model lacks the knowledge entirely

Intermediate cases (1-4 correct) indicate genuine incoherence: the model possesses the knowledge, which is evidenced by success on some query phrasings, but fails to access it consistently. These represent opportunities for improvement.

Results. Figure 5.1 shows this distribution comparing baseline against q-RAG. Key observations:

1. **All models show baseline incoherence**: 6–16% of clusters fall in the intermediate region (1-4 correct), with Mixtral showing the highest incoherence rate (15.7%).
2. **Question prompting shifts mass toward extremes**: For all models, mass moves from the incoherent region to 0/5 and 5/5.
3. **Increase in 0/5 clusters**: Three models (Mixtral, Phi-3, Smaug) show more 0/5 clusters with q-RAG. We hypothesize this reflects *increased appropriate uncertainty*: question prompting helps models recognize when they truly lack knowledge, leading to consistent *"I don't know"* responses rather than inconsistent guessing. The failure cases in Table 5.7 support this interpretation. The exception is Llama2-70b, which actually shows a *decrease* in 0/5 clusters (from 4210 to 4124) with corresponding increase in 5/5 clusters (from 607 to 704). This is consistent with its already high baseline coherence: q-RAG helps Llama2 convert uncertain responses into correct ones rather than triggering additional uncertainty.

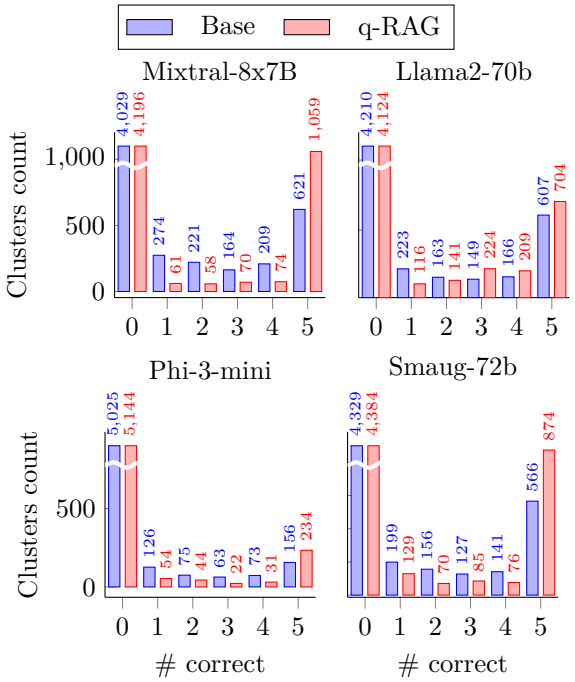

Figure 5.1: Distribution of correct answers per cluster (0-5) on PopQA-TP. Blue bars: baseline; red bars: question prompting. A coherent model shows mass at extremes (0 and 5); mass in the middle (1-4) indicates incoherence. Question prompting shifts distributions toward the extremes for all models.

5.2.6 End-to-End Evaluation with Retrieved Questions

The previous experiments demonstrated that q-RAG improves both accuracy and coherence when using "gold-standard" SQs from annotated clusters. However, in practical applications such clusters are not available: support questions must be retrieved from the database, potentially introducing noise from imperfect retrieval. This section evaluates whether q-RAG benefits persist in a realistic end-to-end setting. Given the cost of human evaluation and the number of configurations to test, we focus this analysis on Mixtral-8x7B, which showed the largest coherence improvement (+28 points) in previous experiments on PopQA-TP. The evaluation is conducted on the previously introduced Open-Domain QA Datasets consisting of 2000 open-domain questions.

Configurations

We compare q-RAG against several baselines to isolate the contribution of different components. Specifically, we want to understand: (i) whether retrieved questions alone provide benefit, (ii) whether adding answers to the retrieved questions helps further, and (iii) how q-RAG compares to traditional document-based RAG.

- **Base:** Direct question answering without retrieval
- **q-RAG (only question):** Input question + top-k similar questions from our DBQA system
- **q-RAG (question-answer):** Input question + top-k question-answer pairs from our DBQA system

- **RAG (paragraphs):** input question + top-k Wikipedia paragraphs via DPR

We evaluate $k \in \{1, 3, 5\}$, consistent with the gold-standard benchmarks where 5 equivalent questions per cluster are available. Additionally, as shown in Figure 5.2, performance gains diminish beyond $k = 3$, suggesting that 3-5 support questions provide sufficient semantic redundancy without introducing noise.

Results

Table 5.9 and Figure 5.2 presents results as k varies from 1 to 5.

Table 5.9: End-to-end accuracy on the Open-Domain QA dataset (2,000 questions, Mixtral-8x7B) with different augmentation strategies.

Configuration	Correct	Correct & Natural
Base (no retrieval)	68.6	49.0
q-RAG, only question (k=1)	77.1	54.7
q-RAG, only question (k=3)	77.5	56.4
q-RAG, only question (k=5)	**78.2**	55.3
q-RAG, question-answer (k=1)	74.9	**57.6**
q-RAG, question-answer (k=3)	73.4	55.7
q-RAG, question-answer (k=5)	74.0	55.5
RAG, paragraphs (k=1)	68.2	49.0
RAG, paragraphs (k=3)	70.1	52.6
RAG, paragraphs (k=5)	74.7	53.9

Key Findings

The end-to-end evaluation reveals three main results:

q-RAG Benefits Persist with Retrieved Questions (RQ2). In end-to-end evaluation with retrieved SQs, q-RAG consistently outperforms both the Base and RAG baselines. With questions only, q-RAG achieves 78.2% accuracy at k=5, compared to 68.6% for Base model (+9.6 points), and 74.7% for paragraph-based RAG (+3.5 points). This confirms that q-RAG's benefits persist in realistic deployment conditions. Performance increases with k for paragraph-based RAG, from 68.2% at k=1 to 74.7% at k=5, but for question-based approaches, the benefit is already strong at k=1 with diminishing returns as k increases (Figure 5.2).

Questions Alone vs Question-Answer Pairs. Comparing q-RAG only question with q-RAG question-answer reveals an interesting pattern. Questions alone achieve higher correctness (78.2% vs 74.0% at k=5), while question-answer pairs achieve comparable naturalness (55.5% vs 55.3%). This suggests that for pure accuracy, the semantic signal from multiple question phrasings is more valuable than the factual grounding from pre-computed answers confirming that question understanding is the primary driver of improvement. Additionally, answers may sometimes introduce noise or conflicting information that slightly hurts performance.

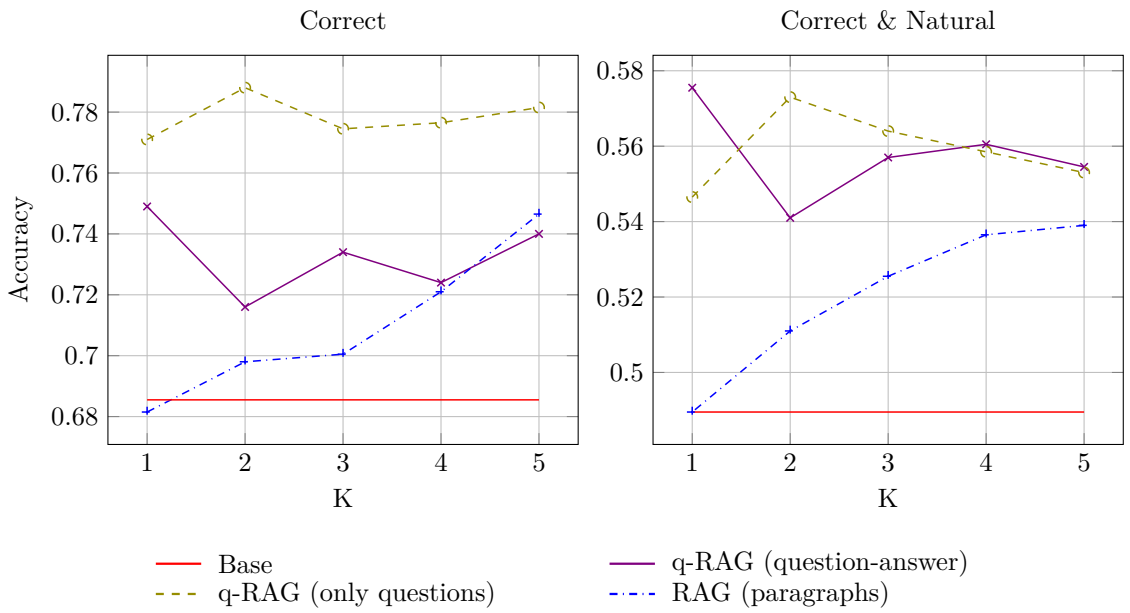

Figure 5.2: End-to-end accuracy on 2,000 open-domain questions (Mixtral-8x7B) as k increases from 1 to 5. Left: correctness only. Right: correct and natural answers. All augmentation strategies improve over the baseline, with q-RAG (question-answer) achieving best performance.

Table 5.10: Standalone retrieval performance (top-1) on the Open-Domain QA dataset. DBQA retrieves question-answer pairs; RAG retrieves Wikipedia paragraphs via DPR.

Metric	DBQA	RAG
Correct	56.9	56.2
Correct & Natural	37.4	31.7

Q-RAG Outperforms Classical RAG (RQ4). Question-based approaches consistently outperform paragraph-based RAG. At k=5, q-RAG (questions only) achieves 78.2% accuracy compared to 74.7% for classical RAG (+3.5 points). This improvement is notable because SQs contain no new factual information: *they only help the model understand the question better, supporting our hypothesis that LLM incoherence is fundamentally an understanding problem, not a knowledge problem.*

5.3 Ablation studies

5.3.1 Retrieval System Comparison: DBQA vs Traditional RAG

A natural question is whether q-RAG's superiority stems from the prompting technique itself or from retrieval quality differences. To disentangle these factors, we compare the standalone retrieval performance of both systems on the same 2,000 open-domain queries, evaluating retrieved content *without* LLM generation.

For DBQA, we evaluate the pre-computed answer associated with the top-1 retrieved question. For RAG, we evaluate whether the top-1 Wikipedia paragraph contains the correct answer. Table 5.10 presents results.

DBQA and traditional RAG achieve comparable correctness (56.9% vs 56.2%), indicating that the retrieval components have similar quality. The difference in naturalness (+5.7% for DBQA) is expected: paragraphs contain more information and do not directly answer the query, making them less likely to be rated as natural. This result is important because it demonstrates that q-RAG's improvement over traditional RAG (Section 5.2.6) does not stem from better retrieval, but it comes from the nature of the augmentation. Semantically equivalent questions help the model understand user intent, whereas paragraphs primarily provide factual grounding. The former addresses an understanding problem; the latter addresses a knowledge problem.

5.3.2 Support Questions: Retrieval vs Generation

The previous experiments used DBQA retrieval to obtain support questions. However, q-RAG is agnostic to the source of SQs: any method that produces semantically equivalent questions could work. This raises a practical question: can LLMs generate effective support questions on-the-fly, eliminating the need for a pre-built question database? We compare three approaches: retrieval from DBQA, LLM-based generation, and Chain-of-Thought (CoT) prompting. All prompts are available in Appendix A.1.

Configuration

We evaluate four configurations to compare different sources of support questions:

- **Base:** Baseline without support questions
- **q-RAG:** Retrieved questions from 38M question database
- **Gen-q-RAG:** LLM generates 5 paraphrases, then uses them as SQs
- **CoT-q-RAG:** A single-pass prompt that encourages the model to consider alternative phrasings before answering, in line with the intuition of Wei et al. (2022c). The prompt first asks the model to reflect on possible rewordings of the question and then use this reasoning to generate the final answer.

Results

Table 5.11 presents results on 500 queries randomly sampled from NQ, Quora, PAQ, and TriviaQA, using Mixtral-8x7B with k=5.

Table 5.11: Comparison of methods for obtaining support questions. DBQA retrieval achieves best performance.

Configuration	Correct	Correct & Natural
Base	73.5	52.5
q-RAG	**79.3**	**62.3**
Gen-q-RAG	78.0	58.8
CoT-q-RAG	75.3	58.5

Key Findings

All SQ Methods Improve Over Base Method. (RQ1) Even the weakest approach (CoT at 75.3%) outperforms Base method (73.5%), confirming that redundant question information helps regardless of source.

DBQA Retrieval Outperforms Generation. Retrieved questions achieve the best performance with +5.8% improvement over Base, compared to +4.5% for Gen-q-RAG and +1.8% for CoT-q-RAG. This is notable because generated and retrieved questions have comparable semantic equivalence (95% vs 92% as shown in the qualitative analysis below and Table 5.12). The difference lies in the diversity of phrasings: while LLMs tend to produce surface-level transformations, retrieved questions from real users expose different conceptual angles on the same information need.

CoT Underperforms Explicit SQs. CoT-q-RAG combines question generation and answering in a single pass, asking the model to consider alternative phrasings as part of its reasoning before generating an answer. Despite the intuition that models might not need explicit SQs if they can reason about them internally, this approach achieves only +1.8% improvement over baseline which is significantly less than both q-RAG (+5.8%) and Gen-q-RAG (+4.5%). This suggests that explicit support questions in the prompt provide a stronger and more reliable signal than implicit reasoning about potential reformulations.

Qualitative Analysis

To understand why retrieved questions outperform generated ones, we manually evaluated the semantic similarity between input queries and their support questions. We annotated 100 SQs generated by the LLM and 100 SQs retrieved through QRS, assessing whether each SQ was semantically equivalent to the input query. The results show comparable equivalence rates: 95% of generated questions and 92% of retrieved questions are semantically equivalent to the input. However, qualitative inspection reveals important differences in the *nature* of the variations:

- **Generated questions**: Surface-level transformations (synonyms, syntax changes)
- **Retrieved questions**: Diverse framings from real users that expose additional facets

Table 5.12 illustrates these differences.

Table 5.12: Generated vs retrieved support questions. Retrieved questions often expose different conceptual framings.

Input	Generated	Retrieved
Is it dangerous to eat expired yogurt?	Is consuming out-of-date yogurt hazardous to one's health?	How long after the expiration date is yogurt safe?
How do you calculate dimensions?	What is the method for determining dimensions?	How do you work out a volume of a shape?

Retrieved questions reframe queries in ways that may activate broader parametric knowledge. The organic diversity of real user questions provides richer signal than LLM-generated paraphrases.

Efficiency Comparison

Beyond accuracy, practical deployment requires consideration of computational costs. This analysis compares the latency of q-RAG versus Gen-q-RAG and CoT-q-RAG approaches for obtaining support questions.

q-RAG requires only ∼10ms per query: the 33M parameter MiniLM encoder generates the query embedding, followed by a FAISS nearest-neighbor lookup over the 38M question index. In contrast, LLM-based approaches are significantly slower: Gen-q-RAG requires ∼2-5 seconds to generate 5 paraphrases before answering, while CoT-q-RAG, despite being a single-pass approach, still requires ∼1-3 seconds as the model must reason about alternative phrasings before generating an answer. This represents a 200-500× speedup for q-RAG over generation-based methods.

Combined with the accuracy results from Section 5.3.2, this analysis shows that q-RAG achieves both better performance and dramatically lower latency than generation-based alternatives, making it practical for production deployment.

5.3.3 Summary

The ablation studies confirmed that retrieved questions outperform both LLM-generated paraphrases and chain-of-thought reasoning for q-RAG, and that DBQA retrieval provides comparable quality to traditional RAG with substantially lower latency.

The experiments in previous sections demonstrated that q-RAG improves coherence at inference time, but this approach requires retrieval infrastructure at deployment. A natural question arises: *can we internalize this coherence signal during training, producing a standalone model with improved coherence?* This mirrors the approach we take for retrieval in Section 5.6, where CR Loss optimizes for ranking consistency during training rather than compensating at inference time. The following section investigates this direction for LLMs.

5.4 Coherence-Aware LLM Training

The previous sections demonstrated that q-RAG improves LLM coherence at inference time by augmenting prompts with semantically equivalent questions. However, this approach requires maintaining a retrieval system at deployment. This section presents original work conducted for this thesis, investigating whether Direct Preference Optimization (DPO) and Supervised Fine-Tuning (SFT) can distill q-RAG's benefits directly into model parameters.

5.4.1 Methodology

The key insight is that q-RAG responses can serve as training signal: if a model augmented with equivalent questions produces better, more coherent answers, we can use these responses to train the base model directly.

We construct training data from the QRC corpus (Chapter 3), selecting clusters where at least one similar question has a positive label. For each cluster:

1. **Generate baseline responses**: For each question in the cluster, generate an answer using the base model with temperature 0 (greedy decoding). These serve as *rejected* responses for DPO.

Table 5.13: Effect of coherence-aware training on PopQA-TP. EM = Exact Match accuracy (%), Coh = answer coherence measured as average within-cluster similarity (%).

Model	Training	EM	F1	Coh	Δ Coh
Phi-3-mini	Base	5.01	6.69	43.46	–
	q-RAG	5.14	7.01	61.71	+18.25
	DPO	8.86	**10.75**	65.25	+21.79
	SFT	**8.94**	10.74	**67.42**	**+23.96**
Mistral-7B	Base	9.54	11.11	49.87	–
	q-RAG	10.41	11.95	53.98	+4.11
	DPO	**11.26**	**12.66**	**55.36**	**+5.49**
	SFT	10.38	12.01	53.97	+4.10

2. **Generate q-RAG responses**: Using the cluster's questions and their reference answers as context, following the q-RAG prompting strategy from Section 5.2.3, generate an answer for the cluster's seed question. This response is then mapped to all questions in the cluster as the *chosen* response.

3. **Quality filtering**: We use Claude Sonnet 4.5 to verify that each chosen response correctly answers its associated question, removing pairs where the mapped answer is incorrect for the specific question phrasing.

This procedure yields 25,450 training samples and 4,230 development samples. Crucially, the training signal comes from the model's own q-RAG-augmented outputs, making this a form of self-distillation where inference-time coherence is distilled into model parameters.

We compare two training objectives: (i) DPO which trains the model to prefer chosen over rejected responses, directly optimizing for the preference "*given equivalent questions, produce consistent answers*", and (ii) SFT which trains the model to generate chosen responses directly, providing a simpler baseline. We evaluate on Phi-3-mini and Mistral-7B-Instruct using full fine-tuning with DeepSpeed ZeRO Stage 3 on 8 NVIDIA L40S GPUs. Hyperparameters are selected via grid search: learning rate from 5×10^{-7} to 1×10^{-6}, DPO $\beta \in \{0.05, 0.1, 0.3, 1.0\}$, training for 3 epochs with early stopping. No system prompt is used during training to preserve generalization.

5.4.2 Results

Since the QRC test set requires human annotation, we evaluate on PopQA-TP (Rabinovich et al., 2023), the same benchmark used in Section 5.2.3. This tests whether coherence improvements transfer across datasets. Table 5.13 presents the results.

Key Findings

Both DPO and SFT Improve Coherence Beyond q-RAG (RQ6). Both training approaches successfully transfer and amplify q-RAG's coherence benefits into model parameters. Phi-3-mini shows the largest gains: EM improves by +3.93 points (5.01% → 8.94%) and coherence increases by +24.0 points (43.46 → 67.42) with SFT, substantially exceeding q-RAG's coherence of 61.71. Mistral-7B shows consistent improvements: +1.72 EM and +5.49 coherence points with DPO, again surpassing q-RAG (55.36 vs 53.98). These results

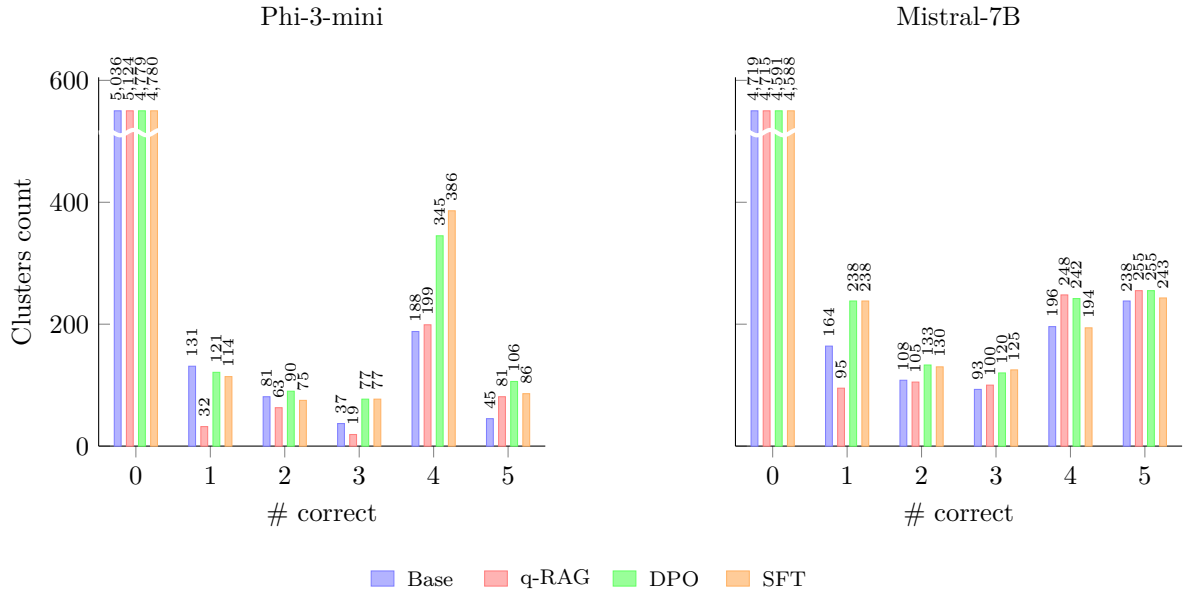

Figure 5.3: Distribution of correct answers per cluster (0–5) on PopQA-TP comparing baseline, q-RAG (inference-time), and coherence-aware training (DPO/SFT). For Phi-3-mini, q-RAG shows minimal change in cluster distribution despite large coherence improvement (+18.25 points), while DPO/SFT substantially reduce fully-incoherent clusters (0/5) and increase highly-coherent clusters (4–5/5). For Mistral-7B, all methods show similar patterns with training approaches achieving the largest shifts.

demonstrate that coherence is a learnable property that can be optimized during training to exceed inference-time augmentation.

Smaller Models Benefit More, With Clear Distribution Shifts. Phi-3-mini's relative improvement (+78% EM with SFT) substantially exceeds Mistral-7B's (+18% EM with DPO), aligning with our earlier finding that smaller models exhibit larger coherence gaps (Section 5.2.2). The cluster distribution in Figure 5.3 reveals an important distinction between inference-time and training-time approaches. For Phi-3-mini, q-RAG improves coherence substantially (+18.25 points) but shows minimal change in cluster distribution: fully incoherent clusters (0/5 correct) actually increase slightly from 5036 to 5124, while highly coherent clusters (4–5/5 correct) increase modestly from 233 to 280. In contrast, DPO and SFT produce dramatic distributional shifts: 0/5 clusters decrease to 4779–4780, while 4–5/5 clusters nearly double to 451–472. This suggests that q-RAG improves answer similarity within clusters without necessarily improving correctness, while training-time approaches fundamentally change which questions the model answers correctly.

DPO and SFT Show Complementary Strengths With Cross-Dataset Transfer. Neither training approach dominates uniformly: for Mistral-7B, DPO achieves higher coherence (+5.49 vs +4.10) and F1 (+1.55 vs +0.90); for Phi-3-mini, SFT achieves higher coherence (+23.96 vs +21.79) while both substantially exceed q-RAG. The choice may depend on the deployment scenario, with DPO explicitly optimizing preferences while SFT provides a simpler training signal. Importantly, training on QRC and evaluating on PopQA-TP demonstrates that coherence improvements generalize beyond the training distribution as the models learn to handle equivalent question phrasings more consistently, not merely to memorize QRC-

specific patterns.

5.4.3 Discussion

These results suggest that inference-time coherence techniques like q-RAG can be distilled, and even surpassed, through preference learning or supervised fine-tuning. The key insight is that coherence, rather than being an emergent property that requires external intervention at inference time, can be explicitly optimized during training. Notably, both DPO and SFT exceed q-RAG's coherence improvements while eliminating the need for retrieval infrastructure at deployment. In deployment scenarios where latency or infrastructure constraints preclude retrieval augmentation, training-time approaches offer not just an alternative but a superior solution.

The connection to Section 5.6 is instructive. CR Loss improves retrieval coherence through training-time optimization by aligning query embeddings and similarity margins across equivalent questions; DPO and SFT improve LLM coherence through the same principle by aligning generated answers. Both demonstrate that coherence, initially observed as an inference-time phenomenon, can be internalized through appropriate training objectives that leverage question cluster structure. This suggests a broader research direction: any task exhibiting coherence gaps may benefit from training objectives that explicitly optimize for consistency across semantic equivalents.

Several limitations warrant acknowledgment. First, the quality filtering step relies on an external LLM (Claude Sonnet 4.5), which may introduce biases in determining answer correctness. Future work could explore self-consistency filtering or human annotation for higher-stakes applications. Second, we evaluate only on factoid QA with short answers; generalization to open-ended questions, multi-hop reasoning, or longer-form generation remains unexplored. Third, the absolute accuracy on PopQA-TP remains modest (10% EM), though this reflects the dataset's difficulty as it is designed to challenge even large models, rather than a limitation of the approach. The key finding is the relative improvement in coherence metrics, demonstrating that the training signal successfully transfers.

Having established that coherence can be improved through both inference-time (q-RAG) and training-time (DPO/SFT) approaches for English, a natural question arises: *do these coherence patterns generalize across languages?* The following section extends our analysis to six typologically diverse languages.

5.5 Multilingual Coherence Analysis

The previous sections analyzed coherence exclusively on English questions. However, the equivalence relation that defines clusters should be language-independent, while the *surface realization* of equivalent questions varies dramatically across languages due to differences in morphology, syntax, and writing systems. This section extends our analysis to six typologically diverse languages, examining how coherence varies with model size, architecture, and language characteristics.

This question is particularly relevant for question clusters. The equivalence relation that defines clusters (Section 5.1) should be language-independent: two questions are equivalent if they share the same information-seeking intent, regardless of the language in which they are expressed. However, the *surface realization* of equivalent questions varies dramatically across languages due to differences in morphology, syntax, and writing systems. Languages with

richer morphological systems, such as German and Italian, allow more lexical variations for the same semantic content, potentially increasing the space of equivalent phrasings a model must handle consistently.

We therefore extend our coherence analysis to six typologically diverse languages, examining how coherence varies with model size, architecture, and language characteristics.

5.5.1 Multilingual Question Clusters

The equivalence relation that defines question clusters (Section 5.1) is inherently language-independent: two questions are equivalent if they share the same information-seeking intent, regardless of the language in which they are expressed. A question asking about the capital of France remains semantically equivalent whether phrased in English (*"What is the capital of France?"*), Italian (*"Qual è la capitale della Francia?"*), or Chinese (*"法国的首都是什么"*). The answer, Paris, is the same across all formulations.

However, equivalent questions may exhibit highly diverse *surface realizations* across languages, driven by morphological, syntactic, and script-level differences. This diversity gives rise to specific coherence challenges:

Morphological Complexity: languages differ substantially in how much information is encoded through word inflection. German and Italian mark grammatical relationships through case endings, verb conjugations, and gender agreement, while Vietnamese and Chinese, as analytic languages, encode grammatical relationships through word order and particles rather than inflection (Bentz and Berdicevskis, 2016; Coupé et al., 2019).

This typological difference *may* affect coherence: if morphologically simpler languages constrain the space of possible surface variations for a given semantic content, models might exhibit higher coherence simply because there are fewer distinct phrasings to handle consistently. However, this hypothesis requires empirical validation, as other factors (e.g., pre-training data distribution) may dominate.

Script and Tokenization: languages using logographic (Chinese) or syllabic (Japanese) writing systems undergo different tokenization compared to alphabetic languages. These differences affect how questions are represented at the input level and may influence whether the model recognizes equivalent phrasings as related.

Pre-training Distribution: English dominates most pre-training corpora, with other languages receiving varying degrees of representation. Models may have developed more robust question understanding for well-represented languages, potentially leading to higher coherence. Conversely, under-represented languages may exhibit more brittle behavior.

These considerations motivate a systematic cross-lingual study. Rather than assuming that English findings transfer, we directly measure coherence across languages to understand how linguistic and distributional factors interact with model behavior.

5.5.2 Experimental Setup

To test the hypotheses outlined above, we designed a multilingual evaluation spanning six typologically diverse languages, eleven models of varying scales, and metrics suitable for cross-lingual comparison.

Languages

We selected six languages to maximize typological diversity along the dimensions discussed above:

- **English**: analytic, high-resource, serves as baseline for comparison with previous sections
- **German**: fusional, rich morphology with grammatical case, medium-resource
- **Italian**: fusional, rich verbal morphology with grammatical gender, medium-resource
- **Chinese**: isolating, logographic script, tonal, high-resource
- **Japanese**: agglutinative, mixed script (kanji and kana), low-resource
- **Vietnamese**: analytic, minimal inflection, tonal, low-resource

This selection spans different morphological types (analytic, fusional, agglutinative), writing systems (alphabetic, logographic, syllabic), and levels of representation in typical pre-training corpora (high, medium, low). The inclusion of Vietnamese as a low-resource language is particularly relevant for testing whether coherence patterns depend on pre-training exposure.

Multilingual Cluster Construction

Starting from 1000 seed questions, 500 from PopQA-TP (Rabinovich et al., 2023) and 500 from SimpleQA (Haas et al., 2025), we construct multilingual question clusters by automatically generating semantically equivalent variations in each target language using a proprietary state-of-art multilingual translation language model. For each seed question, we generate 3 equivalent formulations per language, for a total of 18 multilingual variations per seed (6 languages $\times$ 3 variations). This results in a total of 18000 questions across all languages and seeds.

This procedure extends the monolingual clusters used in previous sections to a multilingual setting while preserving the equivalence relation. To ensure cluster quality, we performed two validation steps: (i) manual verification of English and Italian variations by native speakers, and (ii) back-translation validation for other languages, checking that back-translated questions preserved the original semantic content. Manual review of 200 randomly sampled questions confirmed near-perfect semantic preservation across translations.

We acknowledge that machine-generated variations may exhibit different characteristics compared to questions naturally formulated by native speakers. However, this approach enables controlled comparison across languages with identical semantic content, isolating the effect of language-specific surface variation from differences in question difficulty or domain.

Models.

The multilingual analysis uses a different set of models compared to Section 5.2.1. The English-only experiments used models representative of the state-of-the-art at the time (Mixtral, Llama2, Smaug, Phi-3), selected to demonstrate coherence gaps across different architectures. The multilingual analysis instead focuses also on understanding how coherence *scales* with model size. We therefore selected model families that offer multiple size variants, enabling within-family comparisons:

- **Phi-4**: 3.8B and 14B parameters (Abdin et al., 2024b)
- **Qwen3**: 4B, 14B, 80B, and 235B parameters (Yang et al., 2025)
- **DeepSeek-R1**: 32B parameters (DeepSeek-AI, 2025)
- **Apertus**: 8B and 70B parameters (Hernández-Cano et al., 2025)
- **GPT-OSS**: 20B and 120B parameters (Agarwal et al., 2025)

Metrics

We measure accuracy as the proportion of correctly answered questions, evaluated using an LLM-as-judge approach with Claude Sonnet 4.5 comparing generated answers against reference answers from the source datasets.

For semantic coherence, we use the formulation from Equation 2.4, but replace the monolingual encoder (all-mpnet-base-v2) with LaBSE (Feng et al., 2022), a language-agnostic sentence encoder trained on 109 languages. This substitution is essential: monolingual encoders would not provide comparable embeddings across languages, precluding meaningful cross-lingual coherence comparison.

Results

We evaluate all models on the multilingual question clusters, measuring both accuracy and semantic coherence for each language. This setup allows us to examine (i) whether coherence patterns observed in English generalize to other languages, (ii) how coherence varies across languages with different typological properties, and (iii) whether larger models within the same family exhibit higher coherence. Table 5.14 presents accuracy and semantic coherence for all evaluated models across languages.

Table 5.14: Accuracy (Acc) and Coherence (Coh) per model across languages. Results reveal significant variation in coherence across both models and languages.

	All		**EN**		**IT**		**DE**		**ZH**		**JA**		**VI**	
Model	Acc	Coh	Acc	Coh	Acc	Coh	Acc	Coh	Acc	Coh	Acc	Coh	Acc	Coh
Phi-4 3.8B	26.0	41.7	25.1	48.0	27.5	44.9	26.8	45.3	23.7	49.2	23.8	47.8	29.2	42.0
Phi-4 14B	27.3	47.3	27.8	53.8	28.5	52.6	30.5	52.9	23.2	51.6	24.0	53.5	29.7	52.2
Qwen3 4B	31.6	43.1	28.4	49.1	27.6	46.7	32.2	47.1	36.4	54.0	34.9	53.2	29.9	56.5
Qwen3 14B	29.4	44.0	27.1	48.4	26.8	47.4	27.0	46.7	34.8	45.2	30.8	45.7	29.7	50.9
Qwen3 80B	35.1	49.1	39.3	57.2	36.6	54.5	38.6	54.9	29.9	59.1	28.8	56.7	37.4	59.9
Qwen3 235B	41.3	46.1	47.0	55.0	42.2	51.9	45.6	51.1	36.0	53.1	32.9	51.8	44.2	55.1
DeepSeek-R1 32B	28.9	38.6	24.0	44.0	26.4	41.3	26.5	42.1	26.0	39.6	27.3	39.9	42.9	38.6
Apertus 8B	21.5	45.4	20.2	52.9	21.4	51.1	22.1	50.8	19.6	53.5	22.2	51.7	23.8	49.9
Apertus 70B	31.5	51.6	30.7	57.3	32.6	57.7	31.4	57.1	28.4	57.4	32.8	58.6	32.9	59.6
GPT-OSS 20B	29.2	41.3	27.3	44.5	28.7	42.3	27.8	44.0	33.2	41.2	31.1	39.6	27.3	44.4
GPT-OSS 120B	29.9	49.9	31.2	55.0	30.6	51.8	30.4	53.8	26.7	51.5	29.3	49.3	31.2	52.5

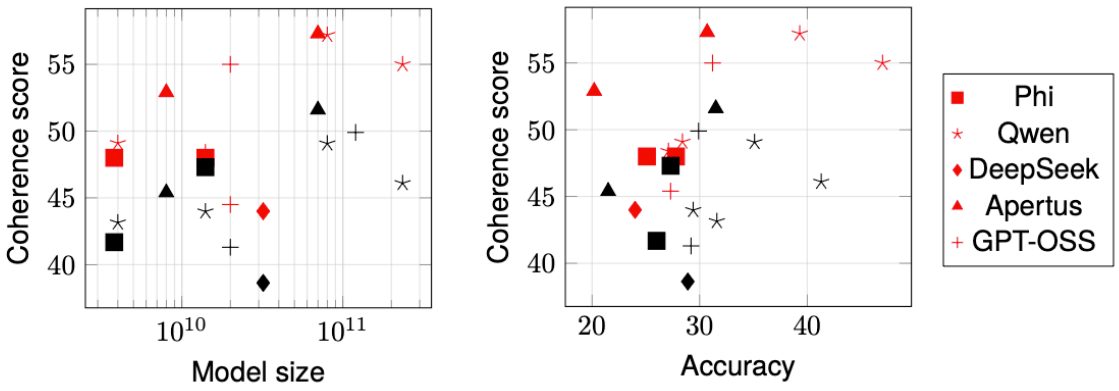

Figure 5.4: Coherence scaling by model size (left) and by accuracy (right). Black points represent all languages aggregated; red points represent English only. The positive correlations confirm that coherence patterns observed in English generalize across languages.

Key Findings

Coherence Correlates with Accuracy and Size (RQ5). Consistent with previous work (Section 5.2.2) on English, we observe positive correlations between coherence and both accuracy and model size in the multilingual setting. As shown in Figure 5.4, the Pearson correlation between coherence and accuracy is $\rho = 0.39$ across all languages, strengthening to $\rho = 0.64$ for English alone. The stronger English correlation aligns with the observation that English, as the dominant pre-training language, exhibits more predictable coherence-accuracy relationships. The correlation between coherence and model size is $\rho = 0.42$ across all languages and $\rho = 0.49$ for English.

However, these correlations reflect aggregate trends across different model families; they do not imply that scaling reliably improves coherence within a given family. A model can achieve high accuracy while remaining incoherent (correctly answering some phrasings but not others), and as the next paragraph demonstrates, a larger model is not guaranteed to be more coherent than a smaller one from the same family. This confirms that coherence captures a distinct dimension of model behavior that merits independent evaluation.

Size Does Not Guarantee Coherence. Even within the same model family, increasing size does not guarantee higher coherence. The Qwen3 family illustrates this pattern: coherence increases from 4B (43.1) to 80B (49.1), but then *decreases* with the 235B model (46.1). This non-monotonic relationship suggests that factors beyond scale, such as training methodology, data composition, and optimization choices, play crucial roles in determining model coherence.

Cross-Lingual Coherence Variability. Smaller models display substantial variation in coherence across languages. Qwen3 4B ranges from 46.7 (Italian) to 56.5 (Vietnamese), a difference of nearly 10 percentage points. Phi-4 3.8B shows similar variability, ranging from 42.0 (Vietnamese) to 49.2 (Chinese).

Importantly, scaling reduces this variability for some model families but not others:

- **Phi-4**: Cross-lingual variability decreases dramatically with scale. The standard deviation across languages drops from $\sigma = 2.4$ (3.8B) to $\sigma = 0.7$ (14B), and the max-min difference reduces from 7.2 to 2.2 points. This suggests that Phi-4's training procedure

Table 5.15: Number of clusters with 1 or 2 out of 3 correct answers (incoherent clusters) for Qwen3 models across languages. Lower values indicate higher coherence. See Figure 5.5 for the full distribution.

Model	EN	IT	DE	ZH	JA	VI
Qwen3 4B	389	412	398	367	371	385
Qwen3 14B	402	425	418	378	385	398
Qwen3 80B	358	389	375	348	352	368
Qwen3 235B	343	401	442	354	360	426

produces increasingly language-agnostic coherence as the model scales.

- **Qwen3**: Cross-lingual variability remains high even at large scales, with standard deviations of $\sigma = 3.7$ (4B), $\sigma = 1.9$ (14B), $\sigma = 2.0$ (80B), and $\sigma = 1.6$ (235B). The 80B model still shows a 5.4 point range across languages (54.5 to 59.9).

These patterns indicate that cross-lingual coherence consistency is not an automatic byproduct of scale but depends on model-specific training choices.

Language Typology Does Not Determine Coherence. We initially hypothesized that morphologically simpler languages might exhibit higher coherence by constraining the space of possible surface variations. Vietnamese, as an analytic language with minimal inflection, represented an ideal test case.

The results do not support this hypothesis. While Vietnamese exhibits high coherence for some models, such as Qwen3 80B: 59.9, and Apertus 70B: 59.6, Phi-4 3.8B shows its *lowest* coherence on Vietnamese (42.0). This inconsistency indicates that pre-training data distribution likely dominates over typological factors: a model with limited Vietnamese exposure cannot benefit from the language's morphological simplicity if it lacks robust representations for Vietnamese text in the first place.

5.5.3 Per-Cluster Analysis

To complement the continuous semantic coherence metric, we analyze coherence through a discrete lens analogous to the per-cluster analysis in Section 5.2.4. For each cluster of 3 semantically equivalent questions in a given language, we count how many receive correct answers (0, 1, 2, or 3). A perfectly coherent model would show mass only at 0 (consistently wrong) and 3 (consistently correct); mass at 1 or 2 indicates incoherence where the model possesses the knowledge, as evidenced by success on at least one phrasing, but it fails to access it consistently.

Figure 5.5 visualizes this distribution for the Qwen3 family across all languages, while Table 5.15 presents the count of incoherent clusters (1 or 2 correct) per language.

The results reveal an asymmetry across language groups. The 235B model achieves highest coherence (fewest incoherent clusters) on English (343), Chinese (354), and Japanese (360), but substantially lower coherence on Italian (401), German (442), and Vietnamese (426). This pattern suggests that the model's knowledge access mechanisms differ across language families, with stronger coherence for languages that likely received more emphasis during Qwen3's training process.

Interestingly, the discrete analysis shows Vietnamese with *lower* coherence than Chinese and Japanese for the largest model, despite Vietnamese showing *higher* semantic coherence

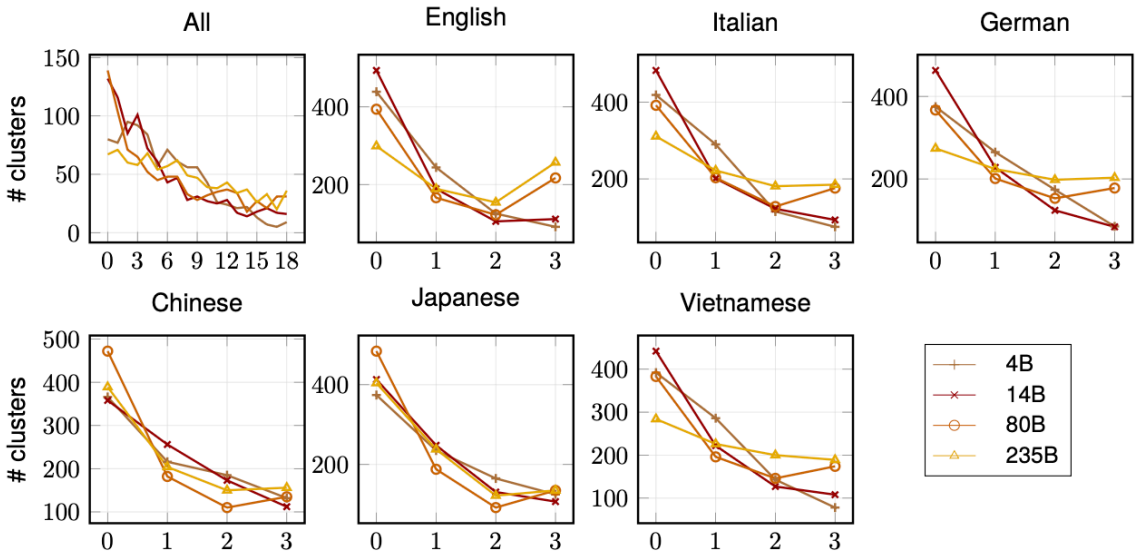

Figure 5.5: Number of clusters with 0, 1, 2, or 3 out of 3 correct answers for Qwen3 models across languages. The central region (1–2 correct) represents incoherent clusters where the model has the knowledge but fails to access it consistently. Larger models generally shift mass toward the extremes (0 or 3), indicating improved coherence.

in Table 5.14. This apparent contradiction reflects the different aspects captured by the two metrics: semantic coherence measures answer similarity regardless of correctness, while discrete coherence measures consistency in correctness judgments. A model can produce semantically similar but incorrect answers (high semantic coherence, low discrete coherence), which appears to be the case for Qwen3 235B on Vietnamese.

5.5.4 Practical Implications

The multilingual analysis yields two practical recommendations for deploying coherence-sensitive systems:

1. **Evaluation**: Coherence should be assessed across multiple languages rather than assuming English results transfer. As demonstrated, cross-lingual patterns can differ substantially even for the same model.

2. **Model Selection**: When cross-lingual coherence consistency is important, model family matters. Phi-4 achieves more uniform coherence across languages than Qwen3 (standard deviation $\sigma = 0.7$ for Phi-4 14B vs $\sigma \geq 1.6$ for Qwen3 235B), suggesting that training procedures affect cross-lingual generalization independently of overall accuracy.

3. **Training**: Model developers should consider coherence as a training objective, particularly for multilingual deployment where consistent behavior across languages is essential.

Whether the q-RAG approach introduced in Section 5.2.3 and the coherence-aware training methods from Section 5.4 would yield similar benefits in multilingual settings remains an open question. Both approaches require high-quality question clusters, which for multilingual settings would necessitate either large-scale question databases for each language

or cross-lingual retrieval models, resources that do not currently exist. The observed cross-lingual coherence variability suggests that benefits may differ across languages, but empirical validation is left for future work.

5.5.5 Summary

This section extended the coherence analysis to a multilingual setting, addressing RQ5. Coherence correlates positively with both accuracy ($\rho = 0.39$) and model size ($\rho = 0.42$) across languages, but cross-lingual consistency varies substantially by model family. These findings suggest that multilingual deployments require explicit coherence evaluation rather than extrapolation from English results.

The previous sections focused on coherence at the generation stage. However, in RAG pipelines, incoherence can arise upstream: if the retrieval component returns different documents for equivalent queries, the LLM receives different contexts, compounding the coherence problem even if the generator itself were coherent. The following section addresses this upstream problem.

5.6 Coherence in Document Retrieval

Dense retrieval models may return substantially different document rankings for queries that express the same information need. Consider a user asking about a medical condition: different phrasings might retrieve clinical guidelines versus patient forum posts, leading to different answers regardless of LLM coherence.

This sensitivity has important practical implications. Behavioral studies have shown that users frequently reformulate their queries when initial search results are unsatisfactory (Jansen et al., 2007, 2005), with estimates suggesting that up to 50% of search engine traffic consists of query reformulations (Wang et al., 2021). If a retrieval system returns different documents for equivalent phrasings of the same question, users may need multiple attempts to find relevant information, increasing both user frustration and computational cost.

The coherence problems in retrieval and generation are not independent as they compound each other in modern RAG pipelines. Consider a user asking a question that gets routed through a RAG system. If the retrieval component is incoherent, equivalent phrasings of the question will retrieve different documents. These different documents then form the context for the LLM, which generates answers grounded on different evidence. Even if the LLM itself were perfectly coherent, it would produce different answers because it receives different inputs. When combined with the LLM's own coherence gaps (Section 5.2.2), the effect is amplified: the system exhibits compounding incoherence where both retrieval and generation contribute to inconsistent behavior.

This observation motivates addressing coherence at both stages of the pipeline. Section 5.2.3 showed that q-RAG improves LLM coherence by providing redundant question signal at generation time. This section addresses the upstream problem: ensuring that the retrieval component itself produces consistent results regardless of how the user phrases their query.

Previous work has explored various approaches to reduce query sensitivity, including synthetic data augmentation (Chaudhary et al., 2024; Meng et al., 2024) and query reformulation (Ma et al., 2023; Shi et al., 2024). However, data augmentation alone shows inconsistent results across datasets, and query reformulation introduces additional latency and cost by

requiring an LLM to rewrite queries at inference time. We take a different approach: we use an LLM to generate clusters of equivalent queries for training, then introduce a Coherence Ranking (CR) loss that penalizes ranking inconsistencies across queries within the same cluster. In this way, the LLM's ability to recognize semantic equivalence is transferred to the retrieval model during training, without requiring the LLM at inference time.

5.6.1 Problem Definition

To formalize retrieval coherence, we first establish notation for dense retrieval and then define coherence as ranking consistency across equivalent queries.

Let $\delta : \mathcal{Q} \times \mathcal{D} \to [0, 1]$ be a dense retrieval model that computes similarity between a query q and a document d. For a query q and document collection $\mathcal{D}$, define the top-k retrieved documents as:

$$\tau_{\delta,\mathcal{D}}(q, k) = [d_1^q, d_2^q, \ldots, d_k^q] \tag{5.1}$$

where documents are ranked by decreasing similarity $\delta(q, d_i^q) \geq \delta(q, d_{i+1}^q)$.

Given a cluster $\mathcal{C} = \{q_1, \ldots, q_n\}$ of semantically equivalent queries, we define *retrieval coherence* as the average rank similarity between retrieved document lists:

$$\text{Coherence}_{\text{DR}}(\mathcal{C}, \delta) = \frac{2}{n(n-1)} \sum_{i<j} \text{RBO}(\tau_{\delta,\mathcal{D}}(q_i, k), \tau_{\delta,\mathcal{D}}(q_j, k)) \tag{5.2}$$

where RBO (Rank-Biased Overlap) (Webber et al., 2010), described in detail in Section 2.9, is particularly well suited for comparing ranked lists: it is top-weighted, differences in early positions have greater impact, accommodates lists of different lengths, and returns values in $[0, 1]$ with an intuitive interpretation.

5.6.2 Experimental Setup

We evaluate the proposed Coherence Ranking loss on standard information retrieval benchmarks, comparing against multiple baselines and analyzing both relevance and coherence metrics. This section addresses the research questions summarized in Table 5.16.

Table 5.16: Research questions for document retrieval coherence.

RQ	Research Question	Section
RQ7	Do retrieval models return consistent document rankings?	§5.6.3
RQ8	Can coherence-aware training improve coherence and relevance metrics?	§5.6.5
RQ9	In which retrieval scenarios is ranking coherence most critical?	§5.6.8
RQ10	Do coherence improvements transfer to out-of-domain benchmarks?	§5.6.9

Datasets

We train and evaluate our models on two widely adopted information retrieval benchmarks. The training data consist of triplets $\langle q, d^+, D^- \rangle$, where q denotes a query, d^+ a relevant passage, and D^- a set of hard negative passages. These negatives, documents that are topically related to the query but not actually relevant, are mined using the dense-retrieval–based procedure described in (Wang et al., 2021), which identifies challenging distractors and thus supports more effective contrastive training. Statistics of the datasets are reported in Table 5.17.

MS-MARCO v1: A large-scale passage retrieval benchmark with 8.8 million passages and 495K training queries. We use 5 hard negatives per query mined following the procedure of Wang et al. (2022). Since official test labels are not public, we split the development set into validation consisting of 3490 queries and test consisting of 3490 queries.

Natural Questions (NQ): Originally containing 132K queries paired with Wikipedia pages, we successfully extracted hard negatives for 120K queries using the same mining procedure as for MS-MARCO v1, generating 10 hard negatives per query. We randomly selected 3000 queries for development. The test set consists of 3452 queries over 2.68 million passages.

Table 5.17: Dataset statistics for retrieval experiments.

	MS-MARCO v1	**NQ**
Index Passages	8,841,823	2,681,468
Training Queries	495,260	119,554
Development Queries	3,490	3,000
Test Queries	3,490	3,452
Hard Negatives per Query	5	10
Generated Queries per Original	10	10

Query Cluster Generation

To create question clusters for training and evaluation, we generate query variations using Phi-3-mini (3.8B parameters). Note that here generated queries serve as data augmentation to teach the model invariance to phrasing, rather than as semantic input at inference time as in Section 5.3.2.

The generation prompt, visible in Appendix C.1, instructs the model to produce 10 semantically equivalent reformulations of each query. To encourage lexical diversity, we provide style templates from different QA datasets: QuAD-style (introductory phrase focusing on specific information), MS-MARCO-style (open-ended information requests), DuoRC-style (using "if" to set up context), HotpotQA-style (combining indicators with follow-up questions), NQ-style (concise and direct), TriviaQA-style (open-ended, sometimes requiring nuanced answers), and WebQA-style (requests for lists or sets of information). The model outputs a structured JSON with each reformulation tagged by style. We validated the prompt by manually evaluating 100 random queries and their 10 generated variations, finding 100% equivalence accuracy.

Table 5.18 shows examples of generated query clusters.

Models

To verify that our approach generalizes across architectures, we experiment with three encoder models of varying size and design:

- **MPNet** (110M parameters): Strong baseline on IR benchmarks. We use the multi-qa-mpnet-base-cos-v1 checkpoint, pre-trained on various supervised and self-supervised sentence similarity tasks.
- **MiniLM-v2-L12** (33M parameters): Efficient model for resource-constrained deployments using all-MiniLM-L12-v2.

Table 5.18: Examples of generated query clusters. Each block shows the original query from a given dataset and representative reformulations in different styles.

MS-MARCO v1
Original: What is the typical function of simple epithelium?
QuAD-style: What specific function does simple epithelium serve? *MS-MARCO-style:* I need to know what simple epithelium typically does. Can you tell me its main function? *DuoRC-style:* If we look at simple epithelium in the body, what is its main function? *HotpotQA-style:* Considering tissue types and their roles, what is the main function of simple epithelium, and why is it important? *NQ-style:* Could you explain the main function of simple epithelium? *TriviaQA-style:* What role does simple epithelium play in the body? *WebQA-style:* simple epithelium purpose
NQ
Original: what are signs of a mouse dying
QuAD-style: What are the key signs that indicate when a mouse is dying? *MS-MARCO-style:* I need to know the most common signs that a mouse is dying. Can you provide me with that information? *DuoRC-style:* If a mouse is close to dying, what signs would you expect to see? *HotpotQA-style:* Considering typical symptoms and behavioral changes, what are the common signs that a mouse is dying? *NQ-style:* What are the typical symptoms that indicate a mouse is dying? *TriviaQA-style:* Identify the most common signs that a mouse is approaching the end of its life. *WebQA-style:* signs a mouse is dying

- **ModernBERT-base** (133M parameters): Recent architecture designed for long sequences. Since ModernBERT was trained only with MLM objective, we continuously pre-trained the checkpoint on 1.5B text-similarity pairs before fine-tuning on retrieval tasks using the same pretraining of Chapter 3.

Training Configurations

We evaluate several training strategies to isolate the contribution of each component:

- **Public checkpoint:** the base model without any fine-tuning, pre-trained on various supervised and self-supervised sentence similarity tasks.

- **Fine-Tuning (FT):** the public checkpoint fine-tuned on target training data (MS-MARCO v1 or NQ) using standard Multiple Negative Ranking (MNR) loss on triplets $\langle q, d^+, D^- \rangle$.

- **+ Query Augmentation:** training data is expanded with generated query variations. For each triplet $\langle q, d^+, D^- \rangle$, we add 10 extra examples $\{\langle q_i, d^+, D^- \rangle\}_{i=1}^{10}$, where each q_i is an equivalent query generated from q.

- **+ $\mathcal{L}_{\mathbf{QQ}}$:** multi-task learning that alternates between MNR (query-document) and query similarity objectives in round-robin fashion. On each iteration: (i) one optimization step with MNR loss; (ii) one step optimizing query similarity on pairs $\langle q_i, q_j \rangle$ where $q_i \equiv q_j$.

- **+ $\mathcal{L}_{\text{CR}}$:** our proposed Coherence Ranking loss (Section 5.6.4), which jointly optimizes MNR and query-similarity through a unified objective.

- **Full:** combines $\mathcal{L}_{\text{CR}}$ with query augmentation.

We also compare against two lexical baselines: BM25, using the Pyserini corpus with query expansion, and SPLADE++ (Formal et al., 2022), a learned sparse retrieval method.

Training uses learning rate grid search over $\{5 \times 10^{-7}$ to $3 \times 10^{-5}\}$, batch sizes from 32 to 1024, AdamW optimizer with 10% warmup, up to 15 epochs with early stopping (patience 5), and 8× NVIDIA H100 GPUs.

5.6.3 Baseline Coherence Analysis

Before introducing our proposed loss, we first quantify the coherence of existing retrieval methods to establish the severity of the problem. For each test query q, we use its cluster of 10 generated equivalent queries $C = \{q_1, \ldots, q_{10}\}$. We run the retrieval model on both the original query q and all generated queries in C, obtaining top-k ranked lists. We then measure the similarity between the ranking produced by q and those produced by each $q_i \in C$, using RBO and Spearman correlation. Details on both metrics are available in Section 2.9.1. We focus on top-5 results (RBO@5, Spearman@5) and average across all test queries. Higher values indicate more coherent rankings: *the model returns similar documents regardless of how the query is phrased*. For this baseline analysis, we use MPNet as the representative dense retrieval model; generalization to other architectures is evaluated in Section 5.6.7.

Table 5.19 shows baseline coherence of different retrieval approaches.

Table 5.19: Baseline coherence of retrieval methods. RBO@5 and Spearman@5 measure ranking overlap between original and generated queries (higher = more coherent). Results averaged over 5 runs.

	MS-MARCO		**NQ**	
Model	RBO@5	Spearman@5	RBO@5	Spearman@5
BM25	$0.22_{\pm.24}$	$0.45_{\pm.11}$	$0.40_{\pm.27}$	$0.49_{\pm.15}$
SPLADE++	$0.46_{\pm.28}$	$0.49_{\pm.15}$	$0.65_{\pm.23}$	$0.54_{\pm.18}$
MPNet (public)	$0.42_{\pm.25}$	$0.46_{\pm.12}$	$0.57_{\pm.22}$	$0.49_{\pm.15}$
MPNet (fine-tuned)	$0.46_{\pm.26}$	$0.47_{\pm.13}$	$0.54_{\pm.23}$	$0.49_{\pm.16}$

Key Findings

All Retrieval Methods Exhibit Coherence Gaps (RQ7). Lexical methods are highly sensitive to query phrasing: BM25 produces RBO@5 of only 0.22 on MS-MARCO, meaning top-5 documents differ substantially across query phrasings. SPLADE++, thanks to query expansion, achieves better coherence (0.46 on MS-MARCO). Dense models improve but remain incoherent: fine-tuned MPNet achieves 0.46 RBO@5 on MS-MARCO (from 0.42 for the public checkpoint), but this is still far from perfect coherence. Interestingly, on NQ fine-tuning actually *decreases* coherence from 0.57 to 0.54, suggesting that standard MNR training may overfit to surface patterns rather than learning robust semantic matching.

5.6.4 Coherence Ranking Loss

The baseline analysis reveals that standard fine-tuning does not address, and can even worsen, the coherence problem. In light of this, we propose a Coherence Ranking (CR) loss that explicitly optimizes for ranking consistency across equivalent queries. Rather than relying on data augmentation alone, which showed inconsistent results, the CR loss directly penalizes ranking discrepancies within query clusters. The loss extends the standard Multiple Negative Ranking (MNR) loss with two additional components.

Loss Components

As mentioned, the CR loss builds upon standard MNR contrastive training. We first review the base objective, then introduce our two additional terms.

Base: Multiple Negative Ranking (MNR): Standard dense retrieval training uses contrastive loss over triplets $\langle q, d^+, D^- \rangle$:

$$\mathcal{L}_{\text{MNR}}(q, d^+, D^-) = -\log \frac{\exp(s(q, d^+)/\tau)}{\exp(s(q, d^+)/\tau) + \sum_{d \in D^-} \exp(s(q, d)/\tau)} \tag{5.3}$$

where $s(q, d) = \cos(q, d)$ is the cosine similarity between query and document embeddings, and τ is a temperature parameter. This loss encourages the model to rank the positive document d^+ above negatives D^-, but does not explicitly encourage consistent rankings across equivalent queries.

Extension 1: Query Embedding Alignment (QEA): Given a cluster of equivalent queries $C = \{q, q_1, \ldots, q_m\}$, this term penalizes differences between their embeddings, encouraging the model to map equivalent queries to similar representations:

$$\mathcal{L}_{\text{QEA}}(q, C) = \frac{1}{|C|} \sum_{q_i \in C} \|\mathbf{q} - \mathbf{q}_i\|_2^2 \tag{5.4}$$

Intuitively, if equivalent queries have similar embeddings, they will retrieve similar documents.

Extension 2: Similarity Margin Consistency (SMC): While QEA operates in embedding space, SMC acts directly on retrieval scores. It enforces that equivalent queries maintain the same similarity margins between positive and negative documents:

$$\mathcal{L}_{\text{SMC}}(q, C, d^+, D^-) = \sum_{q_i \in C} \sum_{d \in D^-} \big(m(q, d^+, d) - m(q_i, d^+, d)\big)^2 \tag{5.5}$$

where the margin $m(q, d^+, d) = s(q, d^+) - s(q, d)$ measures how much the model prefers d^+ over d. This ensures that equivalent queries not only retrieve similar documents, but rank them similarly.

The full Coherence Ranking loss combines MNR with the two coherence terms:

$$\mathcal{L}_{\text{CR}} = \mathcal{L}_{\text{MNR}} + \lambda_1 \mathcal{L}_{\text{QEA}} + \lambda_2 \mathcal{L}_{\text{SMC}} \tag{5.6}$$

where λ_1 and λ_2 control the weight of each coherence component, selected via grid search over $\{0, 0.2, 0.5, 0.8, 1.0\}$. The ablation study in Section 5.6.6 shows that both terms are necessary: using either alone yields suboptimal results.

Figure 5.6 provides a visual overview of the CR loss architecture.

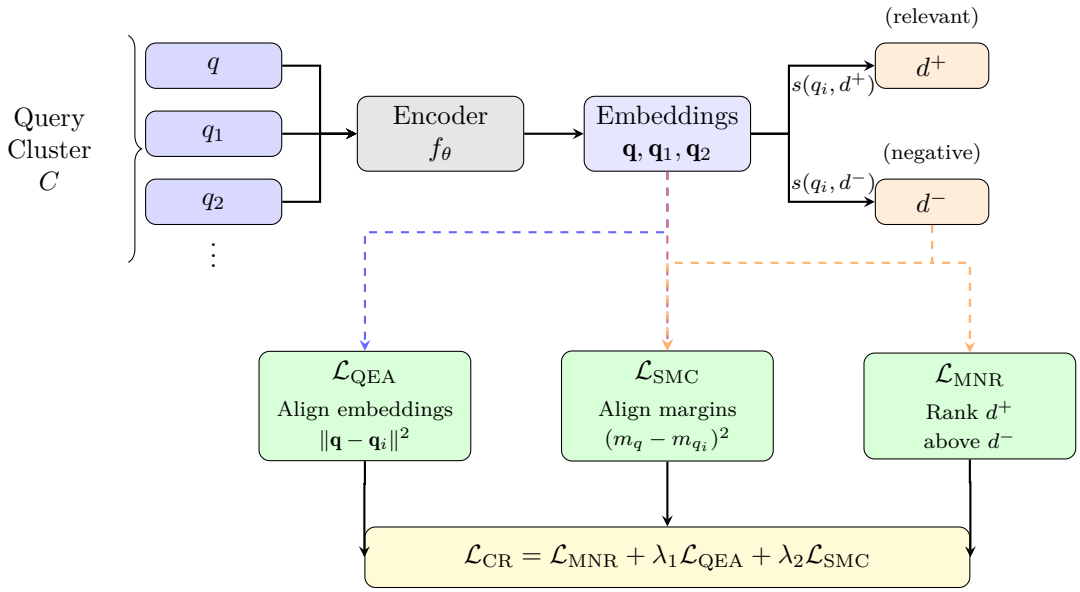

Figure 5.6: Overview of the Coherence Ranking (CR) Loss. A cluster of equivalent queries $C = \{q, q_1, q_2, \ldots\}$ is encoded into embeddings. The loss combines three components: $\mathcal{L}_{\text{QEA}}$ aligns the embeddings of equivalent queries in the representation space; $\mathcal{L}_{\text{SMC}}$ ensures that equivalent queries produce consistent similarity margins with respect to documents; $\mathcal{L}_{\text{MNR}}$ ensures that relevant documents rank above negatives. Both coherence terms are necessary: using either alone degrades performance (Section 5.6.6).

5.6.5 Main Results

Having defined the CR loss, we evaluate its effectiveness on MS-MARCO v1 and Natural Questions. A natural concern is whether optimizing for coherence might hurt relevance, the results show this is not the case. Table 5.20 presents the full comparison.

Key Findings

CR Loss Improves Both Coherence and Relevance (RQ8). On MS-MARCO, CR loss achieves 0.60 RBO@5 (+30% relative over FT baseline) while also improving relevance: +0.47 NDCG@10 and +0.19 P@1. On NQ, improvements are even larger: RBO@5 increases from 0.54 to 0.70 (+30% relative), alongside +1.69 NDCG@10 and +1.33 P@1. This confirms that coherence and relevance are complementary objectives.

Query Augmentation Trades Off Relevance and Coherence. Simple data augmentation with generated queries improves coherence, 0.46 to 0.59 RBO on MS-MARCO, but hurts relevance, from 41.51 to 40.05 NDCG, and from 22.82 to 21.85 P@1). CR loss achieves better coherence *and* better relevance, demonstrating that the loss formulation is superior to naive augmentation.

Combining CR with Augmentation (Full) Maximizes Coherence. The Full configuration achieves the highest coherence, 0.63 on MS-MARCO v1 and 0.71 RBO on NQ, but slightly lower relevance than CR alone. For applications prioritizing consistency, Full is preferred.

Lexical Baselines Confirm Dense Retrieval Advantages. BM25 shows poor coherence, 0.22 RBO and relevance 16.74 P@1 on MS-MARCO v1. SPLADE++, which learns sparse representations, achieves coherence comparable to dense baselines, 0.46 RB, with competitive

Table 5.20: Document retrieval results with MPNet on MS-MARCO v1 and Natural Questions. Best results in bold; results averaged over 5 runs with standard deviation.

		Relevance			Coherence	
Configuration	**P@1**	**NDCG@10**	**MRR@10**	**MAP@100**	**RBO@5**	**Spear.@5**
			MS-MARCO v1			
Public ckpt	21.58	39.88	33.79	34.27	0.42±.25	0.46±.12
FT	22.82±.11	41.51±.08	35.34±.12	35.68±.11	0.46±.26	0.47±.13
+ Q.Augm.	21.85±.12	40.05±.21	33.96±.41	34.31±.21	0.59±.27	0.54±.17
+ $\mathcal{L}_{\text{QQ}}$	22.87±.21	41.31±.10	35.10±.08	35.50±.10	0.51±.27	0.49±.15
+ $\mathcal{L}_{\text{CR}}$	**23.01**±.10	**41.98**±.17	**35.73**±.16	**35.70**±.13	0.60±.26	0.53±.17
Full	22.46±.10	41.43±.18	34.71±.18	35.18±.15	**0.63**±.26	**0.55**±.18
BM25	16.74	33.19	27.13	27.85	0.22±.24	0.45±.11
SPLADE++	21.74	40.08	33.72	34.35	0.46±.28	0.49±.15
			Natural Questions			
Public ckpt	30.71	46.53	42.59	40.79	0.57±.22	0.49±.15
FT	38.16±.17	52.16±.13	49.50±.17	47.50±.18	0.54±.23	0.49±.16
+ Q.Augm.	38.57±.11	53.00±.01	49.89±.08	47.66±.16	0.66±.23	0.54±.19
+ $\mathcal{L}_{\text{QQ}}$	38.84±.04	53.42±.07	50.23±.08	48.25±.10	0.59±.23	0.51±.17
+ $\mathcal{L}_{\text{CR}}$	**39.49**±.11	**53.85**±.08	**50.65**±.09	**48.56**±.04	0.70±.22	0.55±.19
Full	39.36±.11	53.73±.07	50.50±.10	48.29±.08	**0.71**±.21	**0.57**±.20
BM25	16.48	30.55	26.34	25.86	0.40±.27	0.49±.15
SPLADE++	29.66	44.89	41.11	39.43	0.65±.23	0.54±.18

Table 5.21: Ablation study on CR loss components. MNR is always included. Results on MS-MARCO with MPNet.

Configuration	**P@1**	**NDCG@10**	**RBO@5**
$\mathcal{L}_{\text{MNR}}$ only (FT baseline)	22.82±.11	41.51±.08	0.46±.26
+ $\mathcal{L}_{\text{QEA}}$ only	22.78±.12	41.26±.14	0.20±.16
+ $\mathcal{L}_{\text{SMC}}$ only	22.81±.12	41.51±.14	0.22±.18
+ $\mathcal{L}_{\text{CR}}$ (both)	**23.01**±.10	**41.98**±.17	**0.60**±.26

relevance. However, CR-trained dense models substantially outperform both lexical methods on both metrics.

5.6.6 Ablation Study on Loss Components

We investigate whether both loss components, QEA and SMC, are necessary, or whether one alone would suffice. Table 5.21 isolates the contribution of each component on MS-MARCO v1.

The results reveal a surprising pattern: using either component alone actually *decreases* coherence compared to baseline, with RBO dropping from 0.46 to 0.20–0.22. Only when combined do QEA and SMC achieve the desired improvement (0.60 RBO). We hypothesize the following dynamics. QEA alone pushes embeddings of equivalent queries closer together, but without any constraint on document relationships, this may cause the model to map queries too aggressively into a narrow region of embedding space, losing fine-grained distinctions that matter for ranking. On the other hand, SMC alone penalizes margin inconsistencies, but since query embeddings remain spread out, there is no QEA regularization, the optimization

signal is noisy and may lead to unstable training dynamics. When combined, QEA provides a stable foundation by aligning equivalent queries in embedding space, while SMC ensures this alignment translates into consistent similarity scores with documents. The two components address complementary aspects of the coherence problem: QEA operates at the query representation level, SMC at the query-document interaction level.

5.6.7 Generalization Across Models

All experiments so far have used MPNet as the encoder model. To verify that CR loss generalizes beyond a single architecture, we evaluate it on two additional models: MiniLM-v2-12L (33M parameters), a compact model designed for efficient dense retrieval, and ModernBERT-base (133M parameters), a recent model designed for long sequences. Table 5.22 presents the results for all configurations, including lexical baselines for reference.

Table 5.22: CR loss generalization across MiniLM-v2-12L and ModernBERT-base models. Results on MS-MARCO v1 and NQ. For reference, BM25 and SPLADE++ have also been reported. Best results in bold; results averaged over 5 runs with standard deviation.

	MiniLM-v2-12L			**ModernBERT-base**		
Configuration	**P@1**	**NDCG@10**	**RBO@5**	**P@1**	**NDCG@10**	**RBO@5**
			MS-MARCO v1			
Public ckpt	21.61±0.12	39.12±0.10	0.39±0.24	15.02±0.13	31.03±0.12	0.40±0.25
FT	22.58±0.11	40.47±0.09	0.44±0.25	22.82±0.12	41.63±0.10	0.39±0.24
+ Q.Augm.	22.72±0.14	40.38±0.12	0.55±0.26	21.68±0.14	39.92±0.13	0.56±0.26
$\mathcal{L}_{\text{QQ}}$	22.83±0.13	40.52±0.11	0.57±0.26	21.91±0.13	40.58±0.11	0.49±0.25
$\mathcal{L}_{\text{CR}}$	**23.32**±0.10	**41.11**±0.10	**0.57**±0.27	**23.01**±0.11	**41.91**±0.10	**0.56**±0.26
BM25	16.73±0.01	33.18±0.01	0.22±0.23	16.48±0.01	30.52±0.02	0.40±0.24
SPLADE++	21.73±0.08	40.09±0.07	0.46±0.27	29.69±0.10	44.91±0.09	0.65±0.24
			Natural Questions			
Public ckpt	26.29±0.13	41.38±0.10	0.53±0.23	21.81±0.14	37.58±0.12	0.58±0.23
FT	34.81±0.12	48.29±0.09	0.46±0.22	36.57±0.12	50.43±0.10	0.15±0.19
+ Q.Augm.	35.38±0.13	48.12±0.10	0.61±0.25	35.91±0.14	50.18±0.11	0.61±0.23
$\mathcal{L}_{\text{QQ}}$	35.42±0.11	48.68±0.09	0.44±0.23	36.82±0.12	50.97±0.10	0.38±0.23
$\mathcal{L}_{\text{CR}}$	**36.11**±0.10	**49.21**±0.09	**0.65**±0.22	**37.18**±0.11	**51.08**±0.10	**0.65**±0.22
BM25	16.48±0.01	30.61±0.02	0.40±0.27	16.19±0.01	30.33±0.02	0.39±0.26
SPLADE++	29.68±0.10	44.92±0.08	0.65±0.23	39.41±0.10	56.31±0.09	0.67±0.23

Key Findings

CR Loss Generalizes Across Architectures (RQ8). Both MiniLM and ModernBERT show consistent improvements with CR loss, confirming that the approach is not specific to MPNet. On MS-MARCO v1, CR loss achieves the best relevance metrics for both models, 23.32 P@1 for MiniLM, 23.01 for ModernBERT, while maintaining strong coherence of 0.57 and 0.56 RBO respectively. The pattern observed with MPNet holds: query augmentation and $\mathcal{L}_{\text{QQ}}$ improve coherence but often hurt relevance, while CR loss improves both simultaneously.

CR Loss Recovers Coherence Lost During Fine-tuning. The results on NQ reveal an important phenomenon. ModernBERT's coherence after fine-tuning drops dramatically, from 0.58 RBO Public checkpoint to only 0.15 RBO, suggesting that standard MNR training causes

the model to overfit to surface patterns in ways that severely harm coherence. CR loss recovers this lost coherence, improving RBO from 0.15 to 0.65, while also achieving the best relevance: 37.18 P@1 and 51.08 NDCG@10. MiniLM shows a similar pattern: fine-tuning decreases coherence from 0.53 to 0.46, and CR loss restores it to 0.65.

CR Loss Outperforms Lexical Baselines. CR-trained dense models outperform BM25 and SPLADE++ on both relevance and coherence across all configurations. This confirms that the coherence improvements from CR loss do not come at the cost of competitiveness with strong lexical baselines, but rather enhance the overall retrieval quality.

CR Loss Acts as Architecture-Independent Regularizer. The results suggest that CR loss is particularly valuable when standard training degrades coherence, acting as a regularizer that prevents the model from learning spurious correlations between query surface forms and document rankings. This effect is consistent across models of different sizes and architectural choices.

5.6.8 Retrieval Complexity Analysis

The previous experiments evaluated coherence on the full test set. However, we hypothesize that coherence is particularly critical in scenarios where the retrieval task itself is ambiguous, i.e., when multiple documents have similar relevance scores and the model must make fine-grained distinctions. In such cases, small perturbations in the query embedding, caused by different query wording, can lead to large changes in the final ranking, since documents near the decision boundary may swap positions.

To test this hypothesis, we identify *"complex"* queries as those where the difference in retrieval scores between the top-1 and 50th ranked document is less than 0.1. This criterion captures queries where the model assigns nearly identical scores to many candidates, making the ranking highly sensitive to input variations. The goal of this analysis is to determine whether CR loss provides greater benefits precisely in these challenging scenarios.

For instance, in MS-MARCO v1, queries like *"What constitutional amendment granted American women suffrage?"* exhibit this pattern: many documents discuss constitutional amendments or women's rights, receiving similar scores. In NQ, *"where does the great outdoors movie take place"* similarly retrieves multiple location-related passages with near-identical relevance. Table 5.23 presents the results of this analysis.

Table 5.23: RBO@5 (coherence) on "complex" queries where the retrieval score difference between top-1 and top-50 documents is less than 0.1. These queries are particularly sensitive to input variations.

Configuration	MS-MARCO	NQ
Public ckpt	$0.16_{\pm 0.14}$	$0.41_{\pm 0.21}$
FT	$0.17_{\pm 0.14}$	$0.25_{\pm 0.17}$
+ Q.Augm.	$0.32_{\pm 0.23}$	$0.43_{\pm 0.24}$
+ $\mathcal{L}_{\text{QQ}}$	$0.24_{\pm 0.18}$	$0.30_{\pm 0.20}$
+ $\mathcal{L}_{\text{CR}}$	$0.34_{\pm 0.24}$	$0.49_{\pm 0.23}$
Full	$\mathbf{0.38}_{\pm 0.25}$	$\mathbf{0.52}_{\pm 0.24}$
BM25	$0.07_{\pm 0.14}$	$0.36_{\pm 0.27}$
SPLADE++	$0.23_{\pm 0.21}$	$0.48_{\pm 0.26}$

Key Findings

CR Loss Dramatically Improves Coherence on Complex Queries (RQ9). Coherence on complex queries is substantially lower than on the full test set (compare with Table 5.20): on MS-MARCO v1, FT achieves only 0.17 RBO versus 0.46 on the full set; on NQ, the drop is even more dramatic, from 0.54 to 0.25. This confirms our hypothesis that when many documents have similar relevance scores, small input variations cause large ranking changes. CR loss shows its greatest benefits precisely in these challenging cases: on MS-MARCO v1, coherence improves from 0.16 (public checkpoint) to 0.38 (Full), a +138% relative improvement; on NQ, from 0.25 (FT) to 0.52 (Full), more than doubling coherence. This suggests that CR loss is particularly valuable when document relevance scores are closely clustered. Lexical methods struggle even more: BM25 achieves only 0.07 RBO on MS-MARCO, highlighting that token-based matching is especially brittle when fine-grained semantic distinctions are required. SPLADE++ performs better (0.23 on MS-MARCO, 0.48 on NQ) but still below CR-trained dense models.

5.6.9 Transfer Evaluation: BEIR and TREC-DL

A key question for any training method is whether improvements on the training domain transfer to unseen domains. This is particularly important for CR loss, since the coherence signal comes from generated query variations that may reflect patterns specific to MS-MARCO. To assess generalization, we evaluate MS-MARCO-trained MPNet models on two standard transfer benchmarks: BEIR, a diverse collection of 11 retrieval datasets spanning scientific, legal, and general domains, and TREC-DL 2019/2020, which uses the same MS-MARCO corpus but with different, more comprehensive relevance judgments.

Tables 5.24 and 5.25 present zero-shot transfer results on BEIR. Since models are trained only on MS-MARCO, we focus on average performance across datasets rather than individual benchmarks, which would require domain-specific fine-tuning.

Table 5.24: NDCG@10 on BEIR benchmark (zero-shot transfer from MS-MARCO). Part 1/2.

Configuration	Scifact	SciDocs	FiQA	NFCorpus	Touche	DBPedia
Public	59.98	51.01	32.11	16.55	47.42	35.31
FT	59.08	47.72	31.94	23.89	46.89	36.72
+ Q.Augm.	59.56	46.86	32.26	23.82	44.42	35.86
+ $\mathcal{L}_{\text{QQ}}$	59.95	46.74	31.93	24.52	46.38	36.71
+ $\mathcal{L}_{\text{CR}}$	**60.46**	**48.50**	**32.77**	23.54	46.16	36.62

Table 5.25: NDCG@10 on BEIR benchmark (zero-shot transfer from MS-MARCO). Part 2/2.

Configuration	Climate	Quora	FEVER	NQ	HotpotQA	Avg
Public	25.35	87.69	17.39	58.77	47.57	43.56
FT	25.16	88.15	16.88	60.60	52.04	44.46
+ Q.Augm.	26.51	86.81	16.65	44.07	55.22	42.96
+ $\mathcal{L}_{\text{QQ}}$	25.39	88.91	16.88	44.25	54.48	44.01
+ $\mathcal{L}_{\text{CR}}$	**26.04**	88.61	**17.24**	60.26	**54.19**	**44.94**

Table 5.26: Results on TREC-DL benchmarks (zero-shot transfer from MS-MARCO training).

Configuration	TREC-DL '19		TREC-DL '20	
	NDCG@10	RBO@5	NDCG@10	RBO@5
Public ckpt	64.35	0.14±0.12	63.36	0.16±0.13
FT	69.77	0.15±0.13	65.52	0.16±0.14
+ Q.Augm.	70.46	0.16±0.15	65.49	0.16±0.15
+ $\mathcal{L}_{\text{QQ}}$	69.39	0.16±0.14	65.61	0.16±0.14
+ $\mathcal{L}_{\text{CR}}$	**71.14**	**0.19**±0.14	**65.82**	**0.19**±0.12

Key Findings

CR Loss Improves Transfer and Does Not Harm Generalization (RQ10). On BEIR, CR loss achieves the best average performance (44.94 NDCG@10) compared to standard fine-tuning (44.46) and the public checkpoint (43.56), with notable improvements on Scifact (+1.38), HotpotQA (+2.15), and Climate-FEVER (+0.88). On TREC-DL benchmarks, CR loss achieves the best NDCG@10 on both years: 71.14 on TREC-DL '19 (+1.37 over FT) and 65.82 on TREC-DL '20 (+0.30 over FT), with coherence also improving (RBO@5 from 0.15 to 0.19). These results confirm that the coherence signal from CR loss does not cause overfitting to MS-MARCO-specific patterns; instead, the regularization effect extends to out-of-domain transfer.

5.6.10 Impact on Downstream Applications

Retrieval models are rarely used in isolation as they typically serve as components in larger pipelines such as retrieve-and-rerank or RAG systems. We therefore evaluate whether the coherence improvements observed at the retrieval level propagate to these downstream applications.

Retrieve-and-Rerank Pipelines

In production systems, dense retrieval is often followed by a cross-encoder reranker (Nogueira and Cho, 2020) that rescores the top-k retrieved documents. While rerankers can correct some retrieval errors, they can only select from the documents provided by the first-stage retriever. If an incoherent retriever fails to include the best document in its top-k for certain query phrasings, the reranker cannot recover this document.

To quantify this effect, we define *reranking opportunity*: the probability that the document selected by the reranker (from top-50) appears in the top-50 for all query variations. Formally, let $d^* \in \tau_{\delta,\mathcal{D}}(q, 50)$ be the document selected by the reranker for query q from cluster C. The reranking opportunity is:

$$\text{opportunity}(q) = \frac{1}{|C|} \sum_{q_i \in C} \mathbf{1}_{\tau_{\delta,\mathcal{D}}(q_i,50)}(d^*) \tag{5.7}$$

The higher the opportunity, the lower the risk of dropping the highest-reranked document due to query sensitivity. We use BGE-reranker-large (Xiao et al., 2023b), a state-of-the-art cross-encoder reranker, for this evaluation.

Key Findings

CR Loss Improves Reranking Opportunity. Table 5.27 shows that CR loss substantially improves reranking opportunity across all models and datasets. On MS-MARCO, CR-trained

Table 5.27: Reranking opportunity (%) with BGE-reranker-large. Higher values indicate more consistent retrieval of the best-reranked document across query variations.

	MS-MARCO			NQ		
Configuration	**MPNet**	**MiniLM**	**ModernBERT**	**MPNet**	**MiniLM**	**ModernBERT**
Public ckpt	75.7	73.4	74.9	59.5	31.9	59.3
FT	79.7	78.4	75.8	58.9	52.6	11.8
+ Q.Augm.	85.9	83.7	84.5	67.5	63.6	59.4
+ $\mathcal{L}_{\text{QQ}}$	82.4	80.9	83.0	55.4	66.0	23.7
+ $\mathcal{L}_{\text{CR}}$	**87.0**	**85.7**	**85.5**	**70.9**	**70.4**	**65.8**
BM25		59.4			63.2	
SPLADE++		77.7			67.5	

Table 5.28: RAG accuracy (%) on KILT benchmarks using Mistral-7B-Instruct with top-5 retrieved documents.

Retriever Configuration	**Accuracy**
MPNet (FT)	60.0
MPNet (+ $\mathcal{L}_{\text{CR}}$)	**60.4**

models achieve 85.5–87.0% opportunity compared to 75.8–79.7% for standard fine-tuning. On NQ, the improvements are even more dramatic: ModernBERT's opportunity increases from a catastrophic 11.8% (FT) to 65.8% (+54 absolute), while MPNet improves from 58.9% to 70.9% (+12 absolute). Across all configurations, CR loss consistently achieves the highest reranking opportunity.

Practical Implications. These results have direct practical implications. With standard fine-tuning on NQ, ModernBERT would fail to include the best-reranked document for 88% of query variations as the reranker would almost never see the optimal document. CR loss reduces this failure rate to 34%. For MPNet on MS-MARCO v1, CR loss reduces the failure rate from approximately 20% to 13%, ensuring the reranker has more consistent access to high-quality candidates regardless of query phrasing.

RAG Pipelines

RAG systems combine retrieval with language model generation, using retrieved documents as context for answer generation. In these pipelines, retrieval coherence affects not only which documents the LLM sees, but also the consistency of generated answers across equivalent query phrasings.

To evaluate this, we simulate a RAG pipeline using Mistral-7B-Instruct-v0.2 as the generator, with top-5 documents from our retrieval models providing context. We evaluate on KILT benchmarks (Petroni et al., 2021), which include knowledge-intensive tasks such as fact verification and open-domain question answering.

Key Findings

Coherence Benefits Propagate to RAG. As shown in Table 5.28, CR-trained retrieval improves RAG accuracy by +0.4% absolute, from 60.0% to 60.4%). While this improvement may appear modest, it is notable for several reasons. First, this is a zero-shot evaluation where the retriever was not optimized for KILT tasks. Second, the LLM can partially compensate

Table 5.29: Comparison with query reformulation approaches on TREC-DL benchmarks. Reformulation methods underperform even the baseline without reformulation. Results are computed using the MPNet model.

	TREC-DL '19		TREC-DL '20	
Configuration	P@1	NDCG@10	P@1	NDCG@10
No reformulation (FT)	**83.72**	69.77	81.48	65.52
Centroid	76.74	67.16	78.22	61.68
Best	82.80	65.02	80.95	65.47
+ $\mathcal{L}_{\text{CR}}$	**83.72**	**71.14**	**81.65**	**65.82**

for retrieval inconsistencies through its own reasoning capabilities. Third, small improvements in retrieval quality can compound across many queries in production systems.

5.6.11 Comparison with Query Reformulation

Query reformulation is a popular approach to improve retrieval robustness by rewriting the input query at inference time (He et al., 2016b; Ma et al., 2023). We compare against these methods to determine whether inference-time reformulation can achieve similar coherence improvements to our CR loss, which instead modifies the training objective. We compare against two representative train-free reformulation approaches. All the methods are based on MPNet model.

Centroid (Kostric and Balog, 2024): The retrieval model first computes embeddings for both the original query $\mathbf{e}_q$ and all k reformulations $\mathbf{e}_{r_1}, \ldots, \mathbf{e}_{r_k}$. The final query embedding used for retrieval is the centroid (unweighted average): $\frac{1}{k+1}(\mathbf{e}_q + \sum_i \mathbf{e}_{r_i})$. The intuition is that the center of mass of multiple reformulations will likely correspond better to the user's underlying information need than any single query phrasing. Note that in the original work by Kostric and Balog (2024), a weighted average is used where each reformulation receives a score based on conversation history; since our setting involves single-turn queries without conversation context, we use an unweighted average.

Best: The retrieval model is run on all available reformulations, and the final ranked list is constructed by selecting documents based on the highest score they receive across any reformulation. This allows the model to retrieve documents that may receive low scores with the original query but high scores with a particular reformulation.

Table 5.29 presents results on TREC-DL benchmarks. Both reformulation approaches *underperform* the baseline without reformulation: Centroid drops P@1 by 7 points and NDCG@10 by 2.6 points on TREC-DL '19, while Best drops NDCG@10 by 4.75 points. In contrast, our CR loss maintains the same P@1 while improving NDCG@10 by +1.37 points.

We hypothesize that reformulation methods are better suited to conversational settings where query context evolves across turns, rather than single-turn factoid QA where the original query already expresses the user's intent clearly. In our setting, averaging reformulations (Centroid) may dilute the original query's signal, while selecting by maximum score (Best) may introduce noise from reformulations that drift from the original intent. Additionally, reformulation introduces runtime overhead as it requires an LLM to generate reformulations

for each query. Overall, the results demonstrate that improving coherence through training with CR loss is more effective than attempting to compensate for incoherence at inference time using reformulation approaches. CR loss produces a model that is inherently robust to query variation, with no additional inference cost.

5.6.12 Qualitative Examples

To provide intuition for how CR training affects retrieval behavior, Table 5.30 shows concrete examples of how CR training improves coherence.

Table 5.30: Examples of coherence improvement. FT model retrieves different top-ranked documents for equivalent queries; CR model retrieves consistent results.

Query	Retrieved Top Documents
	Example 1: Flea lifespan
Q1: What is the average lifespan of a flea?	**FT-D1 (Rank #1):** "How long is the life span of a flea? 30–90 days (average). A flea might live a year and a half under ideal conditions... Generally speaking, an adult flea only lives 2–3 months."
Q2: Can you explain the typical duration it takes for a flea to complete its life cycle?	**FT-D2 (Rank #2):** "It takes around 2 days for eggs to hatch, 7 days for the larvae to pupate, and another 7 days until the adult stage is reached."
	CR model: Returns *D1 for both queries.*
	Example 2: Shark thermoregulation
Q1: What mechanism allows some sharks to retain warmth internally?	**FT-D1 (Rank #1):** "White sharks have a counter-current heat exchange system keeping their body temperature above ambient water..."
Q2: How does a select group of sharks maintain a higher body temperature?	**FT-D2 (Rank #2):** "To maintain a warm body in cold water, a warm-bodied shark must burn fuel like a blast furnace..."
	CR model: Returns *D1 for both queries.*

In both examples, the FT model returns different documents for equivalent queries, with the second document being less relevant. The CR model consistently returns the more relevant document for all query phrasings.

5.6.13 Summary

This section demonstrated that dense retrieval models exhibit significant coherence gaps when processing equivalent queries (RQ7), and introduced the Coherence Ranking loss to address this limitation. The key findings are:

1. **Baseline incoherence is substantial**: Even fine-tuned dense models achieve only 0.46–0.54 RBO@5, meaning top-5 rankings differ substantially across equivalent query phrasings. Lexical methods (BM25) are even more sensitive.

2. **CR loss improves both coherence and relevance**: The combination of Query Embedding Alignment and Similarity Margin Consistency improves coherence by up to +30% RBO while simultaneously improving NDCG by up to +1.69 points (RQ8). Critically, individual components hurt performance; only their combination succeeds.

3. **Benefits transfer across domains**: Coherence improvements generalize to BEIR and TREC-DL benchmarks, and propagate to downstream applications including retrieve-and-rerank (+9.3% reranking opportunity) and RAG pipelines (+0.4% accuracy) (RQ10).

4. **Coherence is most critical for complex queries**: When multiple documents have similar relevance scores, coherence degrades severely (0.16–0.17 RBO on MS-MARCO versus 0.46 on the full set). CR loss provides dramatic improvements in these scenarios (+138% on MS-MARCO), more than doubling coherence on NQ (RQ9).

These findings establish that retrieval coherence can be substantially improved through explicit optimization, and that this improvement complements rather than conflicts with accuracy optimization.

5.7 Discussion

This chapter presented four studies: (i) LLM coherence analysis with q-RAG, (ii) coherence-aware LLM training via DPO and SFT, (iii) multilingual evaluation, and (iv) retrieval coherence optimization with CR Loss. All these studies converge on a unified perspective: *coherence is a fundamental property that current training paradigms do not explicitly optimize for, yet can be substantially improved through question clustering.*

5.7.1 Coherence as Understanding, Not Knowledge

A consistent finding across experiments is that coherence failures are primarily failures of understanding rather than knowledge. When a model correctly answers *"What is the capital of France?"* but fails on *"Which city serves as France's capital?"*, the required knowledge is identical, the model demonstrably possesses the information yet fails to access it consistently. This is not an isolated phenomenon: across our LLM experiments, a consistent fraction of question clusters exhibited this incoherent pattern, where models succeeded on some phrasings while failing on equivalent alternatives (Figure 5.1).

The distinction between knowledge gaps and understanding gaps has important implications. A knowledge gap means the model lacks the required information entirely, it fails consistently across all phrasings because the answer is simply not encoded in its parameters. An understanding gap means the model possesses the information but cannot reliably access it. In this scenario, the success depends on whether the surface form of the question happens to activate the right internal representations. The presence of incoherent clusters in our experiments suggests that understanding gaps are a substantial contributor to QA failures, perhaps more than previously recognized.

This reframes how we should approach model improvement. Traditional approaches focus on expanding knowledge: larger training corpora, retrieval augmentation with external documents, continued pre-training on domain-specific text. These approaches address knowledge gaps effectively but leave understanding gaps untouched. Moreover, a model that cannot consistently map equivalent surface forms to the same underlying concept will not be helped by more facts.

The success of q-RAG supports this analysis. Retrieved support questions contain no new factual information as they are simply alternative phrasings of the same information need. The q-RAG method improves accuracy by up to 9 percentage points, outperforming traditional document-based RAG which *does* provide new facts. This counterintuitive result makes sense through the lens of understanding gaps: redundant semantic signal helps the model triangulate the user's intent, increasing the probability of activating the correct parametric knowledge. The bottleneck, for many questions, is not missing knowledge but unreliable access to existing knowledge.

5.7.2 Independence of Coherence and Accuracy

One of the most surprising insights emerging from this work is the disconnect between coherence and accuracy metrics that are often implicitly assumed to vary together. Our results challenge this intuition. For example, Smaug-72B attains the highest accuracy on QRC (83.59%), but its coherence remains only moderate (54.51). In contrast, Llama2-70B exhibits the opposite trend, with lower accuracy (77.69%) but substantially higher coherence (81.36).

This dissociation means the two metrics capture genuinely different aspects of model behavior. A model can be consistently wrong with high coherence and low accuracy or inconsistently right with high accuracy and low coherence. From a user perspective, these failure modes feel quite different. Interacting with Smaug, a user would get correct answers more often on average, but would also experience frustrating inconsistencies where rephrasing a question yields a different answer. Interacting with Llama2, answers would be wrong more often, but at least the behavior would be predictable. Which tradeoff is preferable depends on the application: for a trivia game, Smaug's higher accuracy might win; for a medical information system where users naturally reformulate questions seeking confirmation, Llama2's consistency might matter more.

Additionally, this independence suggests that coherence deserves explicit evaluation alongside traditional accuracy metrics. Current benchmarks average over individual examples, potentially masking systematic inconsistencies. A model with 80% accuracy that fails randomly is quite different from one that fails specifically on certain phrasings, even if both achieve the same aggregate score. The per-cluster evaluation methodology, measuring consistency across equivalent questions rather than aggregate accuracy, reveals failure patterns that traditional metrics obscure. We believe coherence metrics should become standard in QA evaluation, particularly for user-facing applications where query reformulation is common.

5.7.3 Question Clusters as a Unifying Principle

Both q-RAG and the Coherence Ranking Loss leverage question clusters, but in complementary ways that highlight a common underlying principle. The coherence-aware training experiments extend this to a third approach.

First, q-RAG uses clusters to provide redundant semantic signal at inference time. The intuition is that any single question phrasing might or might not activate the relevant parametric knowledge in the LLM as surface features like word choice and syntax influence which internal patterns the model matches. In the same way, a suboptimal match can lead to incorrect retrieval of stored knowledge. By providing multiple equivalent phrasings, we give the model multiple chances to activate the right representations. If even one phrasing triggers strong activation, the model can leverage that signal to produce a correct answer. This explains why retrieved questions outperform generated paraphrases: organically diverse questions from real users probe different aspects of the model's representation space, while LLM-generated paraphrases tend to cluster in similar regions.

On the other hand, CR Loss uses clusters differently: rather than providing redundant signal at inference time, it defines consistency constraints during training. By penalizing ranking discrepancies across equivalent queries, we force the encoder to learn phrasing-invariant representations. The model cannot satisfy the loss by memorizing surface patterns as it must learn that *"What is the capital of France?"* and *"Which city serves as France's capital?"* should produce identical correct document rankings. The result is a model that naturally produces consistent outputs without requiring runtime augmentation.

The coherence-aware training experiments (Section 5.4) demonstrate a third use of clusters: generating preference pairs for alignment. By using q-RAG responses as chosen examples and baseline responses as rejected examples, we create training signal that teaches the model to produce coherent answers directly, without requiring cluster retrieval at inference time. This bridges the inference-time approach of q-RAG and the training-time approach of CR Loss.

Despite targeting different pipeline components, generation vs. retrieval, and operating at different times, inference vs. training, all three approaches share a core insight: *question clusters define the semantic invariances that robust QA systems should respect.* An ideal system would produce identical outputs for all questions within a cluster. The results in this chapter demonstrate that explicit cluster-based optimization, whether providing redundant signal at inference time or encoding consistency constraints during training, moves systems measurably closer to this ideal.

5.7.4 Connection to Previous Chapters

This chapter makes explicit the cluster structure that was implicit in earlier work. Chapter 3 built systems to identify equivalent questions, learning to recognize cluster membership. Chapter 4 trained models to detect ranking perturbations without seeing the query, forcing them to internalize cluster structure implicitly. This chapter directly leverages explicit clusters for coherence optimization, completing the progression from implicit learning to explicit application.

5.7.5 Limitations

Several limitations should be acknowledged. First, both q-RAG and CR Loss depend on cluster quality; noisy clusters could inject misleading signal or incorrect consistency constraints. The retrieval system from Chapter 3 provides high-quality clusters for open-domain questions, and the ablation study in Section 5.6.4 confirms that both QEA and SMC components are necessary, neither alone suffices. However, specialized domains would require adapta-

tion. Second, our methods target factual questions with well-defined answers. For subjective questions, different phrasings may legitimately warrant different responses, requiring careful redefinition of coherence. Third, we did not evaluate q-RAG in multilingual settings. This omission is methodologically motivated: our ablation studies (Section 5.3.2) demonstrated that LLM-generated paraphrases underperform retrieved questions due to limited lexical diversity. Since our multilingual clusters were generated through translation rather than retrieved from a multilingual database, testing q-RAG with these clusters would contradict our own findings. Proper multilingual q-RAG would require either large-scale question databases for each language or cross-lingual retrieval models, resources that do not currently exist.

Regarding the robustness of the coherence metrics, we note that the ablation study in Table 5.21 demonstrates that improvements are not artifacts of specific component choices, and the consistency of results across two datasets, multiple model architectures (Section 5.6.7), and eleven out-of-domain BEIR benchmarks provides evidence of genuine ranking improvements.

Similarly, while Chapter 3 includes a detailed qualitative error analysis of retrieval failures, the present chapter relies on ablation studies and failure case analysis (Table 5.7) as diagnostic tools rather than a structured error taxonomy. A more systematic error analysis across coherence methods would further strengthen the empirical claims.

5.8 Conclusion

This chapter investigated coherence, the consistency of system outputs for semantically equivalent inputs, across question answering pipelines, demonstrating that explicit coherence optimization yields substantial improvements.

5.8.1 Summary of Findings

LLM Coherence (RQ1–RQ4). State-of-the-art LLMs exhibit significant coherence gaps, where models answer some phrasings correctly but fail on semantically equivalent alternatives. Question-Augmented Generation (q-RAG) addresses this by supplementing prompts with retrieved similar questions, improving accuracy by up to 9 percentage points and coherence up to 28 points. The improvement is directly related to a better question understanding rather than external knowledge injection, as confirmed by ablations showing that retrieved questions outperform both LLM-generated paraphrases and traditional document-based RAG.

Multilingual Analysis (RQ5). Coherence correlates with accuracy ($\rho = 0.39$) and model size ($\rho = 0.42$) across six languages and eleven models (3.8B–235B parameters). However, cross-lingual patterns vary substantially by model family: some achieve uniform coherence across languages as they scale, while others maintain significant gaps even at the largest sizes. These findings indicate that multilingual deployments require explicit per-language coherence evaluation.

Coherence-Aware LLM Training (RQ6). Original experiments conducted for this thesis demonstrate that q-RAG's inference-time benefits can be distilled into model parameters through Direct Preference Optimization (DPO) and Supervised Fine-Tuning (SFT). Training on QRC clusters and evaluating on PopQA-TP, both Phi-3-mini and Mistral-7B show improvements in accuracy (+3.93 and +1.72 EM points respectively) and coherence (+23.96 and +5.5 points), with training-time approaches surpassing q-RAG's inference-time coherence improvements. The improvements generalize across datasets, suggesting that coherence is a

learnable property that can be optimized during training without requiring retrieval augmentation at inference time.

Retrieval Coherence (RQ7–RQ10). Dense retrieval models exhibit substantial query sensitivity (0.46–0.54 RBO@5 baseline). The Coherence Ranking Loss, combining Multiple Negative Ranking loss with Query Embedding Alignment and Similarity Margin Consistency, improves coherence by up to 30% RBO while simultaneously improving relevance by up to 1.7% NDCG. Importantly, these improvements are not confined to the training domain. Models trained with CR Loss on MS-MARCO show consistent gains on out-of-domain benchmarks, including the diverse BEIR collection and the TREC-DL 2019/2020 test sets, confirming that coherence training produces genuinely more robust representations rather than overfitting to dataset-specific patterns. The benefits also propagate downstream: in retrieve-and-rerank pipelines, coherent first-stage retrieval increases the probability that the reranker sees the best document across all query phrasings, with +7 to +12 percentage points in reranking opportunity depending on the dataset. In RAG pipelines, more consistent retrieval translates to more consistent generation, though the accuracy improvement is modest (+0.4%).

5.8.2 Implications and Future Directions

The findings suggest that coherence evaluation should complement traditional accuracy metrics, particularly for applications where users naturally reformulate queries. Training on equivalence classes rather than individual examples, as demonstrated by CR Loss, may produce more robust models across NLP tasks beyond QA.

Promising future directions include scaling coherence-aware LLM training to larger models and more diverse datasets. Our preliminary DPO and SFT experiments on Phi-3-mini and Mistral-7B demonstrate feasibility, but extending to 70B+ parameter models and incorporating coherence objectives into pre-training or RLHF pipelines remains unexplored; cross-lingual q-RAG contingent on developing multilingual question retrieval resources; and unified pipelines combining coherence-optimized retrieval with question-augmented generation.

5.8.3 Closing Remarks

The question clustering framework developed throughout this thesis, from identifying equivalent questions (Chapter 3), through implicit cluster learning (Chapter 4), to explicit coherence optimization (this chapter), provides a principled approach to building more robust QA systems. The findings demonstrate that coherence can be substantially improved through explicit optimization, whether at inference time (q-RAG) or during training (CR Loss for retrieval, DPO/SFT for generation).

Beyond coherence, the ability to identify and retrieve semantically equivalent questions has broader applications. The same question retrieval infrastructure that enables q-RAG, mapping an input question to equivalent alternatives, can serve other purposes where semantic equivalence matters. The following chapter explores one such application: using question equivalence to declassify proprietary QA datasets by replacing sensitive questions with semantically equivalent public alternatives, enabling dataset sharing and model training without exposing confidential information.

Chapter 6

Dataset Equivalence and Declassification

The previous chapters developed a comprehensive framework for question understanding, following a natural arc of abstraction. Chapter 3 introduced large-scale question retrieval with QUADRo, operationalizing *question-question equivalence*: given two questions, do they seek the same information? Chapter 4 refined the equivalence signal through specialized pretraining, providing direct technical foundations for this chapter's methodology. Chapter 5 lifted the concept to *cluster-level equivalence*: groups of questions that share answers and exhibit coherent behavior, demonstrating that equivalence relationships have measurable effects on downstream system behavior. While the clustering techniques themselves are not directly employed here, the conceptual progression is essential: understanding that questions form equivalence classes with consistent properties enables thinking about dataset-level equivalence, not just question-level matching.

This final chapter takes the last step to *dataset-level equivalence*. Two datasets are equivalent if models trained on one perform comparably to models trained on the other. This is a functional definition where equivalence is measured by utility preservation rather than content similarity. The concept has immediate practical implications: if we can construct a public dataset equivalent to a proprietary one, we enable data sharing without exposure. The declassification framework operationalizes this concept: given a proprietary dataset, construct a public equivalent that preserves training effectiveness while containing entirely different content.

Building effective Question Answering systems requires training data that reflects real user needs. The most valuable datasets are those derived from customer support logs, enterprise knowledge bases, and production search systems, as they capture authentic information-seeking behavior that synthetic benchmarks cannot replicate. However, these datasets are almost always proprietary, creating a persistent gap between industrial practice and open research.

This gap arises well before public release becomes a concern. Within large organizations, access to sensitive customer data is itself highly constrained by rigid governance policies: access approvals across departments, restricted execution environments, audit requirements, and strict controls on data movement. Researchers may be required to work inside sealed environments where data can be consumed but not exported, debugged interactively, or combined with external tools. These constraints make experimentation slow, fragile, and expensive, substantially impeding research velocity and model iteration, even for teams within the same

organization. And even when these internal barriers are overcome, a more fundamental limitation remains: the resulting research cannot be reproduced or validated by the broader scientific community, undermining a core principle of the scientific method.

Existing data-release strategies fail to resolve this issue. *Data anonymization* focuses on removing personally identifiable information (PII) such as names, addresses, or account numbers, but privacy risk is not limited to explicit identifiers. Seemingly innocuous content can uniquely identify individuals or organizations through contextual cues: combinations of locations, services, or domain-specific references may be sufficient for re-identification, even when no traditional PII is present. At scale, this leads to an unsatisfactory trade-off: either anonymization is too weak to guarantee privacy, or it becomes so aggressive that virtually any token is treated as sensitive, rendering the data unusable. *Synthetic data generation* offers a different path, but typically fails to preserve the fine-grained linguistic, semantic, and distributional properties that make real QA datasets valuable for training and evaluation. *Paraphrasing* retains semantics at the sentence level but often requires sending proprietary content to external APIs, which may itself violate contractual or regulatory constraints.

We propose *dataset declassification*, a fundamentally different approach grounded in a simple but powerful principle: *content that is already public cannot be subject to privacy or confidentiality restrictions*. Rather than modifying or anonymizing proprietary data, declassification replaces it entirely with public content that is semantically equivalent for question answering. Concretely, given a proprietary question-answer pair (q, a), we seek a public question-answer pair (q', a') such that q' induces the same learning and evaluation behavior as q for downstream QA models. This guarantees complete privacy safety by construction, while preserving training and evaluation utility. The term "declassification" draws an analogy to government document declassification, where sensitive information is replaced or redacted to enable public release while preserving the document's essential function.

This concept addresses two complementary scenarios that together capture the main privacy challenges in working with proprietary QA data. Scenario A concerns organizations that want to share valuable datasets but cannot expose original content. Scenario B concerns organizations that want to use sensitive data for training but face regulatory or contractual barriers. In both cases, declassification provides a path forward by decoupling the data's utility from its content.

Scenario A: Dataset and Model Release. Organizations developing QA systems face multiple interrelated challenges when attempting to share their work:

- **Evaluation integrity and contamination.** Organizations that release proprietary evaluation sets face dual risks: exposing sensitive content and invalidating the benchmark as LLMs trained on web crawls may memorize released questions. Declassified "shadow benchmarks" address both concerns since the public version protects original content while the private original remains valid for authoritative evaluation, avoiding contamination and memorization risks.

- **Model memorization and extraction.** Models trained on sensitive data can memorize and regurgitate training examples. Even without releasing training data, releasing a trained model exposes the organization to extraction attacks that may recover private content. Training on declassified data eliminates this risk: if the model memorizes examples, it memorizes public content.

- **Regulatory compliance and auditing.** Emerging AI regulations, such as EU AI Act, GDPR transparency requirements, may require organizations to demonstrate properties of their training data without disclosing the data itself. Declassified datasets serve as auditable proxies: regulators can inspect the declassified version to verify distributional properties, topic coverage, and potential biases without accessing customer data.

- **Academic collaboration and reproducibility.** Organizations wishing to publish research or collaborate with universities face different issues between scientific norms, such as data sharing and reproducibility, and business constraints, as IP protection and contractual obligations. Declassification enables releasing "*shadow datasets*" that support reproduction of experimental results.

- **Right to deletion compliance.** Under GDPR and similar regulations, users can request deletion of their data. If a model was trained on that data, strict compliance may require retraining the model, which is expensive. Training on declassified equivalents from the start sidesteps this issue entirely, as no user data enters in the training pipeline.

Scenario B: Privacy-Preserving Training. An organization has access to valuable domain-specific data that cannot be used directly for training:

- **Customer support data.** A company's support ticket history contains thousands of real user questions with verified solutions: ideal training data for a customer-facing QA system. However, these tickets contain customer names, account details, and proprietary product information. Training directly on this data risks both privacy violations and competitive exposure if the model is later compromised.

- **Internal knowledge bases.** Enterprise wikis, Slack conversations, and internal documentation capture institutional knowledge in question-answer form. Organizations want to build QA systems over this knowledge without exposing internal communications to cloud providers or risking leakage through deployed models.

- **Medical and legal domains.** Healthcare providers hold patient questions alongside clinician responses, and law firms maintain client queries paired with attorney answers. These are highly valuable resources for training domain-specific question answering systems, but using them directly would violate HIPAA, attorney client privilege, and comparable protections.

- **Multi-tenant SaaS platforms.** A SaaS provider wants to improve their product's QA capabilities using patterns learned from customer interactions. However, training directly on Customer A's data and deploying that model for Customer B would violate data isolation agreements as the model itself becomes a vector for cross-tenant data leakage. Training on declassified equivalents avoids this problem: the model learns from public proxies that reflect the same question and answer patterns without including any real customer data.

All these cases share a common solution: mapping proprietary questions and answers to public equivalents, then training or evaluating on the declassified data. The resulting models handle the original domain effectively without ever being exposed to sensitive content.

Our declassification framework operationalizes this concept through two complementary components. First, question retrieval and ranking identifies public counterparts for proprietary questions, adapting systems originally designed to locate equivalent questions for answering (Chapters 3–4) to the task of privacy-preserving dataset transformation. Second, answer reconstruction generates corresponding answers from domain-appropriate corpora, using dense retrieval and LLM-based annotation to preserve label distributions. Together, these mechanisms create declassified datasets that maintain utility for training and evaluation without exposing original content.

This chapter makes the following contributions:

1. **Declassification Framework**: a complete pipeline combining question declassification via QUADRo retrieval with answer declassification from domain-appropriate corpora, preserving both semantic equivalence and label distributions
2. **Training Set Declassification**: empirical validation demonstrating that models trained on declassified data achieve performance within $\Delta \approx 0$ of models trained on original data for WikiQA and TrecQA
3. **Test Set Declassification**: extension to evaluation data, enabling organizations to release "shadow benchmarks" that preserve evaluation validity while protecting original content from contamination
4. **Analysis of Success Conditions**: identification of critical factors, establishing that domain-matched answer reconstruction is essential and that declassification effectiveness depends on the coverage overlap between proprietary questions and public retrieval corpora

The work presented in this chapter has been submitted for publication and is currently under review.

The remainder of this chapter is organized as follows. Section 6.1 formalizes the declassification problem and defines utility preservation. Section 6.2 describes the declassification framework, including question retrieval and answer reconstruction. Section 6.3 presents the experimental setup. Section 6.4 reports results across evaluation, training, and benchmark declassification settings. Section 6.5 analyzes mapping quality, success conditions, and boundary cases. Section 6.6 discusses implications and limitations. Section 6.7 concludes the chapter.

6.1 Problem Formulation

Having motivated declassification through practical scenarios, we now formalize the problem and establish the properties that a successful declassification must satisfy.

A QA dataset is modeled as a collection of triplets

$$\mathcal{D} = \{(q_i, c_i, a_i)\}_{i=1}^{N},$$

where q_i is a question, c_i is its associated context, and a_i is an answer. The interpretation is task-dependent:

- **Answer selection (AS2)**: c_i contains candidate answer sentences, a_i indicates which candidates are correct (i.e., $c_i = (\mathcal{A}_i^+, \mathcal{A}_i^-)$ with correctness labels)
- **Generative QA**: c_i may be empty, a_i is the reference answer used for evaluation

6.1.1 Dataset Declassification

Given a private dataset $\mathcal{D}$, the goal of declassification is to construct a new dataset $\mathcal{D}'$ that (i) contains no private or sensitive content, and (ii) preserves the utility of the original dataset for both training and evaluation. Formally, declassification is defined as a mapping

$$\Phi : \mathcal{D} \to \mathcal{D}',$$

where $\mathcal{D}'$ consists exclusively of public content and is safe to release.

We focus on pointwise declassification, decomposing the dataset-level mapping Φ into a function that operates on individual examples:

$$\phi(q, c, a) = (q', c', a'),$$

which maps a private example to a declassified one. The dataset-level mapping is then $\Phi(\mathcal{D}) = \{\phi(q_i, c_i, a_i)\}_{i=1}^{N}$. In our setting, ϕ is further decomposed into independent mappings for each component:

$$\phi(q, c, a) = (\phi_Q(q), \phi_C(c), \phi_A(a)),$$

where ϕ_Q maps questions via retrieval, and ϕ_C, ϕ_A reconstruct context and answers from domain-appropriate corpora.

6.1.2 Utility Preservation

Utility preservation is defined with respect to a downstream model m and a performance metric $p_m(\cdot)$. We distinguish two complementary notions.

Evaluation utility measures whether a model's performance changes when evaluated on declassified data. Let T be an original test set and $T' = \Phi(T)$ its declassified counterpart. We define the evaluation utility loss as

$$\Delta_{\text{test}} = p_m(T) - p_m(T'),$$

and say that declassification preserves evaluation utility if $|\Delta_{\text{test}}| \approx 0$.

Training utility measures whether training on declassified data produces equivalent models. Let m be a model trained on original data T_r and m' a model trained on its declassified version $T_r' = \Phi(T_r)$, both evaluated on the same test set T. We define the training utility loss as

$$\Delta_{\text{train}} = p_m(T) - p_{m'}(T),$$

and say that declassification preserves training utility if $|\Delta_{\text{train}}| \approx 0$.

These definitions formalize the requirement that declassified datasets behave equivalently to their private counterparts for both evaluation and training.

6.1.3 Sufficient Conditions for Equivalence

The declassification function ϕ takes different forms depending on the task, but the core conditions for equivalence remain consistent. We identify three sufficient conditions for declassification to preserve utility (i.e., $|\Delta_{\text{train}}| \approx 0$ and $|\Delta_{\text{test}}| \approx 0$):

1. **Question Similarity**: each mapped question q_i' is semantically similar to q_i, seeking the same or closely related information

2. **Answer Correctness**: the reconstructed answers correctly answer the mapped questions (for AS2: positive candidates are correct, negative candidates are incorrect; for generative QA: reference answers are correct)
3. **Domain Consistency**: the reconstructed answers come from the same domain as the originals

Note that Condition 1 alone is insufficient: since mapped questions are similar but not identical, the original answers may not correctly answer q'_i. This motivates Condition 2 and our answer declassification approach.

Our declassification pipeline enforces these conditions as follows:

- **Condition 1** is enforced by QUADRo retrieval with cross-encoder reranking, which finds semantically similar questions from public corpora.
- **Condition 2** is enforced by LLM-based annotation that verifies correctness for the mapped question.
- **Condition 3** is enforced by reconstructing answers from domain-appropriate corpora (Wikipedia for encyclopedic questions, CCNews for news-based questions).

We return to these conditions in Section 6.6, where ablation experiments reveal what happens when each condition is violated.

6.2 The Declassification Framework

Having established the conditions for dataset equivalence, we now describe how our framework enforces them in practice. The declassification pipeline consists of two main stages: question mapping and answer reconstruction. Figure 6.1 illustrates the complete process.

6.2.1 Question Declassification

The first stage maps each proprietary question to a semantically similar public question. We use retrieval rather than generation for two reasons: retrieval guarantees that the output exists in a known public corpus, ensuring no proprietary content leaks through generated text, and retrieval provides a similarity score that quantifies mapping confidence.

Given a proprietary question q, we search the QUADRo database containing 38 million question-answer pairs as described in Chapters 3–4:

1. **Dense Retrieval**: Encode q using the QUADRo bi-encoder, which was specifically trained for query-to-question matching, and retrieve the top-k candidates (we use $k = 100$) based on cosine similarity
2. **Optional Reranking**: Apply the QRP-enhanced cross-encoder from Chapter 4 to rerank the top candidates, improving precision at the first position
3. **Selection**: Select the highest-ranked question q' as the mapped surrogate

Not all questions map equally well. Questions about common topics ("*What causes the northern lights?*") typically find near-exact matches in QUADRo, while highly specific or unusual questions may only find partial matches. As we show in Section 6.4, the framework is robust to this variation.

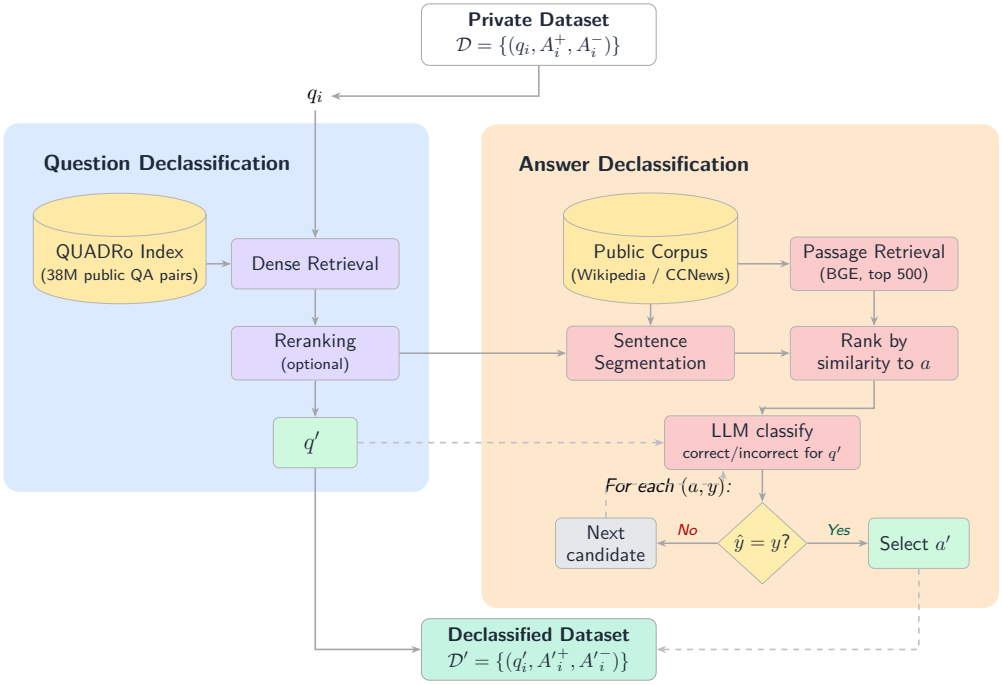

Figure 6.1: The declassification pipeline. Given a proprietary example $(q, \mathcal{A}^+, \mathcal{A}^-)$, question declassification finds a similar public question q' via QUADRo retrieval and reranking. Answer declassification then reconstructs answers from a domain-appropriate corpus: for each original answer a with label y, we retrieve passages, segment into sentences, rank by similarity to a, and use an LLM to find a replacement a' with matching label (with early stopping for efficiency).

6.2.2 Answer Declassification

The mapped question q' requires associated answers. Since q' is similar but not identical to the original question q, the original answers may not be correct for q'. Answer declassification reconstructs answers from domain-appropriate public corpora while preserving label distributions.

The pipeline consists of three steps:

1. **Passage Retrieval**: For each mapped question q', we retrieve the top 500 passages from the domain corpus using Pyserini (Lin et al., 2021) with BGE-base-v1.5 (Xiao et al., 2023a) as the encoder. The corpus choice depends on the dataset: Wikipedia for WikiQA and generative QA benchmarks, CCNews for TrecQA. In general, corpus selection should match the original answer domain (e.g., PubMed for medical QA, case law databases for legal QA).

2. **Sentence Segmentation**: We split retrieved passages into individual sentences using NLTK's (Bird and Loper, 2004) sentence tokenizer, then deduplicate to obtain approximately 2000 candidate sentences.

3. **Answer Replacement**: For each original answer a with label y (positive or negative), we find a replacement a':

   (a) Rank candidates by semantic similarity to a using MPNet embeddings[1]

[1] `sentence-transformers/paraphrase-mpnet-base-v2`

Algorithm 1 Answer Declassification procedure with label-preserving replacement and early stopping.

Input: Public corpus $\mathcal{C}$, mapped question q', original answers $\{(a_j, y_j)\}$
Output: Declassified answers $\{(a'_j, y'_j)\}$

Retrieve top 500 passages from $\mathcal{C}$ for q' using a dense passage retriever.
Segment passages into a sentence pool $\mathcal{S} = \{s_1, s_2, \dots\}$.
for each original answer (a_j, y_j) **do**
    Rank sentences $s_i \in \mathcal{S}$ by semantic similarity to a_j.
    Let $\mathcal{S}_K$ be the top $K = 200$ ranked sentences.
    found $\leftarrow$ FALSE
    for $s_i \in \mathcal{S}_K$ in ranked order **do**
        Use LLM to classify s_i as correct or incorrect for q', yielding $\hat{y}_i$.
        if $\hat{y}_i = y_j$ **then**
            $a'_j \leftarrow s_i$, $y'_j \leftarrow y_j$, *found* $\leftarrow$ TRUE
            break
        end if
    end for
    if not *found* **then**
        $a'_j \leftarrow s_1$, $y'_j \leftarrow \hat{y}_1$ ▷ Fallback to most similar
    end if
end for

(b) Starting from the most similar, use an LLM (Qwen3-80B (Yang et al., 2025)) to classify whether the candidate correctly answers q'

(c) If the LLM label matches y, select this candidate as a' and stop

(d) If no match within the top 200 candidates, fall back to the most similar candidate

The early stopping mechanism makes annotation efficient, while the fallback ensures complete coverage. In practice, fallback is rare: 2% of answers for WikiQA and 4% for TrecQA. For generative QA datasets such as OpenBookQA and SimpleQA, we apply the same pipeline but reconstruct only the reference answer, equivalent to a single positive candidate with no negatives. Algorithm 1 summarizes the answer declassification process.

6.3 Experimental Setup

Having described the declassification framework, we evaluate dataset declassification in a fully controlled setting using only public benchmarks. Since proprietary datasets cannot be shared, we simulate declassification by treating public QA datasets as if they were private and constructing declassified counterparts. This enables a direct and reproducible measurement of whether declassified data preserves evaluation behavior (evaluation utility) and learning signal (training utility).

Our experiments address the research questions summarized in Table 6.1.

6.3.1 Datasets

We evaluate declassification across four datasets spanning different QA paradigms. WikiQA and TrecQA are AS2 benchmarks where we declassify questions and candidate answers (both positive and negative). OpenBookQA and SimpleQA are generative QA benchmarks where

Table 6.1: Research questions for dataset declassification evaluation.

RQ	Question	Section
RQ1	Does declassification preserve utility for both training and evaluation?	6.4.1, 6.4.2, 6.4.3
RQ2	Which components are essential for effective declassification?	6.4.4
RQ3	What are the boundary conditions and failure modes of declassification?	6.5.1, 6.5.2
RQ4	Can mapping similarity predict declassification success?	6.5.3, 6.5.2

we declassify questions and reference answers. Together, these datasets cover encyclopedic, news, scientific reasoning, and factual knowledge domains, allowing us to assess how declassification effectiveness varies with domain characteristics and question distributions. Table 6.2 summarizes dataset statistics.

Table 6.2: Dataset statistics for declassification experiments.

Dataset	Task	Train	Dev	Test
WikiQA	AS2	2118	296	237
TrecQA	AS2	94	65	68
OpenBookQA	Generative QA	4057	500	500
SimpleQA	Generative QA	—	—	4326

WikiQA (Yang et al., 2015) is constructed from real Bing query logs, with candidate answer sentences extracted from Wikipedia and manually annotated. A peculiarity of the dataset is that many questions have either no correct answers or only correct answers, creating evaluation challenges. Following standard practice in the literature (Gabburo et al., 2024a), we use the *clean* splits, dev_clean and test_clean, which contain only questions with at least one correct and at least one incorrect answer, ensuring stable and comparable evaluation across models. The encyclopedic nature of the answers makes Wikipedia a natural source for answer reconstruction.

TrecQA (Wang et al., 2007) derives from TREC tracks 8–13 and is one of the classic benchmarks for answer sentence selection. It consists of factoid questions with candidate sentences from newswire corpora. Since the original structure contains noise, processed "clean" versions are now standard: they include questions with consistent annotations and well-defined answers. As for WikiQA, we use the clean splits, dev_clean and test_clean, which retain only unambiguously evaluable examples. Unlike WikiQA, TrecQA questions often concern specific news events, people in the news, and temporal facts from the late 1990s and early 2000s that may not be well-covered in Wikipedia's encyclopedic content. For this dataset, we use CCNews (Wenzek et al., 2019), a large corpus of news articles, as the answer declassification source, better matching the domain characteristics of the original data.

OpenBookQA (Mihaylov et al., 2018) is a multiple-choice dataset designed to evaluate models' ability to combine basic scientific facts with common-sense knowledge through multi-

step reasoning. It includes 4057 training questions, 500 development questions, and a test set of 500 questions that serves as the official benchmark. For our experiments, we use OpenBookQA in an *open-ended* setting: rather than selecting from four choices, models must generate answers directly. We declassify both questions via QUADRo and reference answers from Wikipedia, following the same pipeline used for AS2 positive answers.

SimpleQA (Wei et al., 2024) is a factuality benchmark created by OpenAI containing 4326 questions with unique, verifiable answers. However, SimpleQA was specifically designed to be "uncontaminated": the authors targeted complex and niche facts unlikely to appear in public data. This directly conflicts with QUADRo, which contains common questions from public sources. The mismatch is fundamental: SimpleQA asks questions like "*John Barton "Jack" Grimwood, was an English footballer who played as a half-back, he joined Manchester United in May 1919, and made his debut for the club in the first Manchester derby, on which date?*" while QUADRo contains questions like "*what was the first year of Manchester's football derby in england?*".

6.3.2 Models

We use different model families for different tasks, reflecting how these benchmarks are typically evaluated in the literature. Within each category, we include models of varying sizes to verify that declassification effects are consistent across model capacities.

For AS2, we fine-tune two cross-encoders (Lauriola and Moschitti, 2021a) that jointly encode question-answer pairs and produce relevance scores. DeBERTa-v3-base (He et al., 2023) (184M parameters) represents the current state-of-the-art for AS2 (Gabburo et al., 2024a) and serves as our primary evaluation model. MiniLM-L12 (Wang et al., 2020a) (33M parameters) is a distilled model that maintains competitive performance while being smaller and faster, allowing us to verify that declassification effects hold across model capacities.

For generative QA, we evaluate four instruction-tuned LLMs in zero-shot settings, spanning different scales: Phi-3-mini (Abdin et al., 2024a) (3.8B), notable for strong performance relative to its size; Ministral (Jiang et al., 2023b) (8B), representing mid-range open-weight LLMs; Phi-4 (Abdin et al., 2024b) (14.7B), with improved reasoning capabilities; and Qwen-3-30B (Yang et al., 2025) (30B), among the largest models we evaluate. To assess answer correctness, we use Qwen-3-80B (Yang et al., 2025) as judge following the original SimpleQA evaluation protocol (Wei et al., 2024); the prompt is provided in Appendix D.2.

6.3.3 Declassification Configurations

We compare our declassification method against generative baselines.

As generative baselines, we consider two approaches. **Gen-PR** paraphrases questions and answer sentences via Qwen-3-80B, applied sentence-by-sentence. **Gen-BT** uses backtranslation (EN→DE→EN) via NLLB-200 (Team et al., 2022), also applied sentence-by-sentence.

Map$_{RR}$ is our full declassification method (Section 6.2), implementing question declassification ϕ_Q via QUADRo retrieval with cross-encoder reranking and answer declassification ϕ_A via retrieval from domain-matched public corpora.

We also evaluate three ablations to isolate the contribution of each component. **Map**$_R$ tests the impact of reranking: same as Map$_{RR}$ but using only dense retrieval without cross-encoder reranking, measuring whether the additional reranking step from Chapter 4 provides

meaningful improvements. $\mathbf{Map}_{RR}{}^{-A}$ tests whether question mapping alone suffices: this configuration replaces questions via QUADRo but retains original answers, violating Condition 2 (label correctness) since the original answers were annotated with respect to the original questions, not the mapped ones. $\mathbf{Map}_{RR}{}^{mix}$ tests whether mixing answer sources works: this configuration uses QUADRo's associated answer as positive, but retains additional original positives and all original negatives, creating possible errors and style mismatch between QUADRo's heterogeneous web sources and the original corpus style. This ablation is AS2-specific since generative QA has only a single reference answer per question.

Table 6.3 summarizes the configurations.

Table 6.3: Declassification configuration naming convention.

Name	Description
Original	Baseline: unmodified original data
Gen-PR	LLM paraphrasing via Qwen3-80B
Gen-BT	Backtranslation EN→DE→EN via NLLB-200
Map_{RR}	Full declassification: QUADRo retrieval + reranking + answer declassification
Ablations	
Map_{R}	Without reranking (retrieval only + answer declassification)
$\text{Map}_{RR}{}^{-A}$	Without answer declassification (questions only)
$\text{Map}_{RR}{}^{mix}$	Mixed answer sources: QUADRo pos + original neg (AS2 only)

6.3.4 Evaluation Protocol

Our main metric is answer accuracy: for AS2, we use Precision@1 (P@1) to measure whether the top-ranked candidate is correct. We also report Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and NDCG for completeness. For generative QA, we use accuracy as judged by Qwen-80B following the SimpleQA protocol and prompt; details of the prompts are available on Appendix C.1.

We evaluate three declassification scenarios that together characterize utility preservation:

1. **Evaluation Utility** (Section 6.4.1): Models trained on original data, evaluated on declassified test sets. Tests whether declassified benchmarks preserve evaluation validity.

2. **Training Utility** (Section 6.4.2): Models trained on declassified training data, evaluated on original test sets. Tests whether declassified training data preserves learning signal.

3. **Benchmark Declassification** (Section 6.4.3): Models trained and evaluated entirely on declassified data. Tests whether complete "shadow benchmarks" can replace original datasets.

6.3.5 Training Configuration

For answer selection experiments, we perform hyperparameter optimization for each run to ensure fair comparison across configurations. The search space includes:

- Learning rate: $\{5 \times 10^{-6}, 7 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}\}$
- Batch size: $\{8, 16, 32, 64\}$
- Warmup: 10% of training steps
- Maximum sequence length: 256 tokens
- Training: up to 15 epochs with early stopping (patience of 3 epochs)

We optimize on P@1 using the development set. Each configuration undergoes the same hyperparameter search to ensure differences reflect the declassification strategy rather than suboptimal tuning. We report results averaged over 3 random seeds, with standard deviation computed across runs.

For generative QA experiments, we evaluate LLMs in a zero-shot setting without any fine-tuning. To ensure reproducibility, we use greedy decoding with temperature set to 0 and sampling disabled.

All experiments were conducted on a machine with 8 NVIDIA L40S GPUs (48GB each). Training a single AS2 model takes ≈30 minutes depending on dataset size and hyperparameters.

6.4 Results

We present results organized by the three evaluation scenarios: evaluation utility (declassify test only), training utility (declassify train only), and benchmark declassification (declassify both). Rather than reporting absolute performance, we focus on the performance delta (Δ) relative to the baseline. This framing captures the core question: does declassification preserve utility? A delta near zero means the declassified dataset is functionally equivalent to the original, regardless of whether absolute performance is 70% or 90%.

6.4.1 Evaluation Utility (Declassify Test Only)

We first test whether declassified *test* sets preserve evaluation validity. Models are trained on the original training data and evaluated on (i) the original test set and (ii) the declassified test set. Table 6.4 reports results across all four datasets. Configuration names are summarized in Table 6.3.

Key Findings

Map$_{RR}$ preserves evaluation utility across AS2 and OpenBookQA (RQ1). For AS2, Map$_{RR}$ yields $|\Delta| \leq 2$ points across all dataset-model combinations: WikiQA shows $\Delta = -0.8$ (DeBERTa) and $\Delta = -2.0$ (MiniLM), while TrecQA shows $\Delta = -0.5$ and $\Delta = -0.6$ respectively. These deltas are substantially smaller than generative baselines, which show positive shifts of +1 to +5 points on AS2, indicating that paraphrased questions become *easier* for trained rankers, likely due to simplified syntax. For OpenBookQA, Map$_{RR}$ achieves near-perfect preservation: two models (Phi-3, Phi-4) show exactly $\Delta = 0$, while the others show $|\Delta| \leq 0.4$. Crucially, model rankings are preserved: the relative ordering Qwen-3 > Phi-4 > Phi-3 > Ministral remains constant on both original and declassified questions, demonstrating that declassified benchmarks maintain discriminative power for comparing models.

Table 6.4: Evaluation utility on declassified test sets. Models trained on original data. "Private" shows absolute performance on original test set; other columns show Δ relative to Private. ± denotes std across runs (AS2 only).

Dataset	Model	Private	Gen-PR	Gen-BT	Map_{RR}
Answer Sentence Selection (AS2)					
WikiQA	DeBERTa	$81.4_{\pm 0.8}$	$+5.1_{\pm 0.8}$	$+4.6_{\pm 0.8}$	$\mathbf{-0.8_{\pm 1.8}}$
	MiniLM	$70.6_{\pm 2.5}$	$+3.6_{\pm 2.5}$	$+2.6_{\pm 2.5}$	$\mathbf{-2.0_{\pm 3.3}}$
TrecQA	DeBERTa	$85.8_{\pm 0.8}$	$+1.4_{\pm 1.1}$	$+0.5_{\pm 1.1}$	$\mathbf{-0.5_{\pm 1.1}}$
	MiniLM	$72.5_{\pm 1.3}$	$+0.5_{\pm 1.3}$	$-1.5_{\pm 1.3}$	$\mathbf{-0.6_{\pm 1.9}}$
Generative QA					
OpenBookQA	Phi-3	53.5	−0.6	−0.9	**0.0**
	Ministral	49.2	−1.6	−5.0	**−0.4**
	Phi-4	59.8	−2.4	−3.0	**0.0**
	Qwen-3	60.8	−0.5	−0.2	**−0.2**
SimpleQA	Phi-3	4.0	**−0.8**	+0.8	+11.6
	Ministral	2.7	**−0.4**	−0.4	+8.4
	Phi-4	8.4	**−0.9**	+1.1	+12.0
	Qwen-3	18.5	**−3.3**	−4.1	+5.8

SimpleQA reveals boundary conditions where declassification changes task difficulty (RQ3). Unlike the near-zero deltas on other datasets, SimpleQA shows +5.8 to +12 point *improvements* as the models perform substantially better on mapped questions. This is not an improvement in model capability but a *difficulty shift*: the mapped questions are systematically easier than the originals. The backtranslation baseline (Gen-BT) confirms this diagnosis: it achieves $|\Delta| \leq 4.1$ on SimpleQA, comparable to other datasets, because it paraphrases without changing semantic content. In contrast, Map_{RR} shows $|\Delta| > 5$ because it maps to *different, easier questions* from QUADRo. This reveals a fundamental boundary condition: when the original dataset specifically targets questions *absent* from public data (as SimpleQA does by design), mapping to public questions necessarily changes task difficulty. We analyze this phenomenon in detail in Section 6.5.2.

6.4.2 Training Utility (Declassify Train Only)

We next measure whether declassified *training* data provides comparable learning signal. We train AS2 rankers on declassified training sets and evaluate them on the original (private) test set. Since generative QA models are evaluated zero-shot, this analysis applies only to AS2.

Key Findings.

Map_{RR} preserves training utility universally (RQ1). Map_{RR} achieves $|\Delta| \leq 2$ points across all dataset-model combinations: WikiQA shows $\Delta = +0.05$ (DeBERTa) and $\Delta = -0.3$ (MiniLM), while TrecQA shows $\Delta = -0.8$ and $\Delta = -1.2$ respectively. These near-zero deltas demonstrate that models trained on declassified data learn effectively and generalize to the original test distribution. The consistency across both encyclopedic (WikiQA) and news-

Table 6.5: Training utility on AS2. Models trained on declassified training sets, evaluated on original test sets. Values are Δ relative to models trained on original data.

Dataset	Model	Private	Gen-PR	Gen-BT	Map_{RR}
WikiQA	DeBERTa	$81.4_{\pm 0.8}$	$-4.1_{\pm 2.1}$	$-8.9_{\pm 1.8}$	$\mathbf{+0.05_{\pm 1.1}}$
	MiniLM	$70.6_{\pm 2.5}$	$-3.9_{\pm 2.7}$	$-5.9_{\pm 2.1}$	$\mathbf{-0.3_{\pm 3.2}}$
TrecQA	DeBERTa	$85.8_{\pm 0.8}$	$-29.1_{\pm 6.4}$	$-4.9_{\pm 2.1}$	$\mathbf{-0.8_{\pm 1.9}}$
	MiniLM	$72.5_{\pm 1.3}$	$-25.0_{\pm 16.4}$	$-2.1_{\pm 1.4}$	$\mathbf{-1.2_{\pm 3.7}}$

domain (TrecQA) datasets suggests that our retrieval-based approach preserves the essential characteristics of the training signal regardless of domain.

Generative baselines fail in complementary ways, revealing domain-specific vulnerabilities. Gen-PR loses only 4.1 points on WikiQA but 29.1 points on TrecQA, while Gen-BT shows the opposite pattern: 8.9 points on WikiQA but only 4.9 on TrecQA. Neither approach works universally. The asymmetry reveals why: Gen-PR rewrites questions semantically, which works for WikiQA's generic encyclopedic questions ("*how are glaciers formed?*") but corrupts TrecQA's specific named entities and temporal references ("*Who won the 1998 World Series?*"). Gen-BT preserves entities through round-trip translation but introduces less semantic variation, making it more robust on entity-rich TrecQA but less effective on WikiQA where semantic diversity matters. Additionally, Gen-PR shows extreme instability on TrecQA: the standard deviation for TrecQA/MiniLM is ±16.4, which is more than half of the mean value, indicating that paraphrasing produces highly variable training signals depending on which specific paraphrases are generated. This unpredictability makes generative approaches unsuitable for reliable dataset release.

6.4.3 Benchmark Declassification (Declassify Train and Test)

Finally, we evaluate full benchmark declassification, where both training and test sets are declassified, and models are trained and evaluated entirely on declassified data. This setting corresponds to releasing a fully declassified benchmark intended to replace the original dataset.

Table 6.6: Benchmark declassification on AS2. Models trained and evaluated on declassified data. Values are Δ relative to the private benchmark.

Dataset	Model	Private	Gen-PR	Gen-BT	Map_{RR}
WikiQA	DeBERTa	$81.4_{\pm 0.8}$	$-1.2_{\pm 0.8}$	$-3.6_{\pm 0.9}$	$\mathbf{-0.4_{\pm 1.8}}$
	MiniLM	$70.6_{\pm 2.5}$	$-3.5_{\pm 2.5}$	$-2.8_{\pm 2.5}$	$\mathbf{-1.5_{\pm 3.3}}$
TrecQA	DeBERTa	$85.8_{\pm 0.8}$	$-17.7_{\pm 0.8}$	$-4.9_{\pm 0.8}$	$\mathbf{-0.2_{\pm 1.2}}$
	MiniLM	$72.5_{\pm 1.3}$	$-26.4_{\pm 1.3}$	$-2.5_{\pm 1.4}$	$\mathbf{-0.4_{\pm 1.9}}$

Key Findings.

Map_{RR} enables complete shadow benchmarks (RQ1). In the full benchmark setting where both training and test sets are declassified, Map_{RR} yields remarkably small deltas: WikiQA shows $\Delta = -0.4$ (DeBERTa) and $\Delta = -1.5$ (MiniLM), while TrecQA shows $\Delta =$

-0.2 and $\Delta = -0.4$ respectively. These results demonstrate that organizations can release fully declassified "shadow benchmarks" that preserve both training and evaluation behavior. A model developed entirely on the declassified version would achieve nearly identical performance to one developed on the original, making the declassified benchmark a valid substitute for research and development purposes.

Generative baselines remain inconsistent when applied to both splits. Gen-PR loses 17–26 points on TrecQA when both training and test are paraphrased (training-only: 25–29 points). Interestingly, matching training and test declassification provides no advantage for Map_{RR}: performance on matched declassification (train and test both Map_{RR}) is comparable to mismatched settings (original train, Map_{RR} test from Table 6.4). This indicates that declassification does not introduce systematic biases that could be exploited by training on similarly-processed data which is an essential property for fair benchmarking, as it ensures the declassified benchmark cannot be "gamed" by methods that overfit to declassification artifacts.

6.4.4 Ablation Studies

Having established that Map_{RR} preserves utility, we now isolate the contribution of each pipeline component to understand which elements are essential (RQ2). Our full method combines three components: (1) cross-encoder reranking for question mapping, (2) answer declassification via corpus retrieval, and (3) domain-matched answer sources. We systematically remove each component to measure its impact.

The ablations test specific hypotheses. Map_{R} (no reranking) tests whether the computationally expensive cross-encoder step is necessary, or whether dense retrieval alone suffices. $\text{Map}_{RR}{}^{-A}$ (no answer declassification) tests whether question mapping alone preserves utility, or whether answer reconstruction is essential. $\text{Map}_{RR}{}^{mix}$ (mixed answer sources) tests whether domain matching matters, by using QUADRo answers as positives while retaining original corpus negatives.

Table 6.7 presents training utility results. A near-zero delta indicates the removed component is not critical for training utility; a large negative delta indicates the component is essential. Figure 6.2 visualizes the comparison across configurations for DeBERTa.

Table 6.7: Ablation study on AS2 training utility. All values are ΔP@1 relative to Original baseline. Bold indicates $|\Delta| \leq 2$.

Configuration	WikiQA		TrecQA	
	DeBERTa (Original: 81.43%)	MiniLM (Original: 70.60%)	DeBERTa (Original: 85.78%)	MiniLM (Original: 72.54%)
Map_{RR}	$\mathbf{+0.05}_{\pm 1.11}$	$\mathbf{-0.32}_{\pm 3.22}$	$\mathbf{-0.80}_{\pm 1.89}$	$\mathbf{-1.23}_{\pm 3.65}$
Ablations				
Map_{R} (no reranking)	$\mathbf{+0.00}_{\pm 0.94}$	$\mathbf{-1.25}_{\pm 3.94}$	$\mathbf{-1.21}_{\pm 1.56}$	$\mathbf{-1.45}_{\pm 1.89}$
$\text{Map}_{RR}{}^{-A}$ (no answer decl.)	$\mathbf{-1.69}_{\pm 1.20}$	$-14.52_{\pm 2.55}$	$-24.74_{\pm 3.50}$	$-25.70_{\pm 1.43}$
$\text{Map}_{RR}{}^{mix}$ (mixed answers)	$-20.82_{\pm 1.31}$	$-24.33_{\pm 2.87}$	$-5.54_{\pm 1.63}$	$-21.70_{\pm 1.38}$

Key Findings

Answer Declassification is Essential (RQ2) the $\text{Map}_{RR}{}^{-A}$ ablation (question mapping without answer reconstruction) reveals a critical finding: question mapping alone is insuffi-

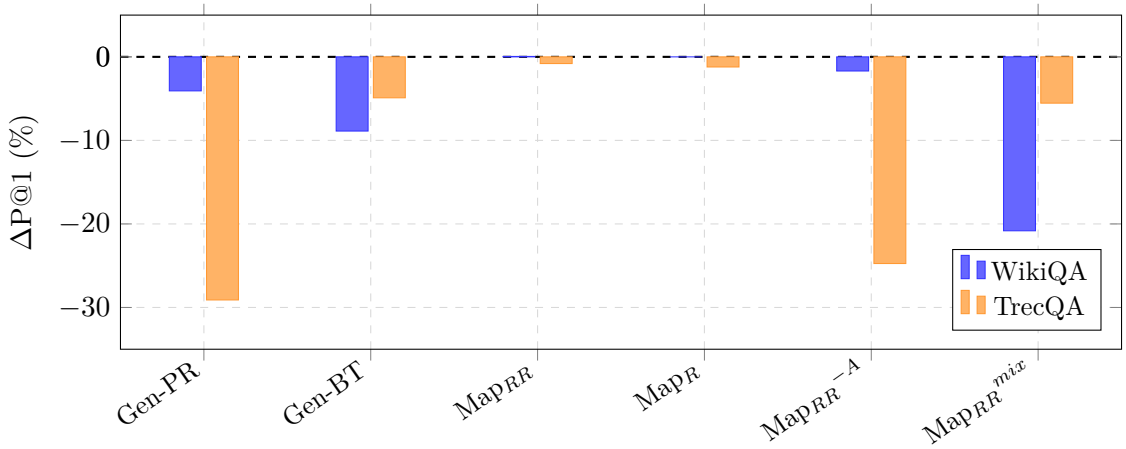

Figure 6.2: Performance delta (ΔP@1) relative to baseline for WikiQA and TrecQA (DeBERTa-v3). The horizontal line at zero represents baseline performance. Only full declassification (Map$_{RR}$, Map$_{R}$) consistently achieves near-zero delta across both datasets.

cient. On WikiQA/DeBERTa, Map$_{RR}^{-A}$ loses only 1.7 points, but on TrecQA/DeBERTa it loses 24.7 points. This asymmetry violates Condition 2 from Section 6.1.3: the original answers were annotated for the original questions, not the mapped ones.

For WikiQA's encyclopedic questions, this mismatch is often tolerable as a sentence explaining glacier formation remains valid whether the question asks "*how are glacier caves formed?*" or "*what are glacial caves formed from?*". For TrecQA's news questions, the mismatch is catastrophic: an answer about Lou Gehrig's consecutive games streak may contain the number 2,130, but if the mapped question subtly shifts focus, the answer's relevance degrades. In this scenario, small models are more sensitive: MiniLM shows $\Delta = -14.5$ on WikiQA Map$_{RR}^{-A}$ compared to DeBERTa's -1.7, suggesting that larger models' additional capacity provides robustness to distributional shifts.

Reranking Provides Consistent Improvement for Training (RQ2) comparing Map$_{RR}$ to Map$_{R}$, reranking provides small but consistent improvements. On WikiQA, both achieve near-zero deltas ($\Delta = +0.05$ vs $\Delta = 0.00$); on TrecQA, Map$_{RR}$ shows -0.80 vs Map$_{R}$'s -1.21. The cross-encoder reranker refines initial retrieval, but the gain is modest because dense retrieval already achieves high-quality mappings when answer reconstruction is complete. However, as we show in Section 6.5.2, reranking becomes *essential* for test set declassification where difficulty preservation matters.

Mixed Answer Sources Fail Due to Style Mismatch the Map$_{RR}^{mix}$ ablation exhibits inverted behavior across datasets: it fails dramatically on WikiQA ($\Delta = -21$) but partially succeeds on TrecQA ($\Delta = -5.5$). This inversion reveals the importance of stylistic consistency.

In Map$_{RR}^{mix}$, the positive answer comes from QUADRo (web sources like forums and FAQs) while negatives remain from the original corpus (Wikipedia for WikiQA, news for TrecQA). On WikiQA, the stylistic gap is large: QUADRo's informal, explanatory answers contrast sharply with Wikipedia's encyclopedic prose. The model learns to distinguish answers by *style* rather than *correctness*, then fails at test time when all candidates come from Wikipedia. On TrecQA, the gap is smaller because both QUADRo and news answers share factual, entity-focused characteristics. This finding motivates our full reconstruction approach: by sourcing both positive and negative answers from the same domain-matched corpus, we

ensure training and test distributions align.

6.5 Analysis

The previous section established that Map_{RR} preserves utility across standard benchmarks, with near-zero deltas on AS2 and OpenBookQA. However, aggregate metrics can obscure important patterns: what do successful mappings look like? Why does SimpleQA show difficulty shifts while other datasets do not? Can we predict which examples will declassify well?

This section addresses these questions through three complementary analyses. First, we examine mapping quality through concrete examples, revealing both successful declassifications and systematic failure modes (Section 6.5.1). Second, we analyze SimpleQA in depth to understand why it represents a boundary condition for our approach (Section 6.5.2). Third, we investigate whether mapping similarity scores can predict declassification success, enabling selective release of high-confidence mappings (Section 6.5.3).

6.5.1 Mapping Quality

Before examining quantitative patterns, we assess mapping quality through concrete examples. This qualitative analysis grounds the abstract notion of "semantic equivalence" in tangible cases and reveals failure modes that aggregate metrics might obscure.

Question Mapping Across Methods

Table 6.8 compares how each declassification method transforms questions across all four datasets. The examples reveal systematic differences: Gen-PR rewrites questions semantically, sometimes corrupting specific details (the book title "*A Biography of Margaret Thatcher*" becomes "*Margaret Thatcher's Life Story*"). Gen-BT preserves named entities through round-trip translation but provides minimal lexical variation. Map_{RR} retrieves genuinely different questions that preserve semantic intent, note how "How are glacier caves formed?" maps to "What are glacial caves formed from?", a distinct question with equivalent meaning.

The contrast with SimpleQA is stark. While other datasets show clean mappings, SimpleQA's deliberately obscure questions have no semantic equivalents in QUADRo. The original asks for a Prime Minister with a specific biographical constraint (cabin crew until 1971); the mapped question loses this constraint entirely. Similarly, a question about a specific musical work's dedicatee becomes a generic query about the composer.

Answer Reconstruction

For AS2 datasets, declassification must reconstruct not only questions but also answer candidates. This is challenging because the reconstructed answers must satisfy two constraints: (1) they must be retrievable from a public corpus, and (2) they must preserve the correct/incorrect distinction relative to the mapped question. The LLM annotator judges each candidate against the mapped question, assigning labels that may differ from the original when the semantic relationship changes.

Tables 6.9 and 6.10 show complete examples from WikiQA and TrecQA respectively, illustrating how reconstruction works across different domains.

The WikiQA example demonstrates the ideal case. The original positive explains glacier cave formation; the reconstructed positive provides a more detailed explanation of the same

Table 6.8: Question mapping comparison across datasets and methods. Map_{RR} retrieves semantically equivalent questions for standard datasets but loses critical specificity on SimpleQA. Gen-PR occasionally corrupts details; Gen-BT preserves entities but adds little variation.

Method	**Question**
WikiQA	
Original	How are glacier caves formed?
Map_{RR}	What are glacial caves formed from?
Gen-PR	What is the process behind the creation of ice caverns within glaciers?
Gen-BT	How do glacier caverns form?
TrecQA	
Original	Who is the author of "The Iron Lady: A Biography of Margaret Thatcher"?
Map_{RR}	Who wrote the biography "The Iron Lady"?
Gen-PR	Can you identify the writer of "The Iron Lady: Margaret Thatcher's Life Story"?
Gen-BT	Who is the author of "The Iron Lady: A Biography of Margaret Thatcher"?
OpenBookQA	
Original	The main component in dirt is
Map_{RR}	Main component of dirt?
Gen-PR	What is the primary element found in soil?
Gen-BT	The main component of the soil is
Original	A Mola Mola might live where?
Map_{RR}	What sort of habitat does a Moa live in? *[entity confusion]*
Gen-PR	In what type of environment would a Mola Mola typically reside?
Gen-BT	Where could a Mola Mola live?
SimpleQA	
Original	What is the name of the former Prime Minister of Iceland who worked as a cabin crew member until 1971?
Map_{RR}	Who was the Prime Minister of Iceland? *[constraint stripped]*
Gen-PR	Which ex-Icelandic Prime Minister was employed as flight attendant through 1971?
Gen-BT	What is the name of the former Prime Minister of Iceland, who worked as a cabin crew member until 1971?
Original	To whom did György Ligeti dedicate his piano étude "Désordre"?
Map_{RR}	Who was György Ligeti? *[question changed entirely]*
Gen-PR	Who was the dedicatee of György Ligeti's piano étude "Désordre"?
Gen-BT	To whom did György Ligeti dedicate his piano étude "Désordre"?

Table 6.9: WikiQA answer reconstruction example. The encyclopedic domain allows clean reconstruction: both positive and negative answers preserve their semantic relationship to the question using entirely different text from Wikipedia.

Original	
Question	How are glacier caves formed?
Positive	A glacier cave is a cave formed within the ice of a glacier.
Negative	Glacier caves are often called ice caves, but this term is properly used to describe bedrock caves that contain year-round ice.
Declassified ($\mathbf{Map}_{RR}$)	
Question	What are glacial caves formed from?
Positive	Glacier caves are formed by melting ice and flowing water within and under glaciers. The cavities are influenced by the very slow flow of the ice, which tends to collapse the caves again.
Negative	Glacier caves are frequently referred to as ice caves, though technically, the term "ice cave" applies to bedrock caves that maintain ice throughout the year.

phenomenon. The original negative clarifies terminology (ice caves vs glacier caves); the reconstructed negative preserves this distinction using different phrasing. Both answers are factually accurate and maintain the same semantic relationship to the question.

Table 6.10: TrecQA answer reconstruction example. The news domain requires finding passages with specific facts; the reconstructed negative shows acceptable drift from player statistics to umpire career.

Original	
Question	How many consecutive baseball games did Lou Gehrig play?
Positive	Fifty years ago today: Lou Gehrig of the New York Yankees did not play against the Detroit Tigers at Briggs Stadium, thus ending his streak of 2,130 consecutive games played.
Negative	In the World Series that year, he started three games for the Red Sox.
Declassified ($\mathbf{Map}_{RR}$)	
Question	How many consecutive baseball games has Lou Gehrig played?
Positive	There were no page-topping headlines in newspapers around the world announcing that on May 2, 1939, Lou Gehrig of the New York Yankees did not play against the Detroit Tigers at Briggs Stadium, ending his streak of 2,130 consecutive games played.
Negative	He umpired in four World Series, seven American League Championship Series and four All-Star Games.

The TrecQA example illustrates news-domain challenges. The positive answer must contain the specific fact (2,130 consecutive games), which our CCNews retrieval successfully finds in a different article about the same event. The negative answer shows *acceptable semantic drift*: the original discussed a Red Sox pitcher's World Series appearances, while the reconstructed negative discusses an umpire's career statistics. Both are correctly labeled as not answering the Lou Gehrig question, what matters is that the positive/negative distinction is preserved, not that the negative answers are semantically similar.

When reconstruction cannot find a candidate matching the original label, the pipeline falls back to the most similar candidate with the LLM's assigned label. This can result in label changes: for example, an original positive answer about Australian tablespoon measurements

(20 mL) might be replaced by a South African variant with incorrect information (15 mL), which the LLM correctly labels as negative. These fallback cases are rare but can shift label distributions, particularly for specific questions where appropriate answers are rare in the corpus.

Quantitative Assessment

To quantify mapping quality systematically, we measure semantic similarity between original and mapped questions (Q sim.) and between original and reconstructed answer sentences (S sim.) using MPNet-base[2]. We also conduct manual assessment of question equivalence on 200 question pairs (50 per dataset), where annotators judged whether two questions share the same information-seeking intent.

Table 6.11: Semantic similarity and manual equivalence assessment across datasets and methods. Map_{RR} achieves high equivalence (90–94%) on standard datasets but only 16% on SimpleQA.

Dataset	Method	Q sim. (%)	S sim. (%)	Q equiv. (%)
WikiQA	Gen-PR	76.5	63.3	90.0
	Gen-BT	84.7	74.2	92.0
	Map_{RR}	84.6	72.3	94.0
TrecQA	Gen-PR	75.8	60.3	86.0
	Gen-BT	80.4	62.5	88.0
	Map_{RR}	79.5	64.6	90.0
OpenBookQA	Gen-PR	78.5	62.1	90.0
	Gen-BT	91.0	75.4	94.0
	Map_{RR}	91.2	67.6	94.0
SimpleQA	Gen-PR	84.4	77.3	92.0
	Gen-BT	88.4	80.0	94.0
	Map_{RR}	51.1	55.7	16.0

The results in Table 6.11 confirm the qualitative observations. Across WikiQA, TrecQA, and OpenBookQA, Map_{RR} achieves equivalence rates of 90–94%, matching or exceeding the generative baselines. This supports the hypothesis that retrieval-based declassification can find genuinely equivalent questions when they exist in the public corpus. The contrast with SimpleQA is dramatic: only 16% of mapped questions are judged equivalent, explaining the large difficulty shifts observed in Section 6.4.1.

Interestingly, generative methods maintain high equivalence on SimpleQA (92–94%) precisely because they paraphrase rather than replace. This confirms that SimpleQA's difficulty shift under Map_{RR} stems from mapping to *different* questions, not from any deficiency in the retrieval mechanism itself.

6.5.2 SimpleQA: Difficulty Preservation Analysis

SimpleQA provides a stress test for declassification: it targets obscure, entity-centric facts designed to probe LLM knowledge boundaries. Unlike OpenBookQA's near-zero deltas, SimpleQA shows +5.8 to +12 point improvements (Table 6.4), indicating that mapped questions are systematically easier.

[2] `sentence-transformers/paraphrase-mpnet-base-v2`

Root Cause: Distributional Mismatch

SimpleQA was deliberately constructed to be "uncontaminated": the authors targeted complex, niche facts unlikely to appear in LLM training data. Questions are designed with multiple nested constraints that uniquely identify obscure facts, for example, asking not just about a Prime Minister, but about one with a specific biographical detail; not just about an award, but about a specific recipient in a specific year. This design philosophy directly conflicts with the assumptions underlying retrieval-based declassification.

QUADRo, conversely, contains real questions from public sources, questions that users actually ask in practice. Because these questions are publicly available, they naturally reflect common information needs: popular entities, frequently-discussed topics, and widely-known facts. When someone wants to know about Iceland's Prime Minister, they ask "*Who was the Prime Minister of Iceland?*", not "*What is the name of the former Prime Minister of Iceland who worked as a cabin crew member until 1971?*". The specificity that makes SimpleQA challenging for LLMs is precisely what makes it unmappable to public question databases.

This distributional mismatch manifests in systematic failure modes, illustrated in Table 6.12:

Table 6.12: Low-similarity mapping examples from SimpleQA showing systematic failure modes. Mappings lose critical specificity that makes the original questions challenging.

	Question
Constraint stripping: specific constraints removed entirely	
Original	What is the name of the former Prime Minister of Iceland who worked as a cabin crew member until 1971?
Map_{RR}	Who was the Prime Minister of Iceland?
Original	How many corners did Barcelona take in the Champions League semi-final match between Barcelona and Milan on April 27, 2006?
Map_{RR}	What year was Barcelona eliminated from the Champions League?
Entity substitution: obscure entities replaced with common ones	
Original	In what year was American chemist Eger Vaughan Murphree awarded the Perkin Medal?
Map_{RR}	What year was Dr. Drew O'Donnell awarded?
Original	What year did the Lego part with ID gal56 first release?
Map_{RR}	When was the first Lego set released?
Constraint dilution: multiple constraints reduced to generic query	
Original	What is the name of the university from which Thabo Cecil Makgoba first graduated with a PhD degree in 2009?
Map_{RR}	What is the name of the university in South Africa?
Original	In the lore of Dungeons and Dragons, what is the name of the fortress in the Astral Plane used as a lair by the red great wyrm Ashardalon?
Map_{RR}	What is the name of the fortress in Dungeons and Dragons?

The examples reveal three failure modes. *Constraint stripping* removes specific constraints that uniquely identify answers: the Iceland PM question loses "cabin crew member until 1971"; the Barcelona question loses the specific match, date, and statistic (corners). *Entity substitution* replaces obscure entities with common ones: Eger Murphree becomes Dr. Drew

O'Donnell; a specific Lego part ID becomes generic. *Constraint dilution* reduces multiple nested constraints to generic queries: the D&D question loses "Astral Plane," "red great wyrm," and "Ashardalon," leaving only "fortress in Dungeons and Dragons."

Stratified Analysis: Reranking is Essential

To test whether high-quality mappings can preserve difficulty (RQ2, RQ4), we stratify results by semantic similarity. We use MPNet[3] to compute similarity between original and mapped questions, select the top 1% most similar mappings (n=43 questions), and evaluate accuracy on both the mapped questions and their corresponding 43 original questions from SimpleQA. By comparing the same questions before and after mapping, we isolate the effect of question transformation from the effect of question selection.

Table 6.13 compares Map_R and Map_{RR} on these top 1% mappings. The contrast is striking: on the same 43 questions, Map_R shows large positive deltas for all models (ranging from +9 to +15 points), indicating systematic difficulty reduction. In contrast, Map_{RR} achieves small deltas for three models ($|\Delta| \leq 2.3$), with only Phi-4 showing a larger shift of +7.2. The deltas of $|\Delta| \leq 2.3$ represent a good difficulty preservation, comparable to the $|\Delta| \leq 2$ observed on AS2 and OpenBookQA.

Table 6.13: SimpleQA top 1% stratified analysis comparing Map_R vs Map_{RR} on the same 43 questions (top 1% by MPNet similarity). Original accuracy is computed on the corresponding source questions; deltas measure the shift from original to mapped.

Model	Original (%)	Map_R Δ	Map_{RR} Δ
Phi-3	16.28	+9.30	**−2.3**
Ministral	11.63	+11.01	**−2.3**
Phi-4	23.25	+15.58	+7.2
Qwen-3	30.23	+13.01	**−2.1**

Why does reranking help so much on the same questions? The bi-encoder retrieves candidates based on surface-level semantic overlap, often selecting questions that share vocabulary without sharing intent. The cross-encoder reranker processes question pairs jointly with answer and promotes candidates with genuine semantic equivalence. For SimpleQA's highly specific questions, this refinement is critical.

Table 6.14 shows what these high-quality mappings look like. The first three examples show near-perfect preservation: Cab Calloway's scat singing teacher, the Soviet Union's second largest republic, and Hetty King's half-sister all map to essentially identical questions. These succeed because they ask about topics that real users also ask about, such as musical history, geopolitics, and entertainment figures. The last two examples show slight simplification but preserve the key constraints: "Kashmir" + "Afghan period" and "biomarkers" + "osteoarthritis" remain intact. The common pattern is that these questions, while specific, target domains with genuine public interest rather than artificially constructed obscurity.

Key Findings

Reranking is essential for difficulty preservation (RQ2). For test set declassification where difficulty preservation matters, reranking is not optional. It is the mechanism that produces high-quality mappings. On the same 43 questions, Map_R systematically reduces

[3] `sentence-transformers/paraphrase-mpnet-base-v2`

Table 6.14: High-similarity mappings from SimpleQA (top 1% by MPNet similarity). These preserve critical constraints because equivalent questions exist in QUADRo.

	Question
Original	Who taught Cab Calloway how to sing in the scat style?
Map_{RR}	Who taught Cab Calloway to sing in the scat style?
Original	Which was the second largest republic in the Soviet Union?
Map_{RR}	What is the second largest republic of the Soviet Union?
Original	What was the name of male impersonator Hetty King's half-sister?
Map_{RR}	What was the name of the half-sister of music hall artist Hetty King?
Original	Name the traditional dance in Kashmir where a male dancer accompanies the chhakri singers and was introduced during the Afghan period.
Map_{RR}	What is traditional dance in Kashmir introduced in Afghan period?
Original	What year were the guidelines outlining requirements for the inclusion of soluble biomarkers in osteoarthritis clinical trials published?
Map_{RR}	When were the guidelines for inclusion of soluble biomarkers in osteoarthritis trials published?

difficulty ($\Delta > +9$ for all models) while Map_{RR} preserves it ($|\Delta| \leq 2.3$ for three of four models), matching the fidelity observed on standard benchmarks. This demonstrates that the cross-encoder's ability to detect subtle semantic distinctions translates directly into better difficulty preservation.

Phi-4 remains an outlier with $\Delta = +7.2$, substantially higher than the other models. This suggests model-specific sensitivity: even when the mapped question is semantically equivalent, Phi-4's performance shifts more than other models. Whether this reflects differences in training data, model architecture, or evaluation dynamics remains an open question, but it highlights that difficulty preservation may vary across models even for high-quality mappings.

Selective Declassification via Similarity Thresholds

The previous analysis showed that reranking enables difficulty preservation on high-quality mappings. A natural question follows: can organizations use similarity scores to selectively release only well-mapped questions? This would trade coverage for fidelity, releasing a smaller but more reliable declassified benchmark.

To test this, we filter SimpleQA mappings by reranking similarity threshold and measure accuracy deltas at each level. For each threshold, we select mapped questions with similarity above that value, then compute accuracy on both these mapped questions and their corresponding original questions. Table 6.15 shows the results.

At the highest threshold ($\geq$95.0), all four models achieve $\Delta = 0$ on the 16 qualifying questions. At $\geq$90.0, three models maintain $\Delta = 0$ while Phi-4 shows a small shift. As the threshold decreases, more questions qualify but fidelity degrades: Phi-4 in particular shows increasing difficulty shift at lower thresholds. This confirms that similarity scores are predictive of declassification success, and organizations can use them to make informed coverage-fidelity tradeoffs.

Key Findings.

Similarity scores enable selective declassification (RQ4). For benchmarks where difficulty preservation is critical, organizations can apply similarity thresholds to release only

Table 6.15: SimpleQA accuracy deltas (Δ) filtered by reranking similarity thresholds. For each threshold, we compare accuracy on mapped questions vs. their corresponding originals. Higher thresholds yield smaller subsets but better fidelity.

Model	**Δ by Similarity Threshold**				
	≥95.0	≥90.0	≥85.0	≥80.0	≥75.0
Phi-3	0.0	0.0	0.0	−2.5	−2.2
Ministral	0.0	0.0	−2.8	−2.5	−2.2
Phi-4	0.0	+3.7	+8.3	+7.5	+7.0
Qwen-3	0.0	0.0	0.0	−2.5	−1.8
# Questions	16	27	36	40	46

questions with high-quality mappings. At the 95% threshold, all models achieve perfect fidelity ($\Delta = 0$), though coverage is limited to 16 questions. At 90%, coverage increases to 27 questions while three of four models maintain $\Delta = 0$. This provides a practical mechanism for partial benchmark release: rather than releasing an entire declassified benchmark with variable quality, organizations can release a curated subset where every question has been verified to map well. The similarity score serves as an automatic quality filter, reducing the need for manual verification.

The fundamental limitation is distributional mismatch (RQ3). Declassification works well for standard benchmarks where most questions have equivalents in QUADRo. It fails for adversarially-constructed benchmarks where deliberately obscure questions have no natural equivalents. The stratified analysis shows that when high-quality mappings exist, even SimpleQA can achieve $\Delta = 0$. The challenge is that such mappings are rare: only 16 questions meet the 95% threshold out of 4,326 total, less than 0.4% of the dataset. This is not a failure of the retrieval system but a fundamental property of SimpleQA's design: it specifically targets questions that users would not naturally ask, which means by construction they cannot exist in databases of real user questions. For standard benchmarks like WikiQA and TrecQA, the vast majority of questions map well because they reflect genuine information needs that appear across multiple sources.

6.5.3 Dataset Size and Sampling Strategy

The previous section showed that similarity scores enable selective test set release by filtering low-quality mappings. The same principle applies to training data: rather than using the entire declassified training set, organizations can prioritize high-similarity examples. This raises a practical question: when only a fraction of the data maps well, is it better to use all available mappings (including low-quality ones) or to be selective?

We evaluate two sampling strategies on WikiQA: *Random (R)* selects a random X% of training examples, while *Similarity (S)* selects the top X% by question mapping similarity score. We focus on WikiQA because its larger training set (2,118 questions) allows meaningful analysis at small percentages. TrecQA's 94 training questions would yield only ∼9 questions at 10%. Table 6.16 reports results using Map_{RR}, and Figure 6.3 visualizes the comparison.

Table 6.16: Effect of dataset size and sampling strategy on WikiQA training utility. R = random sampling, S = similarity-based sampling. Values are ΔP@1 relative to original baseline (DeBERTa: 81.43, MiniLM: 70.60).

		Percentage of Training Data				
Model	**Strat**	**10%**	**30%**	**50%**	**70%**	**90%**
DeBERTa-v3	R	$-20.0_{\pm 6.3}$	$-9.0_{\pm 2.2}$	$-7.1_{\pm 2.6}$	$-3.7_{\pm 2.4}$	$-1.4_{\pm 0.4}$
	S	$-18.6_{\pm 1.6}$	$\mathbf{-4.1_{\pm 2.9}}$	$-2.1_{\pm 1.4}$	$\mathbf{0.0_{\pm 0.6}}$	$\mathbf{0.0_{\pm 1.6}}$
MiniLM-L12	R	$-22.8_{\pm 1.7}$	$-15.2_{\pm 4.4}$	$-9.1_{\pm 0.9}$	$-7.9_{\pm 2.2}$	$-2.0_{\pm 2.9}$
	S	$-20.7_{\pm 2.4}$	$\mathbf{-11.1_{\pm 0.9}}$	$-9.4_{\pm 3.6}$	$-4.4_{\pm 3.3}$	$\mathbf{-1.1_{\pm 4.8}}$

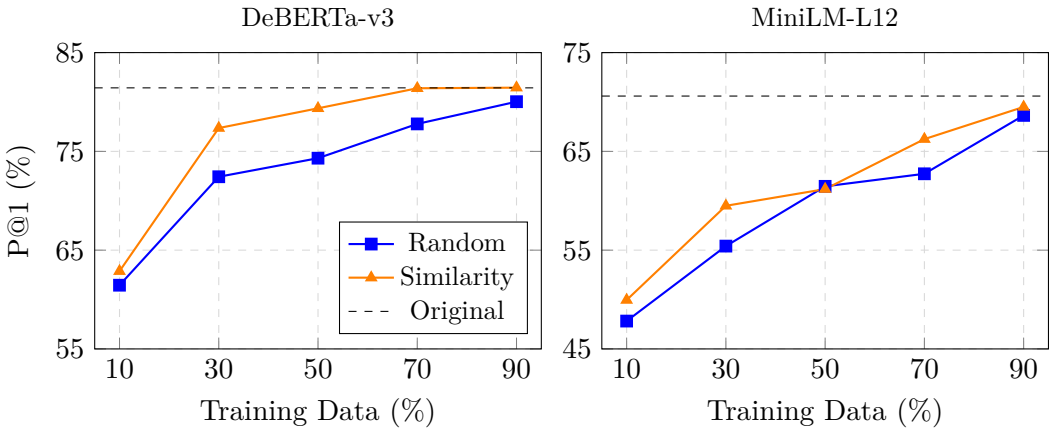

Figure 6.3: Effect of sampling strategy on training utility with Map_{RR}. Random sampling (blue) vs similarity-based sampling (orange). Dashed line indicates original baseline. DeBERTa (left) reaches baseline with similarity sampling at 70%; MiniLM (right) shows a persistent gap but similarity sampling consistently outperforms random at all percentages.

Key Findings.

Similarity-based selection outperforms random sampling (RQ4). Just as similarity thresholds enable selective test set release, similarity-based sampling improves training utility. At 30%, the gap is substantial: similarity sampling achieves $\Delta = -4.1$ on DeBERTa compared to $\Delta = -9.0$ for random sampling, a difference of nearly 5 points. Even at 10%, where both strategies struggle due to insufficient data, similarity sampling shows lower variance (± 1.6 vs ± 6.3), indicating more reliable and stable performance. This confirms that similarity scores are predictive of mapping quality across both evaluation and training scenarios: high-similarity mappings not only preserve test difficulty but also provide more effective training signal.

High-quality mappings compensate for reduced data (RQ4). With similarity sampling, 70% of the data achieves the same performance as 100% for DeBERTa ($\Delta = 0.0$), and 90% achieves near-parity for MiniLM ($\Delta = -1.1$). This has practical implications: organizations with limited high-quality mappings can still achieve effective declassification by being selective rather than exhaustive. The effect is stronger for larger models, with DeBERTa showing a 4.9-point advantage from similarity sampling at 30% compared to 4.1 points for MiniLM. This is consistent with our earlier finding that larger models are more robust to distributional shifts.

Taken together, these results suggest that quality matters more than quantity: a smaller set of well-mapped examples outperforms a larger set of mixed-quality mappings.

6.6 Discussion

6.6.1 Broader Implications

The experimental results validate the core premise introduced at the beginning of this chapter: semantic mapping can transform sensitive content into public equivalents while preserving task-relevant properties. This has concrete implications for the scenarios outlined at the beginning of the chapter.

For organizations facing the challenges of Scenario A (dataset and model release), declassification provides actionable solutions. The benchmark contamination problem, where public evaluation sets are memorized by large language models, can be addressed through shadow benchmarks: organizations release declassified versions publicly while retaining originals for authoritative evaluation. Our results demonstrate this is practical: AS2 test set declassification achieves $|\Delta| \leq 2$ points, and OpenBookQA preserves model rankings with $|\Delta| \leq 0.4$. Similarly, model memorization risks are eliminated by construction: if a model trained on declassified data memorizes examples, it memorizes public content rather than customer data. This enables safer model deployment and reduces exposure to extraction attacks.

For organizations in Scenario B (privacy-preserving training), declassification enables previously impossible workflows. Customer support teams can build QA systems that learn from real interaction patterns without exposing ticket contents. Healthcare and legal organizations can develop domain-specific models without violating HIPAA, GDPR, or attorney-client privilege. Multi-tenant SaaS providers can improve their products using cross-customer patterns without creating data isolation violations. In each case, the model learns from public proxies that reflect authentic question-answer distributions.

The regulatory implications are particularly timely. As AI regulations such as the EU AI Act increasingly require transparency about training data properties, declassified datasets provide auditable proxies: regulators can inspect declassified versions to verify distributional properties, topic coverage, and potential biases without accessing the underlying customer data. This addresses a genuine tension where organizations must demonstrate compliance without compromising confidentiality.

Finally, declassification enables academic collaboration that proprietary constraints typically prevent. Organizations can publish research and share datasets with universities by releasing shadow versions that support reproduction of experimental results. This bridges the gap between industrial practice and open science without requiring data exposure.

6.6.2 Limitations

Several limitations constrain the framework's applicability. First, when the pipeline cannot find a corpus sentence matching the original label, it falls back to the most similar candidate with the LLM-assigned label. This can shift label distributions, particularly for specific questions where appropriate answers are rare in the corpus.

Second, effectiveness depends on overlap between proprietary question distributions and public retrieval corpora. WikiQA, TrecQA, and OpenBookQA succeed because they contain common questions well-covered by QUADRo. SimpleQA partially fails because it deliberately

targets unusual questions. This is not a bug but a fundamental property: declassification cannot create public equivalents for content that has no public equivalent.

Third, specialized domains require appropriate corpora. Medical QA would need PubMed, legal QA would need case law databases, and so on. The current pipeline targets sentence-level answers; other formats such as span extraction or long-form generation would require format-specific reconstruction strategies.

Finally, all experiments use English. Multilingual declassification would require both multilingual QUADRo coverage and appropriate corpora in target languages, presenting both engineering and coverage challenges.

From an ethical perspective, declassification is intended to enable beneficial data sharing while protecting privacy. However, the same techniques could potentially be misused to circumvent legitimate data access restrictions. We emphasize that declassification should complement, not replace, proper data governance practices. Organizations should verify that declassification satisfies their specific legal and ethical obligations before using it for regulatory compliance.

Scope of privacy protection and threat model. Declassification is designed for settings where the underlying knowledge queried by users is already publicly available (e.g., in search engines, voice assistants, or LLM-based systems), and what is proprietary is the *distribution* of user interactions rather than the semantic content itself. In settings where the concepts themselves are private, such as internal procedures, proprietary products, or confidential entities with no public equivalent, declassification cannot apply because no semantically equivalent public content exists by definition. Organizations should assess whether their data falls into the former regime before applying the framework.

Within this scope, declassification provides content-level protection by construction: the released dataset contains only pre-existing public content. However, the framework does not provide formal differential privacy guarantees (Dwork et al., 2006). Several residual risks should be considered. First, the semantic mapping may leak structural information about the proprietary dataset: the distribution of question topics and the relative frequency of domain-specific queries could be inferred from the declassified version. Second, the mapping itself reveals which public questions were selected as replacements, potentially narrowing the space of plausible original questions for highly distinctive queries. Third, mapping private questions to more common public equivalents may introduce contamination asymmetry: if only weak semantic matches are available, a previously unseen private question may be mapped to a formulation that plausibly appears in LLM pretraining data, leading to artificially higher performance on the declassified version. When strong semantic matches are available, this asymmetry is negligible, as both versions are equally likely to appear in public corpora.

These risks are inherent to any content-replacement approach and should be evaluated against organizational threat models. For applications requiring formal guarantees, declassification could be combined with differential privacy mechanisms applied to the mapping distribution, though this remains future work (Chapter 7).

6.7 Conclusion

This chapter presented dataset declassification, a framework that enables releasing semantically equivalent versions of proprietary QA datasets without exposing original content. The approach combines question retrieval through QUADRo, optional reranking through QRP

pre-trained models, and a novel answer reconstruction pipeline using domain-matched corpus retrieval with LLM-based annotation.

The experimental results validate the framework's effectiveness for both training and evaluation. On WikiQA, models trained on fully declassified data achieve performance identical to models trained on original data ($\Delta \approx 0$). On TrecQA, using CCNews as the news-domain corpus achieves $|\Delta| \leq 1.2$ points. Test set declassification also succeeds: models evaluated on declassified test sets show $|\Delta| \leq 2$ points across all configurations, enabling release of complete "shadow" benchmarks. For LLM evaluation, OpenBookQA declassification preserves model rankings with $|\Delta| \leq 0.4$ points.

The contrast with generative baselines is instructive: Gen-PR loses 29 points on TrecQA while Gen-BT loses 9 points on WikiQA, and neither works universally. Retrieval-based declassification achieves near-zero deltas on both datasets.

The SimpleQA experiments reveal important boundary conditions. Declassification shows +5.8 to +12 point difficulty shifts on this adversarially-constructed benchmark. However, the stratified analysis shows that when the reranker identifies high-quality mappings, even SimpleQA can achieve $\Delta \approx 0$. This establishes that cross-encoder reranking, not bi-encoder retrieval, is the mechanism enabling difficulty-preserving declassification.

At a minimum, declassification provides a concrete and verifiable guarantee: it identifies portions of an internal dataset for which sufficiently similar public content exists, and enables their release in full. Independently of downstream utility preservation, this replaces organizational uncertainty with an explicit, auditable notion of public overlap, allowing companies to safely release data that would otherwise remain inaccessible.

Beyond the technical contributions, declassification addresses the practical challenges outlined at the beginning of this chapter. For organizations in Scenario A, it enables shadow benchmark release without contamination risk and eliminates model memorization concerns by construction. For organizations in Scenario B, it unlocks privacy-preserving training workflows: customer support teams can learn from interaction patterns, healthcare organizations can build domain-specific models without HIPAA violations, and multi-tenant platforms can improve their products without cross-customer data leakage. As regulations increasingly demand training data transparency while organizations develop valuable proprietary datasets, the tension between openness and protection will only intensify. Declassification offers a principled path through this tension.

6.7.1 Future Directions

This work opens several directions for future research. First, while declassification is effective for capturing population-level behavior, handling tail and highly specific questions remains an open problem. Developing specialized declassification strategies for such cases, such as hybrid retrieval-generation methods or domain-specific public resources, could extend applicability beyond the head of the distribution.

Second, extending declassification beyond questions and answers to supporting context represents an important direction. Many QA tasks include passages, conversation history, or other contextual information that also requires declassification. Developing methods to preserve contextual equivalence while replacing sensitive content would enable release of richer grounded QA datasets.

Third, our evaluation is necessarily limited to public benchmarks. An important next step is large-scale validation on proprietary industrial datasets, measuring declassification quality,

coverage, and utility under real governance constraints. Such studies would also enable deeper analysis of distributional alignment between customer data and available public corpora.

Fourth, multilingual declassification presents both opportunities and challenges. Extending QUADRo coverage to additional languages and identifying appropriate answer corpora for each language-domain combination would enable declassification for global enterprises with diverse user populations.

Finally, the interaction between declassification and other privacy-preserving techniques warrants investigation. While declassification provides privacy by construction (only public content is used), combining it with formal privacy guarantees, controlled generation, or distillation could address edge cases where even semantic similarity to private questions poses risks.

Chapter 7

Conclusions

In this PhD thesis, we have investigated question semantic equivalence at three levels of granularity: pairwise equivalence for question retrieval, cluster-level equivalence for coherence analysis, and dataset-level equivalence for privacy-preserving transformation.

The thesis is organized around a coherent research agenda centered on *semantic equivalence*: the relation that holds between questions seeking the same information. This seemingly simple relation proves to be surprisingly rich when examined at different scales. We begin by creating the infrastructure for large-scale question retrieval, then develop specialized pre-training methods to reduce annotation requirements and improve ranking performances, extend the analysis to question clusters for coherence optimization in both LLMs and retrieval systems, and finally demonstrate that semantic equivalence enables privacy-preserving dataset transformation. This progression from pairs to clusters to datasets reveals that question equivalence is not just a simple binary relation between strings but a multi-scale phenomenon with different implications at each level.

The research demonstrates that question understanding, when approached systematically through the lens of equivalence, yields both theoretical insights and practical tools. Theoretical insights include the discovery that coherence and accuracy are partially independent properties (Section 5.2.2), that understanding gaps can be as limiting as knowledge gaps (Section 5.2.3), and that semantic structure can be learned implicitly through carefully designed self-supervised objectives (Section 4.1). Practical tools include datasets (Sections 3.2–3.4), pre-training methods (Section 4.1), inference-time augmentation techniques (Section 5.2.3), and privacy-preserving transformation pipelines (Section 6.2) that address real challenges in building reliable QA systems.

7.1 Overall Contributions

The contributions span infrastructure development, algorithmic innovation, empirical analysis, and practical application. Each addresses a specific research gap while building toward a unified understanding of question equivalence and its utility for QA systems:

- **Question Retrieval Infrastructure (QUADRo and QRC):** In Chapter 3, we present QUADRo, a comprehensive Database-based QA framework operating over 6.3 million question-answer pairs aggregated from diverse high-quality sources including Natural Questions, GooAQ, and WikiAnswers. We also introduce the Question Ranking Corpus (QRC), containing 15,211 queries with $\approx 443K$ annotated examples, the

first large-scale resource specifically designed for question retrieval with answer-aware annotations. Key findings show that including the answer at retrieval time improves accuracy (+5 P@1 over answer-agnostic methods). The QAQ input ordering proved optimal, suggesting that models benefit from first analyzing query-answer compatibility before verifying question equivalence.

- **Specialized Pre-Training (QRP):** In Chapter 4, we present Question Ranking Pre-training, a self-supervised method that trains models to detect ranking corruptions without access to the original query. By removing the query, QRP forces models to learn question equivalence patterns from the relationships between retrieved QA pairs candidates. This approach achieves statistically significant improvements (+1.05% P@1, $p = 0.0005$) while reducing model variance by over 50%, improving reliability for both research reproducibility and production deployment. Transfer experiments demonstrate benefits across datasets (+2.99% P@1 on SemEval in zero-shot transfer), confirming that the learned representations capture general properties of question equivalence.

- **LLM Coherence Analysis and q-RAG:** In Chapter 5, we present the first broad analysis of LLM coherence on factual QA, evaluating multiple model families, including GPT-OSS, LLaMA, and Mistral, across multiple sizes on clusters of equivalent questions. The analysis reveals significant coherence gaps where models succeed on some phrasings but fail on semantically equivalent alternatives. Crucially, these failures cannot be attributed to knowledge gaps: the models demonstrably possess the required information yet fail to access it consistently.

  To address this, we introduce Question-Augmented Generation (q-RAG), which improves accuracy up to 9 percentage points and coherence up to 28 points by supplementing prompts with retrieved similar questions. The approach is based on the insight that redundant semantic signal helps models triangulate user intent. Remarkably, q-RAG outperforms document-based RAG despite providing no new factual information, demonstrating that understanding failures rather than knowledge gaps often limit LLM performance.

  Preliminary experiments further demonstrate that q-RAG's benefits can be distilled into model parameters through DPO and SFT training, achieving +3.93 EM and +24.0 coherence points on zeroshot cross-dataset evaluation (Section 5.4).

  The multilingual extension across six typologically diverse languages reveals both universal patterns (coherence decreases with question difficulty across all languages) and language-specific effects in coherence behavior, providing guidance for multilingual QA deployment.

- **Coherence Ranking Loss:** Also in Chapter 5, we introduce a training objective for retrieval models that combines Query Embedding Alignment (QEA) and Similarity Margin Consistency (SMC). Critically, using either component alone decreases coherence; only their combination succeeds. The CR Loss improves coherence by up to 30% (measured by Rank-Biased Overlap) while simultaneously improving relevance by up to 1.69% NDCG, demonstrating that coherence and accuracy are complementary rather than competing objectives. Downstream benefits include +9.3% improvement in reranking opportunity for retrieve-and-rerank pipelines.

- **Dataset Declassification Framework:** Chapter 6 introduces a framework for privacy-preserving dataset transformation. The approach replaces proprietary questions with semantically equivalent public questions retrieved using QUADRo, and reconstructs answers from domain-matched public corpora such as Wikipedia and CCNews. The framework is motivated by two practical use cases: enabling organizations to share high-value datasets without revealing sensitive content, and allowing the use of proprietary data for model training when there are existing constraints in terms of privacy or regulatory requirements.

  Models trained on fully declassified data match baseline performance: WikiQA achieves $\Delta \approx 0$ and TrecQA $|\Delta| \leq 1.2$ points. For test set declassification, OpenBookQA preserves model rankings with $|\Delta| \leq 0.4$ points, enabling safe dataset sharing and benchmark decontamination. The framework validates that dataset equivalence is achievable when domain matching between source and replacement content is maintained, while the SimpleQA experiments identify limitations when proprietary questions fall outside public corpus coverage.

In addition to the published work, this thesis offers further insights. It presents a unifying perspective on question equivalence across multiple scales: the retrieval infrastructure (Section 3.3) enables both coherence analysis (Section 5.1) and dataset declassification (Section 6.2), while the coherence framework developed for LLMs transfers directly to retrieval systems (Sections 5.2–5.6). The thesis draws a clear distinction between knowledge failures and understanding failures (Section 5.2.3), providing evidence that the latter are more common than typically assumed. It further analyzes the central role of domain alignment in dataset transformation (Section 6.5), and reflects on different notions of equivalence, functional, training, and difficulty-based, with implications for future research.

Overall, the technical contributions advance the state of the art in question retrieval, pre-training, coherence optimization, and dataset transformation. But perhaps more importantly, they collectively demonstrate that semantic equivalence provides a powerful organizing principle for understanding and improving QA systems.

7.2 A Unified Perspective

Three themes connect the contributions across chapters, revealing deeper principles about question understanding that extend beyond the specific technical contributions. These themes emerged gradually through the research process and now provide a conceptual framework for understanding how the individual contributions relate to each other and to broader questions in NLP.

7.2.1 From Pairs to Clusters to Datasets

The thesis follows a progression in the granularity of equivalence that reflects the evolution of the research. Chapter 3 focuses on question pairs, addressing the task of retrieving equivalent questions given a query to answering it. The retrieval model learns to identify when two questions express the same information-seeking intent, which makes it possible to transfer answers from previously seen questions to new ones.

Chapter 4 implicitly learns cluster structure through ranking perturbation detection. When retrieving candidates for a query, the returned questions form a cluster around the

same information need, and by learning to recognize this structure without seeing the cluster center, which is the query, models develop robust representations of equivalence. Chapter 5 makes clusters explicit, using them to define coherence constraints for both LLMs and retrieval models. A cluster of equivalent questions is more informative than any individual question, because variation in phrasing helps to disambiguate the underlying intent. Chapter 6 extends this observation to entire datasets, showing that mapping each proprietary question to a semantically equivalent public alternative results in dataset-level equivalence. This enables content replacement while preserving the aggregate statistical properties that determine training utility.

This progression reveals that each level of equivalence enables different applications:

- **Pairwise equivalence** enables answer reuse: if questions are equivalent, an answer to one is an answer to both.
- **Cluster equivalence** enables coherence optimization: a system should behave consistently across all members of a cluster.
- **Dataset equivalence** enables privacy-preserving transformation: replacing questions with equivalents preserves the dataset's utility for training and evaluation.

Each level builds on the previous one while introducing additional considerations. At the level of pairwise equivalence, equivalence is symmetric and transitive. At the level of cluster equivalence, statistical properties become relevant, including the distribution of alternative phrasings and the coverage of linguistic variation. At the level of dataset equivalence, aggregate characteristics come into play, such as label distributions, difficulty profiles, and domain-specific properties. Recognizing these distinctions is essential for selecting the appropriate level of analysis for a given application.

7.2.2 Implicit Learning of Semantic Structure

The QRP approach (Section 4.1) demonstrates that models can learn semantic structure without explicit supervision. By removing the query during pre-training, we force the model to reconstruct what query would have produced a given ranking. This implicit learning proves surprisingly effective, outperforming both baseline models and general pre-training objectives (MLM, RTS, STS) applied to the same data (Section 4.3).

Why does implicit learning work? We hypothesize several complementary explanations. First, without access to the query, the model cannot rely on shallow pattern matching between query and candidates. Instead, it must learn abstract semantic properties shared by highly-ranked candidates, containing representations of concepts like "*calorie questions*" or "*capital city questions*" that transfer to new queries. Second, the query-free formulation emphasizes relationships between candidates rather than relationships to a query, which is particularly aligned with reranking where the model must compare candidates to determine relative quality. Third, making the pre-training task harder prevents overfitting to surface-level patterns; the model must develop robust internal representations to solve the more difficult query-free task, and these representations generalize better.

This implicit learning of cluster structure during pre-training has a natural complement: explicit provision of cluster structure at inference time, which we explore in the next section through q-RAG. Both approaches leverage the insight that equivalent questions form coherent

groups with exploitable properties. This suggests a broader principle: semantic structure in language can be exploited either through learning (when training resources are available) or through explicit provision (when inference-time augmentation is possible).

7.2.3 Understanding vs. Knowledge

A consistent finding across experiments is that failures in question answering often reflect understanding gaps rather than knowledge gaps. When an LLM correctly answers "*What is the capital of France?*" but fails on "*Which city serves as France's capital?*", the required knowledge is identical. The model possesses the information but cannot reliably access it across surface variations. The presence of incoherent clusters across all evaluated models (Section 5.2.2) suggests that understanding gaps are a substantial contributor to QA failures, perhaps more than previously recognized.

This distinction has important practical implications. Traditional approaches to improving QA focus on expanding knowledge: larger training corpora, retrieval augmentation with external documents, continued pre-training on domain-specific text. These approaches address knowledge gaps effectively but leave understanding gaps untouched. A model that cannot consistently map equivalent surface forms to the same underlying concept will not be helped by more facts; it needs better question understanding.

The success of q-RAG provides direct evidence for this analysis (Section 5.2.5). Retrieved support questions contain no new factual information as they are simply alternative phrasings of the same information need. Yet q-RAG improves accuracy by up to 9 percentage points, outperforming traditional document-based RAG which does provide new facts. This counterintuitive result confirms that the bottleneck for many questions is not missing knowledge but unreliable access to existing knowledge through inconsistent question understanding. The redundant semantic signal from multiple phrasings helps the model triangulate user intent, increasing the probability of activating the correct parametric knowledge.

The declassification experiments reinforce this theme from a different angle (Section 6.4). When training on declassified data, the model receives semantically equivalent questions but with different surface forms and different answers, which are reconstructed from public corpora rather than the original source. If question understanding were fragile, this perturbation would degrade performance. Instead, models trained on fully declassified WikiQA match baseline performance exactly ($\Delta \approx 0$), suggesting that what matters for learning is the semantic content of the training signal, not the specific surface realization. Models learn to understand questions, not to memorize specific phrasings.

7.3 Limitations

The contributions have several limitations that inform future research directions and should be considered when applying the methods developed in this thesis. We present these limitations following the order of contributions.

Question Retrieval Infrastructure (Chapter 3). The QUADRo database and QRC dataset are English-only and derive from open-domain sources. Extending to other languages requires multilingual question databases, which are substantially less mature than their English counterparts. Specialized domains such as medical, legal, and scientific would require

domain-specific corpora and potentially different modeling approaches, as user question patterns may differ significantly from open-domain factual queries.

Pre-Training (Chapter 4). The QRP method inherits the language and domain limitations of the underlying retrieval infrastructure. Additionally, the self-supervised objective assumes that retrieved candidates form coherent clusters around shared information needs; this assumption may hold less strongly in domains with highly ambiguous or context-dependent questions.

LLM Coherence and q-RAG (Chapter 5). The coherence analysis and q-RAG method target factual questions with well-defined answers. Subjective questions, where different phrasings may legitimately warrant different responses, require careful redefinition of equivalence. Multi-part questions pose additional challenges: a question asking for both a name and a date cannot be satisfied by an equivalent that asks only for the name. Conversational questions have been excluded, as they depend on dialogue context. Furthermore, while we can measure and improve coherence at inference time through q-RAG, we lack methods for directly optimizing LLM coherence during training beyond our preliminary experiments.

Coherence Ranking Loss (Chapter 5). The CR Loss successfully addresses retrieval coherence (Section 5.6.4), but no analogous training objective exists for LLMs. Current LLM training paradigms optimize primarily for accuracy, leaving coherence as an uncontrolled byproduct. A model can be consistently wrong (high coherence, low accuracy) or inconsistently right (low coherence, high accuracy), and current methods do not fully address this independence.

Declassification (Chapter 6). The framework's effectiveness depends on distributional coverage: proprietary questions outside public corpus coverage cannot be effectively mapped. The SimpleQA experiments (Section 6.5) demonstrate this limitation, where questions about niche entities shift toward more common variants. Domain matching between source and replacement corpora is critical; mismatched domains degrade performance regardless of question mapping quality.

7.4 Future Work

We have identified several possible directions that naturally extend from this work. All future works rely on the foundations defined in this thesis while addressing limitations identified above.

- **Scaling Coherence-Aware LLM Training:** Chapter 5 presents preliminary experiments showing that q-RAG's coherence benefits can be distilled into model parameters through DPO and SFT, achieving +3.93 EM and +24.0 coherence points (Section 5.4). Scaling these experiments to larger models and more diverse training data could produce models with robust coherence across query variations. The full QRC dataset, combined with systematic cluster-based preference generation, offers a path toward making coherence an explicit training objective alongside accuracy.
- **Cross-Lingual Question Retrieval:** Extending QUADRo to multilingual settings would enable cross-lingual QA, querying in one language while retrieving from another. The question equivalence relation is language-independent as a question about the capital of France is equivalent whether asked in English, Italian, or Chinese. The challenge

is building an extension of the QRC dataset in multi- and cross-lingual settings. In addition, when dealing with questions posed in different languages, it is necessary to pay attention to language-specific and culturally grounded interpretations of queries, which may affect how equivalence is perceived across languages.

- **Declassification Beyond Question Answering:** The declassification framework demonstrates that semantic mapping can replace sensitive content while preserving utility. Similar techniques may apply to other NLP tasks where equivalence can be defined: text classification (mapping sensitive documents to semantically equivalent public alternatives), summarization (declassifying source documents while preserving summary-relevant content), or dialogue systems (replacing personal details with equivalent alternatives).

- **Benchmark Declassification at Scale:** Chapter 6 demonstrates that declassified test sets preserve evaluation validity, enabling "shadow benchmarks" where organizations release mapped versions publicly while retaining originals for authoritative assessment (Section 6.4). Scaling this approach to community-wide adoption would require standardization of declassification protocols and coordination across research organizations to maintain benchmark integrity as LLM training scales.

- **Formal Privacy Analysis:** The declassification framework provides practical content protection, as the output contains only pre-existing public content, but does not offer formal guarantees in the differential privacy sense. Extending the framework with formal privacy analysis would strengthen applicability to regulated domains such as healthcare and finance where such guarantees may be required.

- **Declassification of Supporting Context:** Current declassification operates on question-answer pairs. Many QA tasks include passages, conversation history, or other contextual information that also requires declassification. Extending the framework to preserve contextual equivalence while replacing sensitive content would enable release of richer grounded QA datasets.

- **Industrial Validation:** Our evaluation uses public benchmarks as proxies for proprietary data. Large-scale validation on actual proprietary industrial datasets, measuring declassification quality, coverage, and utility under real governance constraints, remains an important next step for establishing the practical applicability of the framework.

- **Adaptive Declassification:** Current declassification uses fixed retrieval and reconstruction pipelines. An adaptive and dynamic system could change and adapt the retrieval strategies based on query characteristics making it possible to use different corpora for different domains. Other possibilities include adjusting retrieval depth based on corpus coverage, or different fallback approaches.

7.5 Final Thoughts

This thesis began with a practical problem: finding equivalent questions in large databases to enable efficient question answering. The QUADRo system solved this issue, but the work also raised deeper questions on how this question equivalence can be used across the QA pipeline.

The progression from retrieval to representation to relationships to replacement reflects a deepening understanding of what questions are. Questions are not just strings of text, but objects defined by "what they seek", that are connected to other questions through shared information. The technical contributions of each chapter are significant on their own, but together they constitute a perspective on question understanding that we hope proves useful to the field.

Working across these chapters has also highlighted more subtle dimensions of equivalence that are relevant for different applications. One such dimension is *functional equivalence*, which concerns whether the same answer can appropriately address both questions, a notion captured in QUADRo through answer-aware retrieval (Section 3.3). *Training equivalence* asks whether a model trained on one question develops the same capabilities as if trained on the other, which is what declassification requires (Section 6.2). *Difficulty equivalence* asks whether both questions pose similar challenges, which is what benchmark preservation demands (Section 6.5). These distinctions emerged from experimental observations rather than theoretical analysis, highlighting the value of empirical investigation in revealing conceptual structure. Future research may identify additional dimensions of equivalence relevant to other applications.

Looking back, the thesis can be summarized in a single insight: *semantically equivalent questions can be identified and grouped, and these groups prove useful at multiple levels of granularity.* At the pair level, equivalence enables answer transfer. At the cluster level, equivalence enables coherence optimization. At the dataset level, equivalence enables content transformation. Each scale reveals new properties and enables new applications, but all derive from the same underlying semantic organization. This insight guided the research throughout and continues to suggest new directions.

The progression from QUADRo to dataset declassification shows that a careful analysis of question understanding can lead to both useful insights and practical solutions. Scientific insights include the discovery that coherence and accuracy are partially independent properties that require separate optimization, that understanding gaps matter as much as knowledge gaps in limiting QA performance, and that semantic structure can be learned implicitly through self-supervised objectives. We hope these insights and the perspective developed here prove useful to others pursuing question understanding, and inspire new investigations into semantic equivalence and its applications across NLP.

Bibliography

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024a. Phi-3 technical report: A highly capable language model locally on your phone.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024b. Phi-4 technical report.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun

Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. 2025. gpt-oss-120b & gpt-oss-20b model card.

Anthropic. 2024. Introducing the next generation of claude.

Fan Bai, Keith Harrigian, Joel Stremmel, Hamid Hassanzadeh, Ardavan Saeedi, and Mark Dredze. 2024. Give me some hard questions: Synthetic data generation for clinical qa.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

Christian Bentz and Aleksandrs Berdicevskis. 2016. Learning pressures reduce morphological complexity: Linking corpus, computational and experimental evidence. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 222–232, Osaka, Japan. The COLING 2016 Organizing Committee.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

J Jansen Bernard, Amanda Spink, Chris Blakely, and Sherry Koshman. 2007. Defining a

session on web search engines. *Journal of the American Society for Information Science and Technology*, 58(6):862–871.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *EMNLP*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Robin D Burke, Kristian J Hammond, Vladimir Kulyukin, Steven L Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the faq finder system. In *AI Magazine*, volume 18, page 57.

Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. 2011. Learning the latent topics for question retrieval in community QA. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 273–281, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Stefano Campese, Ivano Lauriola, and Alessandro Moschitti. 2023. Quadro: Dataset and models for question-answer database retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15573–15587.

Daniel Campos, ChengXiang Zhai, and Alessandro Magnani. 2023. Noise-robust dense retrieval via contrastive alignment post training. *arXiv preprint arXiv:2304.03401*.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1).

Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14550–14565, Miami, Florida, USA. Association for Computational Linguistics.

Aditi Chaudhary, Karthik Raman, and Michael Bendersky. 2024. It's all relative! – a synthetic query generation approach for improving zero-shot relevance prediction. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1645–1664, Mexico City, Mexico. Association for Computational Linguistics.

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

Haitian Chen, Qingyao Ai, Xiao Wang, Yiqun Liu, Fen Lin, and Qin Liu. 2024a. Unsupervised dense retrieval with counterfactual contrastive learning. *arXiv preprint arXiv:2412.20756*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Ziyang Chen, Erxue Min, Xiang Zhao, Yunxin Li, Xin Jia, Jinzhi Liao, Jichao Li, Shuaiqiang Wang, Baotian Hu, and Dawei Yin. 2025. A question answering dataset for temporal-sensitive retrieval-augmented generation. *Scientific Data*, 12:1855.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. 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. Association for Computational Linguistics.

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

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.

Christophe Coupé, Yoon Mi Oh, Dan Dediu, and François Pellegrino. 2019. Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Science Advances*, 5(9):eaaw2594.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 1034–1046, USA. Association for Computational Linguistics.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Luca Di Liello, Nicola Ferretti, and Alessandro Moschitti. 2022a. Pre-training transformer models with sentence-level objectives for answer sentence selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11104–11115.

Luca Di Liello, Matteo Gabburo, and Alessandro Moschitti. 2022b. Effective pretraining objectives for transformer-based autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5533–5547, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022c. Paragraph-based transformer pre-training for multi-sentence inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2521–2531, Seattle, United States. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library.

Cynthia Dwork. 2006. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–12. Springer.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, page 265–284, Berlin, Heidelberg. Springer-Verlag.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria. Association for Computational Linguistics.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1156–1165, New York, NY, USA. Association for Computing Machinery.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

Amer Farea and Frank Emmert-Streib. 2025. Understanding question-answering systems: Evolution, applications, trends, and challenges. *Eng. Appl. Artif. Intell.*, 156(PA).

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building watson: An overview of the deepqa project. *AI Mag.*, 31(3):59–79.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. Splade v2: Sparse lexical and expansion model for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Matteo Gabburo, Stefano Campese, Federico Agostini, and Alessandro Moschitti. 2024a. Datasets for multilingual answer sentence selection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8947–8958, Miami, Florida, USA. Association for Computational Linguistics.

Matteo Gabburo, Siddhant Garg, Rik Koncel-Kedziorski, and Alessandro Moschitti. 2023. Learning answer generation using supervision from automatic question answering evaluators. *arXiv preprint arXiv:2305.15344*.

Matteo Gabburo, Nicolaas Paul Jedema, Siddhant Garg, Leonardo F. R. Ribeiro, and Alessandro Moschitti. 2024b. Measuring retrieval complexity in question answering systems. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14636–14650, Bangkok, Thailand. Association for Computational Linguistics.

Matteo Gabburo, Rik Koncel-Kedziorski, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022. Knowledge transfer from answer ranking to answer generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9481–9495, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2019. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7780–7788.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. Declutr: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 879–895.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: an automatic question-answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61 (Western), page 219–224, New York, NY, USA. Association for Computing Machinery.

Mengtian Guo, Mutasem Al-Darabsah, Choon Hui Teo, Jonathan May, Tarun Agarwal, and Rahul Bhagat. 2025. Learning to rewrite negation queries in product search. In *Proceedings of COLING Industry Track*, pages 575–582.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C Lipton. 2019. Amazonqa: A review-based question answering task.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938.

Lukas Haas, Gal Yona, Giovanni D'Antonio, Sasha Goldshtein, and Dipanjan Das. 2025. Simpleqa verified: A reliable factuality benchmark to measure parametric knowledge.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016a. Learning to rewrite queries. In *Proceedings of CIKM*, pages 1443–1452.

Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016b. Learning to rewrite queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 1443–1452, New York, NY, USA. Association for Computing Machinery.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 1011–1019.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Alejandro Hernández-Cano, Alexander Hägele, Allen Hao Huang, Angelika Romanou, Antoni-Joan Solergibert, Barna Pasztor, Bettina Messmer, Dhia Garbaya, Eduard Frank Ďurech, Ido Hakimi, Juan García Giraldo, Mete Ismayilzada, Negar Foroutan, Skander Moalla, Tiancheng Chen, Vinko Sabolčec, Yixuan Xu, Michael Aerni, Badr AlKhamissi, Ines Altemir Marinas, Mohammad Hossein Amani, Matin Ansaripour, Ilia Badanin, Harold Benoit, Emanuela Boros, Nicholas Browning, Fabian Bösch, Maximilian Böther, Niklas Canova, Camille Challier, Clement Charmillot, Jonathan Coles, Jan Deriu, Arnout Devos, Lukas Drescher, Daniil Dzenhaliou, Maud Ehrmann, Dongyang Fan, Simin Fan, Silin Gao, Miguel Gila, María Grandury, Diba Hashemi, Alexander Hoyle, Jiaming Jiang, Mark Klein, Andrei Kucharavy, Anastasiia Kucherenko, Frederike Lübeck, Roman Machacek, Theofilos Manitaras, Andreas Marfurt, Kyle Matoba, Simon Matrenok, Henrique Mendoncça, Fawzi Roberto Mohamed, Syrielle Montariol, Luca Mouchel, Sven Najem-Meyer, Jingwei Ni, Gennaro Oliva, Matteo Pagliardini, Elia Palme, Andrei Panferov, Léo Paoletti, Marco Passerini, Ivan Pavlov, Auguste Poiroux, Kaustubh Ponkshe, Nathan Ranchin, Javi Rando, Mathieu Sauser, Jakhongir Saydaliev, Muhammad Ali Sayfiddinov, Marian Schneider, Stefano Schuppli, Marco Scialanga, Andrei Semenov, Kumar Shridhar, Raghav Singhal, Anna Sotnikova, Alexander Sternfeld, Ayush Kumar Tarun, Paul Teiletche, Jannis Vamvas, Xiaozhe Yao, Hao Zhao Alexander Ilic, Ana Klimovic, Andreas Krause, Caglar Gulcehre, David Rosenthal, Elliott Ash, Florian Tramèr, Joost VandeVondele, Livio Veraldi, Martin

Rajman, Thomas Schulthess, Torsten Hoefler, Antoine Bosselut, Martin Jaggi, and Imanol Schlag. 2025. Apertus: Democratizing Open and Compliant LLMs for Global Language Environments. https://arxiv.org/abs/2509.14233.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433, Berlin, Germany. Association for Computational Linguistics.

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

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

Doris Hoogeveen, Karin M Verspoor, and Timothy Baldwin. 2015. Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian document computing symposium*, pages 1–8.

Chao-Chun Hsu, Eric Lind, Luca Soldaini, and Alessandro Moschitti. 2021. Answer generation for retrieval-based question answering systems. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4276–4282, Online. Association for Computational Linguistics.

J. Hu et al. 2024a. Prompt sensitivity in large language models. *arXiv preprint arXiv:2402.xxxxx*.

Lijie Hu, Ivan Habernal, Lei Shen, and Di Wang. 2024b. Differentially private natural language models: Recent advances and future directions. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 478–499. Association for Computational Linguistics.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2020. Codesearchnet challenge: Evaluating the state of semantic code search.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. QuoraQP dataset. Online dataset.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Rber, Edouard Grave, Armand Joulin, and Matthijs Douze. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Bernard J. Jansen, Amanda Spink, Chris Blakely, and Sherry Koshman. 2007. Defining a session on web search engines: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 58(6):862–871.

Bernard J. Jansen, Amanda Spink, and Jan Pedersen. 2005. A temporal comparison of altavista web searching: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 56(6):559–570.

Eunjin Jeong, Jangryul Kim, and Soonhoi Ha. 2022. Tensorrt-based framework and optimization methodology for deep learning inference on jetson boards. *ACM Trans. Embed. Comput. Syst.*, 21(5).

Zongcheng Ji, Fei Xu, Bin Wang, and Ben He. 2012. Question-answer topic model for question retrieval in community question answering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2471–2474. ACM.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023b. Mistral 7b. ArXiv:2310.06825 [cs.CL].

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019a. Billion-scale similarity search with gpus. volume 7, pages 535–547.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019b. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. Gooaq: Open question answering with diverse answer types. *arXiv preprint*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Ivica Kostric and Krisztian Balog. 2024. A surprisingly simple yet effective multi-query rewriting method for conversational passage retrieval.

Konstantin Kotschenreuther. 2024. Ehr-ds-qa: A synthetic qa dataset derived from medical discharge summaries for enhanced medical information retrieval systems. RRID:SCR_007345.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Ivano Lauriola and Alessandro Moschitti. 2021a. Answer sentence selection using local and global context in transformer models. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 –April 1, 2021, Proceedings, Part I*, page 298–312, Berlin, Heidelberg. Springer-Verlag.

Ivano Lauriola and Alessandro Moschitti. 2021b. Answer sentence selection using local and global context in transformer models. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 –April 1, 2021, Proceedings, Part I*, page 298–312, Berlin, Heidelberg. Springer-Verlag.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1279–1289, San Diego, California. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Yashar Mehdad Kučer, Raymond Mooney, Wen-tau Yih, Sebastian Riedel, and Pontus Stenetorp. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft coco: Common objects in context.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Wei Chen, and Xueqi Cheng. 2023a. On the robustness of generative retrieval models: An out-of-distribution perspective. *arXiv preprint arXiv:2306.12756*.

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023b. Black-box adversarial attacks against dense retrieval models: A multi-view contrastive learning method. In *Proceedings of CIKM*, pages 1647–1656.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S. Weld. 2020. S2orc: The semantic scholar open research corpus.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint*.

Yosi Mass, Boaz Carmeli, Haggai Roitman, and David Konopnicki. 2020. Unsupervised faq retrieval with question generation and bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 807–812.

Rui Meng, Ye Liu, Semih Yavuz, Divyansh Agarwal, Lifu Tu, Ning Yu, Jianguo Zhang, Meghana Bhat, and Yingbo Zhou. 2024. Augtriever: Unsupervised dense retrieval and domain adaptation by scalable data augmentation.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*.

Moran Mizrahi, Guy Kaplan, Dan Malber, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024a. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Moran Mizrahi et al. 2024b. State of what art? a call for multi-prompt llm evaluation. *TACL*, 12:933–949.

Dan Moldovan, Christine Clark, Sanda Harabagiu, and Steve Maiorano. 2003. COGEX: A logic prover for question answering. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 166–172.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. Covid-qa: a question answering dataset for covid-19.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 335–es, USA. Association for Computational Linguistics.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *European Conference on Machine Learning*, pages 318–329. Springer.

Alessandro Moschitti and Fabio Massimo Zanzotto. 2007. Fast and effective kernels for relational learning from texts. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 649–656, New York, NY, USA. Association for Computing Machinery.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2786–2792. AAAI Press.

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2016a. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545, San Diego, California. Association for Computational Linguistics.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016b. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545, San Diego, California. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Tri Nguyen and et al. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@NIPS*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Jianmo Ni, Chenguang Zhu, Weizhu Chen, and Julian McAuley. 2019. Learning to attend on essential terms: An enhanced retriever-reader model for open-domain question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 335–344, Minneapolis, Minnesota. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert.

Nouha Othman, Rim Faiz, and Kamel Smaïli. 2019. Enhancing question retrieval in community question answering using word embeddings. *Procedia Computer Science*, 159:485–494.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Anwesan Pal, Radhika Bhargava, Kyle Hinsz, Jacques Esterhuizen, and Sudipta Bhattacharya. 2024a. The empirical impact of data sanitization on language models. *CoRR*, abs/2411.05978.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024b. Smaug: Fixing failure modes of preference optimisation with dpo-positive.

Md Rizwan Parvez, Jianfeng Chi, Wasi Uddin Ahmad, Yuan Tian, and Kai-Wei Chang. 2023. Retrieval enhanced data augmentation for question answering on privacy policies. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 201–210, Dubrovnik, Croatia. Association for Computational Linguistics.

Nils Peinelt, Trung Nguyen, and Maria Liakata. 2020. tbert: Topic models and bert joining forces for semantic similarity detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7047–7055, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The text anonymization benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101.

Ella Rabinovich, Samuel Ackerman, Orna Raz, Eitan Farchi, and Ateret Anaby Tavor. 2023. Predicting question-answering performance of large language models through semantic consistency. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 138–154, Singapore. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

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

Harsh Raj, Domenic Rosati, and Subhabrata Majumdar. 2023. Measuring reliability of large language models through semantic consistency.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019a. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019b. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Nils Reimers and Iryna Gurevych. 2019a. 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.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992.

Adam Roberts, Colin Raffel, Noam Shazeer, et al. 2020. How much knowledge can you pack into the parameters of a language model?

Stephen Robertson and Hugo Zaragoza. 2009a. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Stephen Robertson and Hugo Zaragoza. 2009b. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. Faq retrieval using query-question similarity and bert-based query-answer relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1113–1116.

David Sánchez and Montserrat Batet. 2016. C-sanitized: A privacy model for document redaction and sanitization. *J. Assoc. Inf. Sci. Technol.*, 67(1):148–163.

Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. 2024. Towards faithful and robust LLM specialists for evidence-based question-answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 1913–1931, Bangkok, Thailand. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France. Originally arXiv:1611.01603v6.

Yeon Seonwoo, Juhee Son, Jiho Jin, Sang-Woo Lee, Ji-Hoon Kim, Jung-Woo Ha, and Alice Oh. 2022. Two-step question retrieval for open-domain QA. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1487–1492, Dublin, Ireland. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 458–467, Seattle, Washington, USA. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 373–382, New York, NY, USA. Association for Computing Machinery.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *ArXiv*, abs/1701.06538.

Yikang Shen, Wenge Rong, Nan Jiang, Baolin Peng, Jie Tang, and Zhang Xiong. 2017. Word embedding based correlation model for question/answer matching. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3511–3517. AAAI Press.

Yunxiao Shi, Xing Zi, Zijing Shi, Haimin Zhang, Qiang Wu, and Min Xu. 2024. Enhancing retrieval and managing retrieval: A four-module synergy for improved quality and efficiency in rag systems.

Eriks Sneiders. 2002. Automated question answering using question templates that cover the conceptual model of the database. In *International Conference on Applications of Natural Language to Information Systems*, pages 235–239.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Praveena Sunkara. 2024. Enhancing question answering systems with rephrasing strategies: A study on bert sensitivity and refinement techniques.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 464–473.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

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

James Thorne and et al. 2018. Fever: a large-scale dataset for fact extraction and verification. In *NAACL*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Ellen M. Voorhees. 2006. The trec 2005 robust track. *SIGIR Forum*, 40(1):41–48.

Ellen M Voorhees and Dawn M Tice. 1999. The trec-8 question answering track report. In *TREC*, volume 99, pages 77–82.

Anton Voronov, Lena Voronov, Thomas Wolf, and Stas Bekman. 2024a. Mind your format: Towards consistent evaluation of in-context learning improvements. *arXiv preprint arXiv:2401.06766*.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024b. Mind your format: Towards consistent evaluation of in-context learning improvements. In *Findings of ACL*.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360, Seattle, United States. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

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

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024c. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2016a. Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451, San Diego, California. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2016b. Machine comprehension using match-lstm and answer pointer. In *Proceedings of the 5th International Conference on Learning Representations*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020a. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788.

Yaxuan Wang, Hanqing Lu, Yunwen Xu, Rahul Goutam, Yiwei Song, and Bing Yin. 2021. Queen: Neural query rewriting in e-commerce.

Zizhen Wang, Yixing Fan, Jiafeng Guo, Liu Yang, Ruqing Zhang, Yanyan Lan, Xueqi Cheng, Hui Jiang, and Xiaozhao Wang. 2020b. Match^2: A matching over matching model for similar question identification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 559–568.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4).

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Karina Nguyen, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022c. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Terry Winograd. 1972. *Understanding Natural Language*. Academic Press, New York.

W. A. Woods. 1973. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, AFIPS '73, page 441–450, New York, NY, USA. Association for Computing Machinery.

Chen Wu, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Are neural ranking models robust? *ACM Transactions on Information Systems*, 41(2):1–36.

Chunqiu Steven Xia, Yinlin Deng, and Lingming Zhang. 2024. Top leaderboard ranking = top coding proficiency, always? evoeval: Evolving coding benchmarks via llm.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023a. C-pack: Packaged resources to advance general chinese embedding.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023b. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Oleksandr Yermilov, Vipul Raheja, and Artem Chernodub. 2023. Privacy- and utility-preserving NLP with anonymized data: A case study of pseudonymization. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 232–241, Toronto, Canada. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP) — Volume 1: Long Papers*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Jie Ren, Tianqi Zheng, Hanqing Lu, Han Xu, Hui Liu, Yue Xing, and Jiliang Tang. 2025. Mitigating the privacy issues in retrieval-augmented generation (RAG) via pure synthetic data. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24538–24569, Suzhou, China. Association for Computational Linguistics.

Yanjun Zhang and Wee Sun Lee. 2010. A comparison of linguistic and statistical methods for identifying semantic similarity in text. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 546–554.

Zeyu Zhang, Thuy Vu, Sunil Gandhi, Ankit Chadha, and Alessandro Moschitti. 2022. Wdrass: A web-scale dataset for document retrieval and answer sentence selection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4707–4711.

Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. 2021. Joint models for answer verification in question answering systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3252–3262, Online. Association for Computational Linguistics.

Ziqi Zhang, Ali Shahin Shamsabadi, Hanxiao Lu, Yifeng Cai, and Hamed Haddadi. 2025. Membership and memorization in LLM knowledge distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20085–20095, Suzhou, China. Association for Computational Linguistics.

Tony Z Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12697–12706.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023a. Large language models are not robust multiple choice selectors. In *Proceedings of the 12th International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Yoshua X ZXhang, Yann M Haxo, and Ying X Mat. 2023. Falcon llm: A new frontier in natural language processing. *AC Investment Research Journal*, 220(44).

Appendix A

A.1 Annotation Anecdotes

In this section we analyzed representative annotation cases to illustrate the task's inherent complexity

Case A (True Positive, Easy):

Query: *"How old are oldest fossils of organisms?"*

Question: *"The oldest fossils date to how long ago?"*

Answer: *"The oldest documented fossil organisms date to roughly 3.8 billion years ago, shortly after the period of heavy asteroid bombardment in Earth's history."*

Explanation : The questions have similar phrasing and ask the same thing, making this an easy positive

Case B (True Positive, Answer-Dependent):

Query: *"Where did the term bully pulpit come from?"*

Question: *"Who or what is meant by the expression Bully Pulpit?"*

Answer: *"This phrase, used by Theodore Roosevelt, refers to the office of presidency. A pulpit is a place to preach from and "bully" is an older adjective meaning excellent. Therefore a bully pulpit is a great place to make speeches from . Roosevelt appreciated the fact that people listened to him when he spoke as the President."*

Explanation: The questions appear to ask different things (origin vs. meaning), but the comprehensive answer addresses both, making this a correct positive that requires considering the answer.

Case C (Annotation Error):

Query: *"What is the name called of a back end of boat?"*

Question: *"What is the name for the back and front of a boat?"*

Answer: *"he front of a vessel be it a ship or boat, is the bow or stem. The back is the stern."*

Explanation: This was incorrectly labeled negative. While the question asks for more than the query, the answer is correct for the query (stern), so these should be considered equivalent from a QA perspective.

Case D (Annotation Error):

Query: *"How to replace battery in liftmaster remote control?"*

Question: *"How to change battery in liftmaster remote keypad?"*

Answer: *'Slide the battery cover down. The battery is located at the bottom of your keypad.', 'Disconnect the old battery and remove it. Remember to dispose of your old battery correctly.', 'Install the new battery. ... ', 'Put the battery cover back in place.'*

Explanation: This was incorrectly labeled positive. Despite similar phrasing, "remote control" and "remote keypad" are distinct devices with different procedures.

Appendix B

B.1 Prompt Templates for LLM Coherence Experiments

This appendix provides the complete prompt templates used in the LLM coherence experiments described in Section 5.2. All prompts use the Llama/Mistral instruction format with `[INST]` tags; adaptations for other model formats follow the same content structure.

B.1.1 Base QA Prompt (No Retrieval)

Listing 1: Prompt template for generating equivalent query clusters using Phi-3.

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.
[INST]
Instruction: You are a powerful Question Answering System. You should
answer the question without external context.
If you don't know the answer, don't try to answer; just say "I don't
know" and avoid adding further context.
QUESTION: {question}
[/INST]
```

B.1.2 Q-RAG Prompt: Questions Only

Listing 2: Prompt template for generating equivalent query clusters using Phi-3.

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.
[INST]
Instruction:
You are an AI Assistant acting as a Frequently Asked Questions (FAQ)
system able to answer questions.
You should answer the question based on the provided context.
The context consists of N questions similar and related to the input
question (frequently asked) and helps you to reason and formulate the
correct answer to the input question.
You must respect the following rules:
[RULES]
+ If you don't know the answer, don't try to answer; just say "I don't
  know" and avoid adding unnecessary information.
+ Do not explicitly state that you are a FAQ system.
+ Do not explicitly cite the documents you use to answer the question.
+ Do not mention what other people ask.
+ Use the context to infer missing information or to clarify ambiguous
  questions.
+ Provide concise answers.
```

```
Here is context to help:
{context}
QUESTION: {question}
[/INST]
```

B.1.3 Q-RAG Prompt: Question-Answer Pairs

Listing 3: Prompt template for generating equivalent query clusters using Phi-3.

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.
[INST]
Instruction:
You are an AI Assistant acting as a Frequently Asked Questions (FAQ)
system able to answer questions.
You should answer the question based on the provided context.
The context consists of N question-answer pairs, where the questions
are similar and related to the input question. The answer from each
pair is the correct answer for that particular question.
The question-answer pairs help you to reason and formulate the correct
answer to the input question.
You must respect the following rules:
[RULES]
+ If you don't know the answer, don't try to answer; just say "I don't
  know" and avoid adding unnecessary information.
+ Do not explicitly state that you are a FAQ system.
+ Do not explicitly cite the documents you use to answer the question.
+ Do not mention what other people ask.
+ Use the context to infer missing information or to clarify ambiguous
  questions.
+ Provide concise answers.
Here is context to help:
{qa_pairs}
QUESTION: {question}
[/INST]
```

B.1.4 PopQA-TP Prompt: Base (No Retrieval)

Listing 4: Prompt template for generating equivalent query clusters using Phi-3.

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.
[INST]
Instruction: You are a powerful Question Answering System. You should
answer the question without external context.
The answer must always be an entity or a list of entities separated
by a comma.
The input question is about the "{property}" topic.
If you don't know the answer, don't try to answer; just say "I don't
know" and avoid adding further context.
QUESTION: {question}
[/INST]
```

B.1.5 PopQA-TP Prompt: Q-RAG

Listing 5: Prompt template for generating equivalent query clusters using Phi-3.

```
Below is an instruction that describes a task. Write a response that
```

```
appropriately completes the request.
[INST]
Instruction:
You are an AI Assistant acting as a Frequently Asked Questions (FAQ)
system able to answer questions.
You should answer the question based on the provided context.
The context consists of N questions similar and related to the input
question (frequently asked) and helps you to reason and formulate the
correct answer to the input question.
The input question is about the "{property}" topic.
You must respect the following rules:
[RULES]
+ If you don't know the answer, don't try to answer; just say "I don't
  know" and avoid adding unnecessary information.
+ Do not explicitly state that you are a FAQ system.
+ Do not explicitly cite the documents you use to answer the question.
+ Do not mention what other people ask.
+ Use the context to infer missing information or to clarify ambiguous
  questions.
+ The answer must always be an entity or a list of entities separated
  by comma.
Here is context to help:
{context}
QUESTION: {question}
[/INST]
```

B.1.6 Question Generation Prompt

Used for the "Q Generation" configuration in Section 5.3.2, where the LLM first generates paraphrases of the input question.

Listing 6: Prompt template for generating equivalent query clusters using Phi-3.

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.
[INST]
Instruction:
You are a powerful AI that given an input question generates 5 similar
questions.
A similar question is a question that is asking for the same thing as
the input but posed in a different manner or using different words or
in a way that is not trivial for a language model.
You should generate 5 similar questions.
Rules:
+ Your output must be a valid JSON, just a JSON, no other text or
  information is allowed.
+ The structure of the JSON must follow this:
{
   "q1": "generated question 1",
   "q2": "generated question 2",
   "q3": "generated question 3",
   "q4": "generated question 4",
   "q5": "generated question 5"
}
+ The questions must be different from each other and from the input
  but express the same meaning and ask for the same thing.
+ The questions must require the same answer and the same documents
  to be answered.
+ Be sure that the output is valid JSON, escape where necessary.
+ If the definition or the meaning of a word/thing is asked in the
  input question, be sure the generated questions ask for the same
  word/thing meaning.

Here is a couple of examples to help:
```

```
Example 1:
input question: Can lizards fly?
generated questions:
{
    "q1": "Can lizards fly through the air?",
    "q2": "Do lizards fly?",
    "q3": "Are there lizards which can fly?",
    "q4": "Are there any flying reptiles?",
    "q5": "Are there any flying lizards?"
}

Example 2:
input question: What is the capital of France?
generated questions:
{
    "q1": "Can you tell me what the capital of France is?",
    "q2": "Which city serves as France's capital?",
    "q3": "Name the capital city of France",
    "q4": "France's capital is what city?",
    "q5": "What city is the governmental seat of France?"
}

input question: {question}
[/INST]
generated questions:
```

B.1.7 Chain-of-Thoughts Prompt

Used for the "Chain-of-Thoughts" configuration in Section 5.3.2, where the model reasons about alternative phrasings before answering.

Listing 7: Prompt template for generating equivalent query clusters using Phi-3.

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.
[INST]
Instruction: Answer the question following the reasoning process in
Example 1 and Example 2.
If you don't know the answer, don't try to answer; just say "I don't
know."
The output must be only the answer.

Example 1:
input question: At what temperature is a chicken done?
similar questions are:
+ What temperature does a chicken have to be done?
+ What is the temperature supposed to be in the chicken to be done?
+ What temperature should a whole chicken be cooked at?
+ What is the "internal temperature" of done chicken?
+ What temperature do you cook the chicken to?
if these are similar questions, then the answer is: All poultry should
reach a safe minimum internal temperature of 165 F (73.9 C) as measured
with a food thermometer.

Example 2:
input question: elegxo meaning
similar questions are:
+ What does the term elegxo signify?
+ Can you explain the meaning of elegxo?
+ What is the definition of elegxo?
+ What does elegxo mean in Greek?
+ How is elegxo used in a sentence and what does it mean?
if these are similar questions, then the answer is: The Ancient Greek
term "elegxo" means to refute, expose, convict, or examine.
```

```
Let's begin:
input question: {question}
[/INST]
```

B.1.8 Paragraph-Based RAG Prompt

Used for the classical RAG baseline in Section 5.2.6, where Wikipedia paragraphs are retrieved via DPR.

Listing 8: Prompt template for generating equivalent query clusters using Phi-3.

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.
[INST]
Instruction: You are a powerful Question Answering System. You should
answer the question based on the provided context.
The context consists of N documents that are relevant to the input
question.
If you don't know the answer, don't try to answer; just say "I don't
know" and avoid adding further context.
Here is context to help:
{paragraphs}
QUESTION: {question}
[/INST]
```

Appendix C

C.1 Prompt Templates for LLM question generation

To generate equivalent queries, we uses Phi-3-mini (Abdin et al., 2024a), 3.8B parameters. As mentioned in 5, for each query in the MSMARCO and NQ datasets, we generated 10 semantically equivalent rephrasing, maintaining the original intent while introducing linguistic and lexical diversity. The generation process utilized the following is visible in Listing 9

Listing 9: Prompt template for generating equivalent query clusters using Phi-3.

```
You are a powerful question rephraser and question generation system. Given a question coming from the
    MSMARCO dataset, your task is to generate 10 EQUIVALENT questions using different styles coming
    from other different datasets. Two questions are defined EQUIVALENT, is they (i) are asking for
    exact same thing even if they contain a very different wording, and (ii) they require the same
    answer.

You can use the following dataset styles:

+ SQuAD-style: Starts with an introductory phrase and focuses on a specific piece of information.
+ MS-MARCO-style: Framed as a request for information, with a more open-ended tone.
+ DuoRC-style: Asks about typical or common symptoms/indicators, using ``if'' to set up the context.
+ HotpotQA-style: Combines a request for key indicators with a follow-up on how to identify them.
+ NQ-style: Concise and direct, focused on a specific piece of information. Typically starts with
    ``what'', ``who'', ``where'', etc.
+ TriviaQA-style: More open-ended, sometimes requiring nuanced answers. May include additional context.
+ WebQA-style: Framed as a request for a list or set of information. Often starts with ``Can you
    provide...'', ``List the...'', etc.


Your task is to produce a well formatted and
parsable JSON containing the EQUIVALENT questions.
The produced output must be EXACTLY AS FOLLOWS:
   ```{
      "original_question": $INPUT_QUESTION,
      "equivalent_questions":[
         {'question': $EQUIVALENT_QUESTION_1, 'style': $EQUIVALENT_QUESTION_STYLE_1},
         {'question': $EQUIVALENT_QUESTION_2, 'style': $EQUIVALENT_QUESTION_STYLE_2},
         ...,
         {'question': $EQUIVALENT_QUESTION_10, 'style': $EQUIVALENT_QUESTION_STYLE_10}
      ],
   }```


Where the $EQUIVALENT_QUESTION_NTH and $EQUIVALENT_QUESTION_STYLE_NTH are the generated question and
    the used style.
To produce the JSON you MUST respect the following rules:
   + The generated questions should be short and concise when possible.
   + Remember: two questions are equivalent if (i) they are asking for exact same thing, and (ii) they
       require the same answer.
```
```

```
   + Remember: each generated question must follow a different style.
   + Remember: the output must be a valid JSON ready to be used without further post-processing.
Here you can find an example:

INPUT_QUESTION: symptoms of a dying mouse
OUTPUT JSON:
{
   "original_question": "symptoms of a dying mouse",
   "equivalent_questions":[
      {"question":"What are the typical symptoms that indicate a mouse is dying?", "style":"NQ"},
      {"question":"Identify the most common signs that a mouse is approaching the end of its life.",
           "style":"TriviaQA"},
      {"question":"Can you provide a list of the primary indicators that a mouse is in the process of
           dying?", "style":"WebQA"},
      {"question":"According to medical experts, what are the primary symptoms that indicate a mouse
           is nearing the end of its life?", "style":"SQuAD"},
      {"question":"I need to know the most common signs that a mouse is dying. Can you provide me with
           that information?", "style":"MS-MARCO"},
      {"question":"What are the key indicators that a mouse is in the process of dying, and how can
           these be identified?", "style":"HotpotQA"},
      {"question":"If a mouse is showing signs of dying, what are the typical symptoms that would
      be observed?", "style":"DuoRC"},
      {"question":"How does the appearance of a mouse's coat change when it's approaching death?",
      "style":"NQ"},
      {"question":"What changes in eating and drinking habits suggest a mouse is near death?",
      "style":"NQ"},
      {"question":"As a mouse approaches death, it may show this sign related to body temperature.
           What is it?", "style":"TriviaQA"}
   ]
}

Remember, just return the JSON, no additional text.
Here is the input: {question}
Please provide your JSON output
```

Appendix D

D.1 Answer Correctness Annotation

For Answer candidate annotations we use the prompt in Listing 10. This prompt is used during the LLM annotation process in the answer declassification procedure.

Listing 10: Answer Correctness Annotation

```
You are evaluating whether a candidate sentence correctly answers a question.

Question: {question}

Reference answer (known correct): {reference}

Candidate: {candidate}

Does the candidate correctly answer the question? Respond with only "correct" or "incorrect".
```

D.2 Evaluation Annotation Prompt

In this appendix we report the original SimpleQA grading prompt from Wei et al. (2024), which classifies responses as "correct", "incorrect", or "not attempted". Both the system and user prompt are visible in Listing 11 and Listing 12.

Listing 11: SimpleQA evaluation system prompt

```
Your job is to look at a question, a gold target, and a predicted answer, and then assign a grade of
     either ["CORRECT", "INCORRECT", "NOT_ATTEMPTED"].

First, I will give examples of each grade, and then you will grade a new example.

The following are examples of CORRECT predicted answers.

   Question: What are the names of Barack Obama's children?
   Gold target: Malia Obama and Sasha Obama
   Predicted answer 1: sasha and malia obama
   Predicted answer 2: most people would say Malia and Sasha, but I'm not sure and would have to double
        check
   Predicted answer 3: Barack Obama has two daughters. Their names are Malia Ann and Natasha Marian,
        but they are commonly referred to as Malia Obama and Sasha Obama.

These predicted answers are all CORRECT because:
   - They fully contain the important information in the gold target.
   - They do not contain any information that contradicts the gold target.
   - Only semantic meaning matters; capitalization, punctuation, grammar, and order don't matter.
   - Hedging and guessing are permissible, provided that the gold target is fully included and the
        response contains no incorrect information or contradictions.
```

```
The following are examples of INCORRECT predicted answers.

   Question: What are the names of Barack Obama's children?
   Gold target: Malia and Sasha
   Predicted answer 1: Malia.
   Predicted answer 2: Malia, Sasha, and Susan.
   Predicted answer 3: Barack Obama does not have any children.
   Predicted answer 4: I think it's either Malia and Sasha. Or it could be Malia and Jackie. Or it
        could be Joey and Malia.
   Predicted answer 5: While I don't know their exact names, I can tell you that Barack Obama has three
        children.

These predicted answers are all INCORRECT because:
   - A factual statement in the answer contradicts the gold target. Incorrect statements that have some
        hedging (e.g., "it is possible that", "although i'm not sure, i think") are also considered
        incorrect.

The following are examples of NOT_ATTEMPTED predicted answers.

   Question: What are the names of Barack Obama's children?
   Gold target: Malia and Sasha
   Predicted answer 1: I don't know.
   Predicted answer 2: I need more context about which Obama you are talking about.
   Predicted answer 3: Without researching the web, I cannot answer this question. However, I can tell
        you that Barack Obama has two children.
   Predicted answer 4: Barack Obama has two children. I know that one of them is Malia, but I'm not
        sure about the other one.

These predicted answers are all NOT_ATTEMPTED because:
   - The important information in the gold target is not included in the answer.
   - No statements in the answer contradict the gold target.

Also note the following things:
   - For grading questions where the gold target is a number, the predicted answer needs to be correct
        to the last significant figure in the gold answer.
   - The gold target may contain more information than the question. In such cases, the predicted
        answer only needs to contain the information that is in the question.
   - Do not punish predicted answers if they omit information that would be clearly inferred from the
        question.
   - Do not punish for typos in people's name if it's clearly the same name.
```

Listing 12: SimpleQA evaluation user prompt

```
Here is a new example. Simply reply with either CORRECT, INCORRECT, NOT ATTEMPTED. Don't
apologize or correct yourself if there was a mistake; we are just trying to grade the
answer.
   '''
   Question: {question}
   Gold target: {target}
   Predicted answer: {predicted_answer}
   '''
Grade the predicted answer of this new question as one of:
A: CORRECT
B: INCORRECT
C: NOT_ATTEMPTED
Just return the letters "A", "B", or "C", with no text around it.
```

Corpus Retrieval Depth

The answer reconstruction phase retrieves Wikipedia passages to find candidate sentences. We vary retrieval depth to understand the coverage-efficiency tradeoff. Table 1 shows results for WikiQA/DeBERTa using Map_R.

At 100 passages, performance falls 4.6 points below baseline with high fallback rate

Table 1: Effect of corpus retrieval depth on answer reconstruction quality for WikiQA. Deltas computed relative to original baseline.

Passages	Avg Candidates	Fallback %	P@1	ΔP@1	MAP	ΔMAP
100	412	14.3%	$76.82_{\pm 1.43}$	−4.61	$83.41_{\pm 1.12}$	−4.57
250	1034	8.7%	$79.56_{\pm 0.91}$	−1.87	$86.12_{\pm 0.67}$	−1.86
500	2147	5.9%	$81.43_{\pm 0.42}$	0.00	$87.87_{\pm 0.16}$	−0.11
1000	4283	4.2%	$81.67_{\pm 0.38}$	+0.24	$88.03_{\pm 0.21}$	+0.05

(14.3%). At 500 passages, we match baseline ($\Delta = 0.00$) with acceptable 5.9% fallback. Further increasing to 1000 passages provides negligible improvement while doubling computational cost. We adopt 500 passages as the default throughout experiments.

Computational Cost

Table 2 breaks down processing time for declassifying WikiQA (2,118 examples).

Table 2: Computational costs for declassifying WikiQA training set. All experiments on 8× NVIDIA L40S GPUs.

Stage	Time	Notes
Question retrieval (QUADRo)	1–2 min	Full search, 10–35ms/question
Question reranking (QRP)	8 min	Cross-encoder inference
Corpus retrieval (BGE)	45 min	500 passages × 2,118 queries
Sentence segmentation	5 min	NLTK processing
LLM annotation	∼6 hrs	Qwen-3-80B, sharded across 8 GPUs
Total	**∼7 hrs**	

LLM annotation dominates processing time. WikiQA averages ∼63 answer candidates per question. Costs scale linearly with dataset size: TrecQA (94 examples) takes ∼30 minutes total.